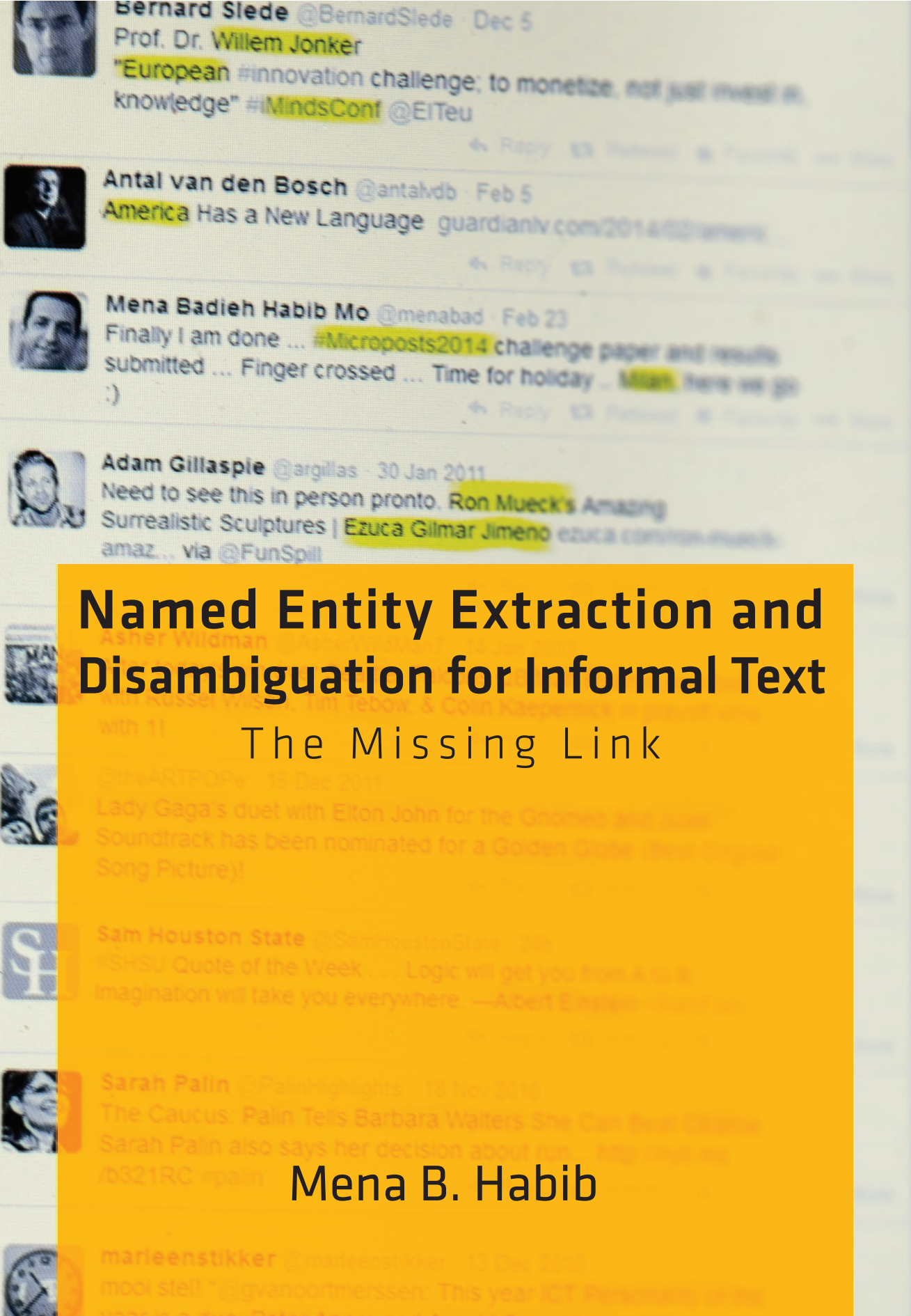


Named Entity Extraction and Disambiguation for Informal Text

The Missing Link

Mena B. Habib

Named Entity Extraction and Disambiguation for Informal Text
The Missing Link

Mena B. Habib

PhD dissertation committee:

Chairman and Secretary:
Prof. dr. P.M.G. Apers, University of Twente, NL
Promotor:
Prof. dr. P.M.G. Apers, University of Twente, NL
Assistant promotor:
Dr. ir. M. van Keulen, University of Twente, NL
Members:
Prof. dr. W. Jonker, University of Twente, NL
Prof. dr. F.M.G. de Jong, University of Twente, NL
Prof. dr. A. van den Bosch, Radboud University Nijmegen, NL

CTIT Ph.D. thesis Series No. 14-301
Centre for Telematics and Information Technology
P.O. Box 217, 7500 AE
Enschede, The Netherlands.

SIKS Dissertation Series No. 2014-20
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN: 978-90-365-3647-9
ISSN: 1381-3617 (CTIT Ph.D. thesis Series No. 14-301)
DOI: 10.3990/1.9789036436479
http://dx.doi.org/10.3990/1.9789036536479

Cover design: Hany Maher
Printed by: Ipskamp Drukkers

NAMED ENTITY EXTRACTION AND DISAMBIGUATION FOR INFORMAL TEXT
THE MISSING LINK

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Friday, May 9th, 2014 at 12:45

by

Mena Badieh Habib Morgan

born on June 29th, 1981
in Cairo, Egypt

This dissertation is approved by:

Prof. dr. P.M.G. Apers (promotor)
Dr. ir. M. van Keulen (assistant promotor)

Dedicated to the soul of my father

Acknowledgments

"I can do all things through Christ who strengthens me. (Philippians 4:13)"

I always say that I am lucky. I am lucky because I always get wonderful and kind people surrounding me.

I am lucky to have Peter Apers as my promoter. He supported my research directions and gave me freedom and independence. His words always gave me confidence and insistence to complete my PhD.

I am lucky to have Maurice van Keulen as my daily supervisor. Although we passed some foggy times, he never lost his positive attitude. He was always there to give advice, optimism, support and ideas. Besides learning how to be a good researcher, I have learned from Maurice how to be a supervisor, which is something I would definitely need through my academic career. Words could never express my sincere gratitude to Maurice.

I am lucky to have Willem Jonker, Franciska de Jong, and Antal van den Bosch as my committee members. I would like to thank them for their careful reading of my thesis.

I am lucky to be a member of the databases group at the university of Twente. I would like to thank them all for providing me the pleasant working climate. Thanks for Maarten Fokkinga, Djoerd Hiemstra, Andreas Wombacher, Robin Aly, Ida den Hamer-Mulder, Suse Engbers, Jan Flokstra, Iwe Muiser, Juan Amiguet, Sergio Duarte, Victor de Graaff, Rezwan Huq, Mohammad Khelghati, Kien Tjin-Kam-Jet, Brend Wanders, Zhemin Zhu, Lei Wang, Ghita Berrada, Almer Tigelaar, Riham Abdel Kader and Dolf Trieschnigg.

I would like to dedicate a special thanks to couple of them Ida den Hamer-Mulder and Juan Amiguet. Ida, the dynamo of the group. Ida helped me with my settlement in the Netherlands. She offered help even for things beyond her duty. The DB group is really lucky to have Ida as their secretary. Juan, my office mate, the person who knows at least one thing about everything. The man who is willing to help at any time. Juan, I am grateful for your help and for our nice conversations we had together discussing almost everything from food recipes to astronomy.

I am lucky to spend my PhD life period at this peaceful quiet spot of the world called Enschede. In Enschede life is easy! I would like also to express my gratitude towards the Egyptian Coptic community in the Netherlands who helped me to overcome my home

sickness. Thanks for bishop Arsany, father Maximos, father Pavlos, Samuel Poulos, Adel Saweiros, Sameh Ibrahim, Moneer Basalyous and Maher Rasla.

I am lucky because I did an internship at the highly reputable databases and information systems group of the Max Planck Institute of Informatics in Saarbrucken, Germany. I learned a lot during my stay there. Thanks for Gerhard Weikum, Marc Spaniol, Mohamed Amir and Johannes Hoffart.

I am lucky to study and work at the Faculty of Computers and Information Sciences in Ain Shams University in Cairo where I received my Bachelor and Master degrees. I would like to thank all my professors and colleagues there specially Abdel-Badieh Salem, Mohammed Roushdy, Mostafa Aref, Tarek Gharib, Emad Monier, Ayad Barsom, Marco Alfonse and many others.

I am lucky to be the son of Badieh Habib and Aida Makien. My parents who did their best to raise me up as researcher. I genetically inherited my interest towards research, math and science from them. I hope I was able to achieve their wishes. I also could never forget to thank my sisters Hanan and Eman in addition to the rest of my family and my family in law who always provide love and support.

I am lucky to have Shery, my lovely wife who did her best to offer the best atmosphere for me. The lady who provide unconditional care and love. Indeed, '*Who can find a virtuous woman? For her price is far above rubies.' (Proverbs 31:10)*.

I am lucky to have Maria and Marina, my sweet twin angels. Whenever I am stressed, only one hour playing with them was enough to release all stress and added smile to my face.

I am lucky to get my Christian doctrine at the Sunday school of Saint George church in El-Matariya, Cairo. The church where I lived my best days ever between its walls. It strongly participated in building my personality. I would like to thank all the church fathers Georgios Botros, Beshoy Boules, Tomas Naguib, Pola Fouad and Shenouda Dawood. I also could never forget all my teachers there, specially Onsy Naguib, for their care, love and support.

Finally, I am lucky to have my friends with whom I shared my best life moments. Thanks for Ehab Gamil, Gerges Saber, Maged Makram, Maged Matta, Mena George, Mena Samir, Mena William, Ramy Anwar, Romany Edwar, Sameh Samir and many others. Thanks for everyone I shared my dreams with one day.

I am lucky to have all these people surrounding me. This thesis would have been much different (or would not exist) without these people.

No it is not luck.. It is God's hand who leads me through life. He said "*I have raised him up in righteousness, and I will direct all his ways. (Isaiah 45:13)*"

Mena B. Habib
Enschede, March 2014.

Contents

List of Figures

List of Tables

Listings

Part I

Introduction

CHAPTER 1

Introduction

1.1 Introduction

Computers cannot understand natural languages like humans do. Our ability to easily distinguish between multiple word meanings is developed in a lifetime of experience. Using the context in which a word is used, a fundamental understanding of syntax and logic, and a sense of the speaker's intention, we understand what another person is telling us or what we read. It is the aim of the Natural Language Processing (NLP) society to mimic the way humans understand natural languages. Although efforts spent for more than 50 years by linguists and computer scientists to get computers to understand human language, there is still long way to go to achieve this goal.

A main challenge of natural language is its ambiguity and vagueness. The basic definition of ambiguity, as generally used in natural language processing, is "*capable of being understood in more than one way*". Scientists try to resolve ambiguity, either semantic or syntactic, based on properties of the surrounding context. Examples include, Part Of Speech (POS) tagging, morphology analysis, Named Entity Recognition (NER), and relations (facts) extraction. To automatically resolve ambiguity, typically the grammatical structure of sentences is used, for instance, which groups of words go together (phrases) and which words are the subject or object of a verb. However, when we move to informal language widely used in social media, the language becomes more ambiguous and thus more challenging for automatic understanding.

What? The rapid growth in the IT in the last two decades leads to the growth in the amount of information available on the World Wide Web (WWW). Social media content represents a big part of all textual content appearing on the Internet. According to an eMarketer report [1], nearly one in four people worldwide will use social networks in 2013. The number of social

network users around the world rose to 1.73 billion in 2013. By 2017, the global social network audience will total 2.55 billion. Twitter as an example of highly active social media network, has 140 million active users publishing over 400 million tweet every day[1].

Why? These streams of user generated content (UGC) provide an opportunity and challenge for media analysts to analyze huge amount of new data and use them to infer and reason with new information. Making use of social media content requires measuring, analyzing and interpreting interactions and associations between people, topics and ideas. An example of a main sector for social media analysis is the area of customer feedback through social media. With so many feedback channels, organizations can mix and match them to best suit corporate needs and customer preferences.

Another beneficial sector is social security. Communications over social networks have helped to put entire nations to action. Social media played a key role in The Arab Spring that started in 2010 in Tunisia. The riots that broke out across England during the summer of 2011 also showed the power of social media. The growing criminality associated with social media has been an alarm to government security agencies. There is a growing demand to automatically monitor the discussions on social media as a source of intelligence. Nowadays, increasing numbers of people within investigative agencies are being deployed to monitor social media. Unfortunately, the existing tools and technologies used are limited because they are based on simple keyword selection and classification instead of reasoning with meaningful information. Furthermore, the processes followed are time and resources consuming. There is also a need for new tools and technologies that can deal with the informal language widely used in social media.

How? Information Extraction (IE) is the research field that enables the use of such a vast amount of unstructured distributed data in a structured way. IE systems analyze human language in order to extract information about different types of events, entities, or relationships. Named Entity Extraction (NEE) is a sub task of IE that aims to locate phrases (mentions) in the text that represent names of persons, organizations or locations regardless of their type. It differs from the term Named Entity Recognition (NER) which involves both extraction and classification to one of the predefined set of classes. Named Entity Disambiguation (NED) is the task of exploring which correct person, place, event, etc. is referred to by a mention. NEE and NED have become a basic steps of many technologies like Information Retrieval (IR), Question Answering (QA).

[1] `https://blog.twitter.com/2012/twitter-turns-six`

Although state-of-the-art NER systems for English produce near-human performance [2], their performance drops when applied to informal text of UGC where the ambiguity increases. It this the aim of this thesis to study not only the tasks of NEE and NED for semi-formal and informal text but also their interdependency and show how one could be used to improve the other and vice versa. We call this potential for mutual improvement, the *reinforcement effect*. It mimics the way humans understand natural language. Natural language processing (NLP) tasks are commonly split into a set of pipelined sub tasks. The residual error produced in any sub task propagates, adversely affecting the end objectives. This is why we believe that back propagation would help improving the overall system quality. We show the benefit of using this *reinforcement effect* on two domains: NEE and NED for toponyms in semi-formal text that represents advertisements for holiday properties; and for arbitrary entity types in informal short text in tweets. We proved that this mutual improvement makes NEE and NED robust across languages and domains. This improvement is also independent on what extractions and disambiguation techniques are used. Furthermore, we developed extraction methods that consider alternatives and uncertainties in text with less dependency on formal sentence structure. This leads to more reliability in cases of informal and noisy UGC text.

1.2 Examples of Application Domains

Information extraction has applications in a wide range of domains. There are many stakeholders that could benefit from UGC on social media. Here, we give some examples for applications of information extraction:

- Security agencies typically analyze large amounts of text manually to search for information about people involved in criminal or terrorism activities. Social media is a continuously instantly updated source of information. Football hooligans sometimes start their fight electronically on social media networks even before the sport event. Another real life example is the Project X Haren[2]. Project X Haren was an event that started out as a public invitation to a birthday party by a girl on Facebook, but ended up as a gathering of thousands of youths causing riots in the town of Haren, Groningen. Automatic monitoring and gathering of such information could be helpful to take actions to prevent such violent, and

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

destructive behaviors. As an example for real application, we contribute to the TEC4SE project[3]. The aim of the project is to improve the operational decision-making within the security domain by gathering as much information available from different sources (like cameras, police officers on field, or social media posts). Then these information is linked and relationships between different information streams are found. The result is a good overview of what is happening in the field of security in the region. Our contribution to this project is to the enrich Twitter stream messages by extracting named entities at run time. The amount and the nature of the flowing data is beyond the possibility of manually tracking. This is why we need new technologies that is capable of dealing with such huge noisy amounts of data.

- As users become more involved in creating contents in a virtual world, more and more data is generated in various aspects of life for studying user attitudes and behaviors. Social sciences study human behavior by studying their physical space and belongings. Now, it is possible to investigate users by studying their online activities, postings, and behavior in a virtual space. This method can be a replacement for traditional surveys and experiments [3]. Prediction and understanding of the attitudes and behaviors of individuals and groups based on the sentiment expressed within online virtual communities is a natural area of research in the Internet era. To reach this goal, social scientists are in dire need of stronger tools to provide them with the required data for their studies.

- Financial experts always look for specific information to help their decision making. Social media can be a very important source of information about the attitudes and behaviors of stakeholders. In general, if extracted and analyzed properly, the data on social media can lead to useful predictions of certain human related events. Such prediction has great benefits in many realms, such as finance, product marketing and politics [4]. For example, a finance company may want to know the stakeholders' reaction towards some political action. Automatically finding such information from user posts on social media requires special information extraction technologies to analyze the noisy social media streams and capture such information.

- With the fast growth of the Web, search engines have become an integral part of people's daily lives, and users search behaviors are much better

[3] `http://www.tec4se.nl/`

Table 1.1: Some challenging cases for NEE and NED in tweets (NE mentions are written in bold).

Case #	Tweet Content
1	– **Lady Gaga** - **Speechless** live @ **Helsinki** 10/13/2010 http://www.youtube.com/watch?v=yREociHyijk ... @ladygaga also talks about her Grampa who died recently
2	**Qld** flood victims donate to **Vic** bushfire appeal
3	**Laelith Demonia** has just defeated **liwanu Hird**. Career wins is 575, career losses is 966.
4	Adding **Win7Beta**, **Win2008**, and **Vista** x64 and x86 images to munin. #wds
5	history should show that **bush jr** should be in jail or at least never should have been president
6	RT @BBCClick: Joy! **MS Office** now syncs with **Google Docs** (well, in beta anyway). We are soon to be one big happy (cont) http://tl.gd/73t94u
7	"Even Writers Can Help..An Appeal For **Australian** Bushfire Victims" http://cli.gs/Zs8zL2

understood now. Search based on bag-of-words representation of documents can no longer provide satisfactory results. More advanced information needs such as entity search, and question answering can provide users with better search experience. To facilitate these search capabilities, information extraction is often needed as a pre-processing step to enrich the document with information in structured form.

1.3 Challenges

NEE and NED in informal text are challenging. Here we summarize the challenges of NEE and NED for tweets as an example of informal text:

- The informal nature of tweets makes the extraction process more difficult. For example, in table 1.1 case 1, it is hard to extract the mentions (phrases that represent NEs) using traditional NEE methods because of the ill-formed sentence structure. Traditional NEE methods might extract '*Grampa*' as a mention because of it capitalization. Furthermore, it is hard to extract the mention '*Speechless*', which is a name of a song, as it requires

further knowledge about '*Lady Gaga*' songs.

- The limited length (140 characters) of tweets forces the senders to provide dense information. Users resort to acronyms to reserve space. Informal language is another way to express more information in less space. All of these problems make both the extraction and the disambiguation processes more complex. For example, in table 1.1 case 2 shows two abbreviations ('*Qld*' and '*Vic*'). It is hard to infer their entities without extra information.

- The limited coverage of a Knowledge Base (KB) is another challenge facing NED for tweets. According to [5], 5 million out of 15 million mentions on the web cannot be linked to Wikipedia. This means that relying only on a KB for NED leads to around 33% loss in disambiguated entities. This percentage is higher on Twitter because of its social nature where users discuss information about infamous entities. For example, table 1.1 case 3 contains two mentions for two users on the '*My Second Life*' social network. It is very unlikely that one could find their entities in a KB. However, their profile pages ('`https://my.secondlife.com/laelith.demonia`' and '`https://my.secondlife.com/liwanu.hird`') can be found easily by a search engine.

- Named entity (NE) representation in KB implies another NED challenge. YAGO KB [6] uses Wikipedia anchor text as possible mention representation for named entities. However, there might be more representations that do not appear in Wikipedia anchor text. Either because of misspelling or because of a new abbreviation of the entity. For example, in table 1.1 case 4, the mentions '*Win7Beta*' and '*Win2008*' do not appear in YAGO KB mention-entity look-up table, although they refer to the entities '`http://en.wikipedia.org/wiki/Windows_7`' and '`http://en.wikipedia.org/wiki/Windows_Server_2008`' respectively.

- The processes of NEE and NED involve degrees of uncertainty. For example, in table 1.1 case 5, it is uncertain whether the word *jr* should be part of the mention *bush* or not. Same for '*Office*' and '*Docs*' in case 6 which some extractors may miss. Another example, in case 7, it is hard to assess whether '*Australian*' should refer to '`http://en.wikipedia.org/wiki/Australia`' or '`http://en.wikipedia.org/wiki/Australian_people`'[4]. Both might be

[4]Some NER datasets consider nationalities as NEs [7].

Table 1.2: Some challenging cases for toponyms extraction in semi-formal text (toponyms are written in bold).

Case #	Semi-formal Text Samples
1	**Bargecchia** 9 km from **Massarosa**.
2	**Olšova Vrata** 5 km from **Karlovy Vary**.
3	Bus station in **Armacao de Pera** 4 km.
4	Airport 1.5 km (2 planes/day).

correct. This is why we believe that it is better to consider possible alternatives in the processes of NEE and NED.

- Another challenge is the freshness of the KBs. For example, the page of '*Barack Obama*' on Wikipedia was created on 18 March 2004. Before that date '*Barack Obama*' was a member of the Illinois Senate and you could find his profile page on '`http://www.ilga.gov/senate/Senator.asp?MemberID=747`'. It is very common on social networks that users talk about some infamous entity who might become later a public figure.

- Informal nature of language used in social media implies many different random representations of the same fact. This adds new challenges to machine learning approaches which need regular patterns for generalization. We need new methods that require less training data and generalize well at the same time.

Semi-formal text is text lacking the formal structure of the language but follows some pattern or format like product descriptions and advertisements. Although semi-formal text involves some regularity in representing information, this regularity implies some challenges.

In table 1.2, cases 1 and 2 show two examples for true toponyms included in a holiday description. Any machine learning approach uses cases 1 and 2 as training samples will annotate '*Airport*' as a toponym following the same pattern of having a capitalized word followed by a number and the word '*km*'. Furthermore, the state-of-the-art approaches performs poorly on this type of text. Figure 1.1 shows the results of the application of three of the leading Stanford NER models[5] on a holiday property description text (see figure 3.6a). Regardless of NE classification, even the extraction (determining if a phrase

[5] `http://nlp.stanford.edu:8080/ner/process`

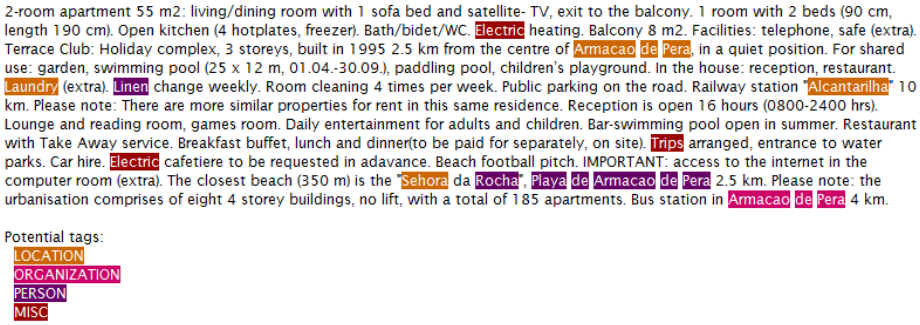
represents a NE or not) is performing poorly. Problems vary between *a*) extracting false positives (like '*Electric*' and '*Trips*' in figure 1.1a); or *b*) missing some true positives (like '*Sehora da Rocha*' in figures 1.1b and 1.1c); or *c*) partially extracting the NE (like '*Sehora da Rocha*' in figures 1.1a and '*Armacao de Pera*' in figure 1.1b).

1.4 General Approach

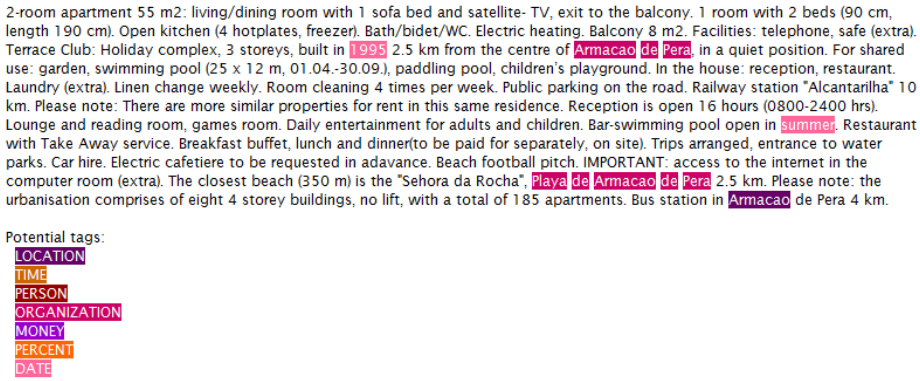
Natural language processing (NLP) tasks are commonly composed of a set of chained sub tasks that form the processing pipeline. The residual error produced in these sub tasks propagates, affecting the final process results. In this thesis we focus on NEE and NED which are two common processes in many NLP applications. We claim that feedback derived from disambiguation would help in improving the extraction and hence the disambiguation. This is the same way we as humans understand text. The capability to successfully understand language requires one to acquire a range of skills including syntax, semantics, and an extensive vocabulary. We try to mimic a human's way of reasoning to solve the NEE and NED problems. Consider the tweet in table 1.1 case 1. One would use syntax knowledge to recognize '10/13/2010' as a date. Furthermore, prior knowledge enables one to recognize '*Lady Gaga*' and '*Helsinki*' as a singer name and location name respectively or at least as names if one doesn't know exactly what they refer to. However, the term '*Speechless*' involves some ambiguity as it could be an adjective and also could be a name. A feedback clue from '*Lady Gaga*' would increase one's certainty that it refers to a song. Even without knowing that '*Speechless*' is a song of '*Lady Gaga*', there are sufficient clues to guess with quite high probability that it is a song. The pattern 'live @' in association with disambiguating '*Lady Gaga*' as a singer name and '*Helsinki*' as a location name, leads to infer '*Speechless*' as a song.

Although the logical order for a traditional Information Extraction (IE) system is to complete the extraction process before commencing the disambiguation, we start with an initial phase of extraction which aims to achieve high recall (find as many reasonable mention candidates as possible) then we apply the disambiguation for all the extracted possible mentions. Finally we filter those extracted mention candidates into true positives and false positives using features (clues) derived from the results of the disambiguation phase such as KB information and entity coherency. Figure 1.2 illustrates our general approach.

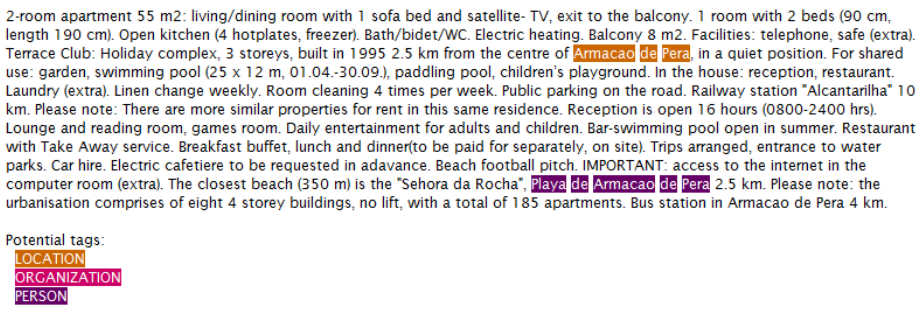
Unlike NER systems which extract entities mentions and assign them to



(a) Stanford 'english.conll.4class.distsim.crf.ser' model.



(b) Stanford 'english.muc.7class.distsim.crf.ser' model.



(c) Stanford 'english.all.3class.distsim.crf.ser' model.

Figure 1.1: Results of Stanford NER models applied on semi-formal text of holiday property description.

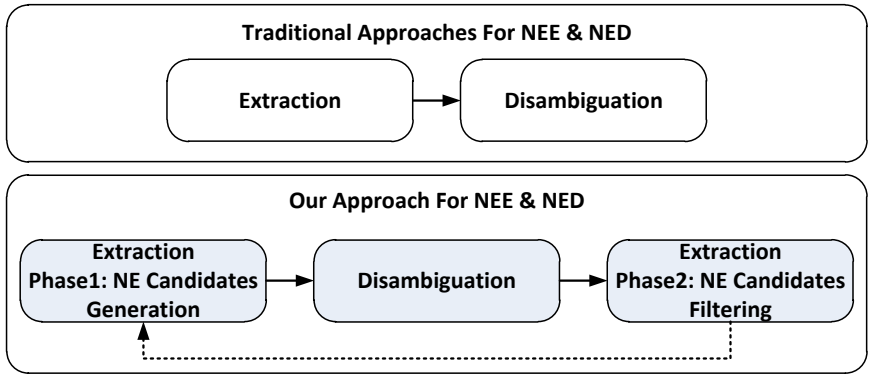


Figure 1.2: Traditional approaches versus our approach for NEE and NED.

one of the predefined categories (like location, person, organization), we focus first on extracting mentions regardless of their categories. We leave this classification to the disambiguation step which links the mention to its real entity.

The potential of this order is that the disambiguation step can give extra clues (such as entity-context similarity and entity-entity coherency) about each NE candidate. This information can help in the decision whether the candidate is a true NE or not.

The general principal we claim is that NED could be very helpful in improving the NEE process. For example, consider the tweet in case 1 in table 1.1. It is uncertain, even for humans, to recognize '*Speechless*' as a song name without having prior information about songs of '*Lady Gaga*'. Our approach is able to solve such problematic cases of named entities.

1.5 Research Questions

The main theme of this thesis is to study NEE and NED and their interdependency in semi-formal and informal text. Within this theme, we need to answer the following research questions regarding the relation between NEE and NED:

- How do the imperfection and the uncertainty involved in the extraction process affect the effectiveness of the disambiguation process and how

can the extraction confidence probabilities be used to improve the effectiveness of disambiguation?

- How can the disambiguation results be used to improve the certainty of extraction and what are the evidences and features that could be derived from disambiguation to improve extraction process?
- How robust is the *reinforcement effect* and whether this concept is valid across domain, approaches, and languages?
- How can we overcome the limited coverage of knowledge-bases and how can the limited context of short messages be enriched?

We investigate the answers for the aforementioned questions on two domains: NEE and NED for toponyms in semi-formal text; and for arbitrary entity types in informal short text of tweets.

1.6 Contributions

The main goal of the thesis is to mimic the human way of recognition and disambiguation of named entities specially for domains that lack formal sentence structure. The proposed methods open the doors for more sophisticated applications based on users' contributions on social media.

Particularly, the thesis makes the following contributions:

- We obtained more insight into how computers can truly understand natural languages by mimicking human ways of language understanding.
- We propose a robust combined framework for NEE and NED in semi and informal text. The achieved robustness of NE extraction obtained from this principle has been proven for several aspects:
  - It is independent on the used combination of the extraction and the disambiguation techniques. It can be applied on any of the widely used extraction techniques: list look-up; rule-based; and statistical. It has also been proven to work with different disambiguation algorithms.
  - Once a system is developed, it can trivially be extended to other languages; all that is needed is a suitably amount of training data for the new language. In this case, we avoid using language dependent features like part of speech (POS) tagging.

  - It works in a domain-independent manner. It generalizes to any dataset. It is suitable for closed domain tasks as well as for open world applications.
  - It is shown to be robust against a shortage of labelled training data, the coverage of KBs, and the informality of the used language.

- We propose the reinforcement approach which makes use of disambiguation results feedback to improve extraction quality.

- We propose a method of handling the uncertainty involved in extraction to improve the disambiguation results.

- We propose a generic approach for NED in tweets for any named entity (not entity oriented). This approach overcomes the problem of limited coverage of KBs. Mentions are disambiguated by assigning them to either a Wikipedia article or a home page. We also introduce a method to enrich the limited entity context.

1.7 Thesis Structure

The thesis is mainly composed of four parts: an introductory part; a part on NEE and NED of toponyms in semi-formal text; a part on NEE and NED in tweets; and a final concluding part. The detailed description of chapters' contents are shown as follows:

- **Part II:**

  - Chapter 2 presents the related work on toponyms extraction and disambiguation.
  - Chapter 3 proofs the existence of the *reinforcement effect* shown on toponyms extraction and disambiguation in holiday cottages descriptions.
  - Chapter 4 exploits *reinforcement effect*. It examines how handling the uncertainty of extraction influences the effectiveness of disambiguation, and reciprocally, how the result of disambiguation can be used to improve the effectiveness of extraction through iteration process. Statistical methods of extraction are tested.

  - Chapter 5 tests the robustness of the *reinforcement effect* over multiple languages (English, Dutch and German) and over variable extraction model settings.

- **Part III:**
  - Chapter 6 presents the related work on NEE and NED in informal text.
  - Chapter 7 presents a proof of concept for our principles applied on tweets. It describes an unsupervised approach for extraction and disambiguation.
  - Chapter 8 presents a generic open world approach for NED for tweets.
  - Chapter 9 presents TwitterNEED, a hybrid supervised approach for NEE and NED for tweets.

- **Part IV:**
  - Chapter 9 concludes and proposes future work directions.
  - Appendix A presents a motivating application.
  - Appendix B presents our participation in #MSM2013 concept extraction challenge [7].

Part II

Toponyms in Semi-formal Text

CHAPTER 2

Related Work

2.1 Summary

Toponyms are named entities which represent location names in text. Toponym extraction and disambiguation are special cases of the more general problem Named Entity Recognition (NER) and Disambiguation (NED) which are main steps in any Information Extraction (IE) system. In this chapter, we introduce Information Extraction (IE) and its phases then we briefly survey the major approaches for Named Entity Recognition and Named Entity Disambiguation in literature.

2.2 Information Extraction

NEE and NED are two processes in the Information Extraction (IE) systems pipeline. IE systems extract domain-specific information from natural language text. The domain and types of information to be extracted must be defined in advance. IE systems often focus on object identification, such as references to people, places, companies, and physical objects. Domain-specific extraction patterns (or something similar) are used to identify relevant information [8]. Figure 2.1 shows an example for a piece of text represents news story as an input for IE system while tables 2.1, 2.2 and 2.3 show respectively the extracted named entities, facts, and a filled template for product release event from that text.

A typical IE system has basic phases for input: tokenization, lexical analysis, name entity recognition, syntactical analysis, and identification of the interesting information required in a particular application [9]. Depending on the particular requirements of the application, IE systems may also include other modules. Figure 2.2 shows the modules that comprise a typical IE system.

Fletcher Maddox, former Dean of the UCSD Business School, announced the formation of La Jolla Genomatics together with his two sons. La Jolla Genomatics will release its product Geninfo in June 1999. Geninfo is a turnkey system to assist biotechnology researchers in keeping up with the voluminous literature in all aspects of their field.
Dr. Maddox will be the firm's CEO. His son, Oliver, is the Chief Scientist and holds patents on many of the algorithms used in Geninfo. Oliver's brother, Ambrose, follows more in his father's footsteps and will be the CFO of L.J.G. headquartered in the Maddox family's hometown of La Jolla, CA.

Figure 2.1: Text represents news story.

Table 2.1: Named Entities extracted from text in Figure 2.1.

Persons	Fletcher Maddox, Dr. Maddox, Oliver, Oliver, Ambrose, Maddox.
Organizations	UCSD Business School, La Jolla Genomatics, La Jolla Genomatics, L.J.G.
Locations	La Jolla, CA.
Artifacts	Geninfo, Geninfo.
Dates	June 1999

Table 2.2: Facts extracted from text in Figure 2.1.

Person	**Employee_of**	**Organization**
Fletcher Maddox	Employee_of	UCSD Business School
Fletcher Maddox	Employee_of	La Jolla Genomatics
Oliver	Employee_of	La Jolla Genomatics
Ambrose	Employee_of	La Jolla Genomatics
Artifact	**Product_of**	**Organization**
Geninfo	Product_of	La Jolla Genomatics
Location	**Location_of**	**Organization**
La Jolla	Location_of	La Jolla Genomatics
CA	Location_of	La Jolla Genomatics

Tokenization phase identifies the sentences boundaries and splits each sentence into set of tokens. Splitting is performed along a predefined set of delim-

Table 2.3: Product release event extracted from text in Figure 2.1.

Company	La Jolla Genomatics
Product	Geninfo
Date	June 1999
Cost	

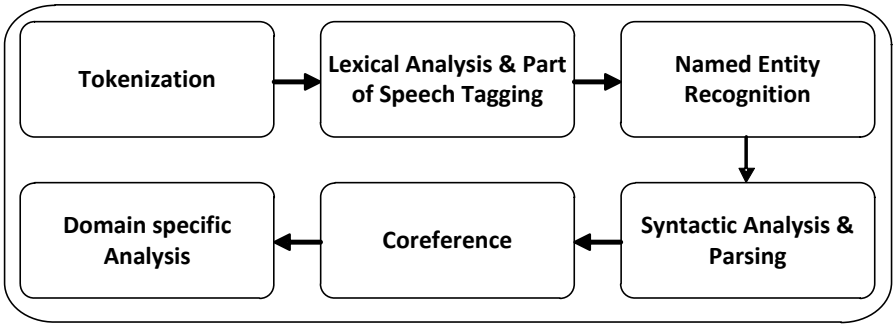


Figure 2.2: Modules for a typical IE System.

iters like spaces, commas, and dots. A token is a word or a digit, or a punctuation.

In the lexical analysis the tokens determined by the tokenization module are looked up in the dictionary to determine their possible parts of speech (POS) tags and other lexical features that are required for subsequent processing. This module assigns to each word a grammatical category coming from a fixed set. The set of tags includes the conventional part of speech such as noun, verb, adjective, adverb, article, conjunct, and pronoun. Examples of well-known tag sets are the Brown tag set which has 179 total tags, and the Penn tree bank tag set that has 45 tags [10].

The next phase of processing identifies various types of proper names and other special forms, such as dates and currency amounts. Names appear frequently in many types of texts, and identifying and classifying them simplifies further processing. Furthermore, names are important for many extraction tasks. Names could be identified by a set of regular expressions which are stated in terms of parts of speech, syntactic features, and orthographic features (e.g., capitalization). Personal names, for example, might be identified by a

preceding title.

The goal of syntactic analyzer is to give a syntactic description to the text. The analyzer marks every word with a syntactic tag. The tags denote the subjects, objects, main verbs, etc. Identifying syntactic structure simplifies the subsequent phase of events extraction. After all, the arguments to be extracted often correspond to noun phrases in the text, and the relationships to be extracted often correspond to grammatical functional relations.

Given a text, relevant entities may be referred to in many different ways. Thus, success on the IE task is dependent on the success at determining when one noun phrase referred to the same entity as another noun phrase.

The domain analysis is the final module of IE systems. The preceding modules prepare the text for the domain analysis by adding semantic and syntactic features to it. This module is responsible of filling the templates. These templates consist of a collection of slots (i.e., attributes), each of which may be filled by one or more values.

2.3 Named Entity Recognition

NER is a subtask of Information Extraction (IE) that aims to annotate phrases in text with its entity type such as names (e.g., person, organization or location name), or numeric expressions (e.g., time, date, money or percentage). The term 'named entity recognition' was first mentioned in 1996 at the Sixth Message Understanding Conference (MUC-6) [11], however the field started much earlier.

The vast majority of proposed approaches for IE in general and NEE in particular fall in two categories: Hand-crafted rule-based approaches and machine learning-based approaches.

2.3.1 Rule-based Approaches

Rule-based approaches are the earliest for information extraction. Rule-based IE systems consist of a set of linguistic rules. Those rules are represented as regular expressions or as zero or higher order logic. Rules are more useful when the task is controlled and well-behaved like the extraction of phone numbers and zip codes from emails. Rules are either manually coded, or learned from example labeled sources.

One of the earliest rule-based systems is FASTUS [12]. FASTUS is a system for extracting information from natural language text for entry into a database

and for other applications. It works essentially as a cascaded, nondeterministic finite state automaton. There are five stages in the operation of FASTUS. In stage 1, names and other fixed form expressions are recognized. In stage 2, basic noun groups, verb groups, and prepositions and some other particles are recognized. In stage 3, certain complex noun groups and verb groups are constructed. Patterns for events of interest are identified in stage 4 and corresponding event structures are built. In stage 5, distinct event structures that describe the same event are identified and merged, and these are used in generating database entries. This decomposition of language processing enables the system to do exactly the right amount of domain-independent syntax, so that domain-dependent semantic and pragmatic processing can be applied to the right larger-scale structures.

Another rule-based approach is LaSIE [13, 14]. LaSIE involves compositionally constructing semantic representations of individual sentences in a text according to semantic rules attached to phrase structure constituents which have been obtained by syntactic parsing using a corpus-derived context-free grammar. For NER, LaSIE matches the input text against pre-stored lists of proper names, date forms, currency names, etc. and by matching against lists of common nouns that act as reliable indicators or triggers for classes of named entity. These lists are compiled via flex program into a finite state recognizer. Each sentence is fed to the recognizer and all single and multi-word matches are used to associate token identifiers with named entity tags. Lists of names are employed for locations, personal titles, organizations, dates/times and currencies. The grammar rules for Named Entity items constitute a subset of the system's noun phrase (NP) rules. All the rules were produced by hand. The rules make use of part of speech tags, semantic tags added in the gazetteer look-up stage, and if necessary the lexical items themselves.

A language that is designed for rule-based IE tasks is Java Annotation Patterns Engine (JAPE). It is a component of the open-source General Architecture for Text Engineering (GATE) platform [15]. It provides finite state transduction over annotations based on regular expressions. JAPE is a version of The Common Pattern Specifications Language (CPSL) [16]. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left-hand-side (LHS) of the rules consist of an annotation pattern description. The right-hand-side (RHS) consists of annotation manipulation statements. Annotations matched on the LHS of a rule may be referred to on the RHS by means of labels that are attached to pattern elements. More details about JAPE rules will be discussed later in chapter 3. More elaborate

discussion of rule-based approaches can be found in [17].

2.3.2 Machine Learning-based Approaches

Machine learning-based approaches applies the traditional machine learning algorithms in order to learn NE tagging decisions from manually annotated text. The most dominant machine learning techniques used for NER are the supervised learning techniques. These techniques include Hidden Markov Models (HMM) [18], Decision Trees [19], Maximum Entropy Models (ME) [20], Support Vector Machines (SVM) [21], and Conditional Random Fields (CRF) [22]. Here we will discuss the basics of HMM, CRF and SVM which will be used in this thesis.

Hidden Markov Models (HMM)

Hidden Markov Models (HMM) are generative models that proved to be very successful in a variety of sequence labeling tasks as Speech recognition, POS tagging, chunking, NER, etc. HMM is a finite state automaton with state transitions and symbol emissions (observations). The automaton models a probabilistic generative processes where a sequence of symbols is produced by starting from a start state, then transitioning to another state, emitting a symbol selected by that state, transitioning again, emitting a new symbol and so on until a final state is reached.

HMM-based classifier belongs to naive Bayes classifiers which are founded on a joint probability maximization of observation and state (label) sequences. The goal of HMM is to find the optimal tag sequence $T = t_1, t_2, ..., t_n$ for a given word sequence $W = w_1, w_2, ..., w_n$ that maximizes:

$$P(T \mid W) = \frac{P(T)P(W \mid T)}{P(W)} \tag{2.1}$$

where $P(W)$ is the same for all candidate tag sequences. $P(T)$ is the probability of the named entity (NE) tag. It can be calculated by Markov assumption which states that the probability of a tag depends only on a fixed number of previous NE tags. Here, in this work, we used $n = 4$. So, the probability of a NE tag depends on three previous tags, and then we have,

$$\begin{aligned} P(T) = P(t_1) \times P(t_2 \mid t_1) \times P(t_3 \mid t_1, t_2) \times P(t_4 \mid t_1, t_2, t_3) \times \ldots \\ \times P(t_n \mid t_{n-3}, t_{n-2}, t_{n-1}) \end{aligned} \tag{2.2}$$

As the relation between a word and its tag depends on the context of the word, the probability of the current word depends on the tag of the previous word and the tag to be assigned to the current word. So $P(W|T)$ can be calculated as:

$$P(W \mid T) = P(w_1 \mid t_1) \times P(w_2 \mid t_1, t_2) \times \ldots \times P(w_n \mid t_{n-1}, t_n) \tag{2.3}$$

The prior probability $P(t_i|t_{i-3}, t_{i-2}, t_{i-1})$ and the likelihood probability $P(w_i|t_i)$ can be estimated from training data. Given a model and all its parameters, named entity recognition is performed by determining the sequence of states that was most likely to have generated the entire document, and extracting the symbols that were associated with target states. To perform extraction, Viterbi algorithm [23] is used for finding the most likely state sequence given a HMM model and a sequence of symbols. Viterbi algorithm is a dynamic programming solution that solves the problem in just $O(MN^2)$ time, where M is the length of the sequence and N is the number of states in the model.

Conditional Random Fields (CRF)

HMMs have difficulty with modeling overlapped, non-independent features of the output part-of-speech tag of the word, the surrounding words, and capitalization patterns. Conditional Random Fields (CRF) can model these overlapping, non-independent features [24]. The linear chain CRF is simplest model of CRF. It defines the conditional probability:

$$P(T \mid W) = \frac{\exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j f_j\left(t_{i-1}, t_i, W, i\right)\right)}{\sum_{t,w}\exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j f_j\left(t_{i-1}, t_i, W, i\right)\right)} \tag{2.4}$$

where f is set of m feature functions, λ_j is the weight for feature function f_j, and the denominator is a normalization factor that ensures the distribution p sums to 1. This normalization factor is called the *partition function*. The outer summation of the *partition function* is over the exponentially many possible assignments to t and w. For this reason, computing the *partition function* is intractable in general, but much work exists on how to approximate it [25].

The feature functions are the main components of CRF. The general form of a feature function is $f_j\left(t_{i-1}, t_i, W, i\right)$, which looks at tag sequence T, the input sequence W, and the current location in the sequence (i).

Here are some examples for features that could be used with CRF:

- The tag of the word.

- The position of the word in the sentence.
- The part of speech tag of the word.
- The shape of the word (Capitalization/Small state, Digits/Characters, etc.).
- The suffix and the prefix of the word.

An example for a feature function which produces a binary value for the current word shape is $Capitalized$:

$$f_i\left(t_{i-1}, t_i, W, i\right) = \begin{cases} 1 & \text{if } w_i \text{ is Capitalized} \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

The training process involves finding the optimal values for the parameters λ_j that maximize the conditional probability $P(T \mid W)$. The standard parameter learning approach is to compute the stochastic gradient descent of the log of the objective function:

$$\frac{\partial}{\partial \lambda_k} \sum_{i=1}^{n} \log p(t_i|w_i)) - \sum_{j=1}^{m} \frac{\lambda_j^2}{2\sigma^2} \tag{2.6}$$

where the term $\sum_{j=1}^{m} \frac{\lambda_j^2}{2\sigma^2}$ is a Gaussian prior on λ to regularize the training.

Support Vector Machines (SVM)

Support Vector Machines (SVM) is a relatively new class of machine learning techniques first introduced in 1995 [26] and has been used for NER in 2003 [21].

Based on the structural risk minimization principle from the computational learning theory, SVM seeks a decision surface to separate the training data points into two classes and makes decisions based on the support vectors that are selected as the only effective elements in the training set. Given a set of N linearly separable points, $S = x_i \in R^n \mid i = 1, 2, ..., N$, each point x_i belongs to one of the two classes, labeled as $y_i \in -1, +1$. A separating hyper-plane divides S into 2 sides, each side containing points with the same class label only. The separating hyper-plane can be identified by the pair (w, b) that satisfies:

$$w \cdot x + b = 0 \tag{2.7}$$

$$\text{and } \begin{cases} w \cdot x_i + b \geq +1 \text{ if } y_i = +1 \\ w \cdot x_i + b \leq -1 \text{ if } y_i = -1 \end{cases}$$

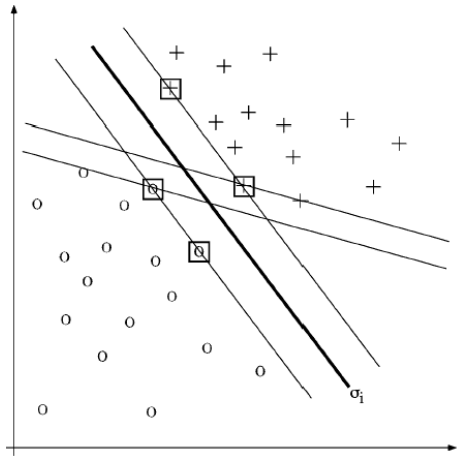


Figure 2.3: Learning Support Vector Machine.

for $i = 1, 2, \ldots, N$; where the dot product operation $(\cdot)$ is defined by:

$$w \cdot x = \sum_i w_i x_i \tag{2.8}$$

for vectors w and x. Thus the goal of the SVM learning is to find the optimal separating hyper-plane (OSH) that has the maximal margin to both sides. This can be formulated as:

$$\text{minimize } \frac{1}{2}||w||^2 \tag{2.9}$$

$$\text{subject to } \begin{cases} w \cdot x_i + b \geq +1 \text{ if } y_i = +1 \\ w \cdot x_i + b \leq -1 \text{ if } y_i = -1 \end{cases} \text{ for } i = 1, 2, \ldots, N$$

Figure 2.3 shows how SVM finds the OSH. The small crosses and circles in figure 2.3 represent positive and negative training examples, respectively, whereas lines represent decision surfaces. Decision surface σ_i (indicated by the thicker line) is, among those shown, the best possible one, as it is the middle element of the widest set of parallel decision surfaces (i.e., its minimum distance to any training example is the maximum). Small boxes indicate the support vectors.

During classification, SVM makes decision based on the OSH instead of the whole training set. It simply finds out on which side of the OSH the test pattern is located. This property makes SVM highly competitive, compared with

other traditional classification methods, in terms of computational efficiency and predictive accuracy [27].

SVM was introduced to the NER problem since 2003 [21]. The classifier tries to predict the class of each token (word) in the text given set of features like affixes and suffixes, token shape features, dictionaries features, etc.

2.3.3 Toponyms Extraction

Few researches focused only on toponym extraction. In [28], a method for toponym recognition is presented that is tuned for streaming news by leveraging a wide variety of recognition components, both rule-based and statistical. The authors presented a comprehensive, multifaceted toponym recognition method designed for streaming news using many types of evidence, including: a dictionary of entity names and cue words; statistical methods including POS tagging and NER, with appropriate post processing steps; rule-based toponym refactoring; and grammar filters involving noun adjuncts and active verbs.

Another interesting toponym extraction work was done by Pouliquen et al. [29]. They present a multilingual method to recognize geographical references in free text that uses minimum of language-dependent resources, except a gazetteer. In this system, place names are identified exclusively through gazetteer look-up procedures and subsequent disambiguation or elimination.

2.3.4 Language Independence

Multilingual NER is discussed by many researchers. The first attention to this topic was made by the shared task of CoNLL-2002. The system of Carreras et al. [30] outperformed all other systems, both on the Spanish test data and the Dutch test data. The two main subtasks of the problem, extraction (NEE) and classification (NEC), were performed sequentially using binary AdaBoost classifiers. A window surrounding a word w represents the local context of w used by a classifier to make a decision on the word. A set of primitive features (like word shape, gazetteer features and left predictions) was applied to each word in the window. Features like words lemmas, the part of speech (POS) tags, the prefixes and suffixes and gazetteer information were used.

Similarly, Florian et al. [31] used classifier-combination experimental framework for multilingual NER in which four diverse classifiers (robust linear classifier, maximum entropy, transformation-based learning, and hidden Markov model) were combined under different conditions. Again a window

of surrounding words were used to train and test the system. Szarvas et al. [32] introduced a multilingual NER system by applying AdaBoostM1 and the C4.5 decision tree learning algorithm.

Other approaches investigated the benefits of Wikipedia parallel articles in different languages in the process of multilingual NER. Richman and Schone utilized the multilingual characteristics of Wikipedia to annotate a large corpus of text with NER tags [33]. Their aim was to pull back the decision-making process to English whenever possible, so that they could apply some level of linguistic expertise. To generate a set of training data in a given language, they selected a large number of articles from its Wikipedia. They used the explicit article links within the text. A search for an associated English language article is done, if available, for additional information. Then they checks for multi-word phrases that exist as titles of Wikipedia articles. Finally they used regular expressions to locate additional entities such as numeric dates.

Similarly, Nothman et al. [34] automatically created silver-standard multilingual training annotations for NER by exploiting the text and structure of parallel Wikipedia articles in different languages. First, they classified each Wikipedia article into NE types, then they transformed the links between articles into NE annotations by projecting the target article's classifications onto the anchor text.

2.3.5 Robustness

Robustness in NER systems is a major issue that researchers looked after. In [35], Robustness was proved by applying the approach onto English and German collections. The authors incorporated a large number of linguistic features. The conditional probability of each token's tag is estimated given the feature vector associated with that token. Features similar to those discussed before were used.

Arnold [36] studied learning transfer between different training and test domains. His goal was to train a model that will extract proteins names from unseen articles captions (target test domain), given labeled abstracts (source training domain). He explored the ways to relax assumptions and exploit regularities in order to solve this problem. He exploited the hierarchical relationship between lexical features, allowing for natural smoothing and sharing of information across features. Structural frequency features were developed to take advantage of the information contained in the structure of the data itself and the distribution of instances across that structure. He also studied leveraging the relationship of entities among themselves, across tasks and labels

within a dataset.

Rüd et al. [37] used search engine results to address a particularly difficult cross-domain NER task. Each token is provided as a query along with a window of context to the Google search engine. Specific features (like the mutual association between any word in the snippets and each entity class) were extracted from the snippet results. Their approach is shown to be robust to noise (spelling, tokenization, capitalization etc.) and to make optimal use of minimal context.

2.4 Named Entity Disambiguation

Named entity disambiguation (NED), also referred to as record linkage, entity Linking or entity resolution, involves aligning a textual mention of a named entity to an appropriate entry in a knowledge base, which may or may not contain the entity. Literature review for NED for different types of named entities is presented in chapter 6. Here, we only focus on toponym disambiguation approaches.

2.4.1 Toponyms Disambiguation

According to [38], there are different kinds of toponym ambiguity. One type is structural ambiguity, where the structure of the tokens forming the name are ambiguous (e.g., is the word '*Lake*' part of the toponym '*Lake Como*' or not?). Another type of ambiguity is semantic ambiguity, where the type of the entity being referred to is ambiguous (e.g., is '*Paris*' a toponym or a girl's name?). A third form of toponym ambiguity is reference ambiguity, where it is unclear to which of several alternatives the toponym actually refers (e.g., does '*London*' refer to a place in '*UK*' or in '*Canada*'?). In this work, we focus on the structural and the reference ambiguities.

Toponym reference disambiguation or resolution is a form of Word Sense Disambiguation (WSD). According to [39], existing methods for toponym disambiguation can be classified into three categories: (i) map-based: methods that use an explicit representation of places on a map; (ii) knowledge-based: methods that use external knowledge sources such as gazetteers, ontologies, or Wikipedia; and (iii) data-driven or supervised: methods that are based on machine learning techniques. An example of a map-based approach is [40], which aggregates all references for all toponyms in the text onto a grid with

weights representing the number of times they appear. References, with a distance more than two times the standard deviation away from the centroid of the name, are discarded.

Knowledge-based approaches are based on the hypothesis that toponyms appearing together in text are related to each other, and that this relation can be extracted from gazetteers and knowledge bases like Wikipedia. Following this hypothesis, [41] used a toponym's local linguistic context to determine the toponym type (e.g., river, mountain, city) and then filtered out irrelevant references by this type. Another example of a knowledge-based approach is [42] which uses Wikipedia to generate co-occurrence models for toponym disambiguation.

Supervised learning approaches use machine learning techniques for disambiguation. [43] trained a naive Bayes classifier on toponyms with disambiguating cues such as '*Nashville, Tennessee*' or '*Springfield, Massachusetts*', and tested it on texts without these clues. Similarly, [44] used Hidden Markov Models to annotate toponyms and then applied Support Vector Machines to rank possible disambiguations.

CHAPTER 3

The Reinforcement Effect

3.1 Summary

Natural language processing (NLP) tasks are commonly divided into set of pipelined sub tasks. The residual error produced in any sub task propagates, adversely affecting the end objectives. This is why we believe that back propagation would help improving the overall system quality. Named entity extraction (NEE) and disambiguation (NED) are two NLP subtasks that have received much attention in recent years. Although NEE and NED are highly dependent, almost no existing works examine this dependency. It is the aim of this chapter to present a proof of concept of their dependency and show how one affects the other, and vice versa. We conducted experiments with a set of descriptions of holiday homes with the aim to extract and disambiguate toponyms as a representative example of named entities. We experimented with a rule-based approach for extraction and three different approaches for disambiguation with the purpose to infer the country where the holiday home is located. We examined how the effectiveness of extraction influences the effectiveness of disambiguation, and reciprocally, how filtering out ambiguous names (an activity that depends on the disambiguation process) improves the effectiveness of extraction. Since this, in turn, may improve the effectiveness of disambiguation again, it shows that extraction and disambiguation may reinforce each other.

The contents of this chapter have been published as [45].

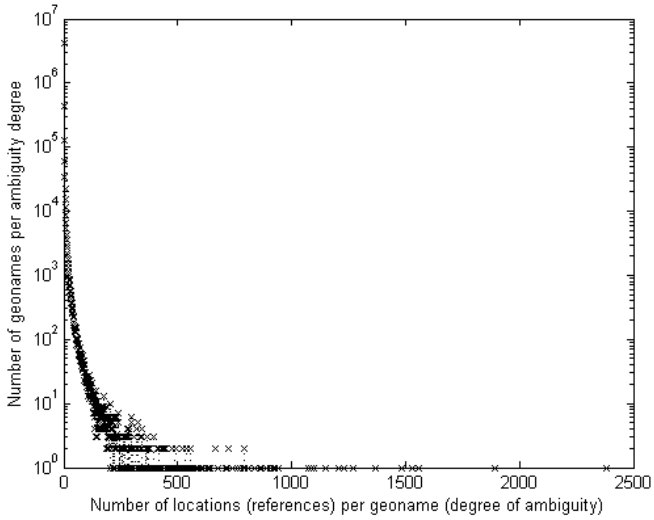


Figure 3.1: Toponym ambiguity in GeoNames: long tail.

3.2 Introduction

In natural language, toponyms, i.e., names for locations, are used to refer to these locations without having to mention the actual geographic coordinates. The process of toponym extraction (a.k.a. toponym recognition) is a sub task of information extraction that aims to identify location names in natural text. This process has become a basic step of many systems for Information Extraction (IE), Information Retrieval (IR), Question Answering (QA), and in systems combining these, such as [46].

Toponym disambiguation (a.k.a. toponym resolution) is the task of determining which real location is referred to by a certain instance of a name. Toponyms, as with named entities in general, are highly ambiguous. For example, according to GeoNames[1], the toponym '*Paris*' refers to more than sixty different geographic places around the world besides the capital of France. Figure 3.1 shows the long tail distribution of toponym ambiguity while figure 3.2 summarizes this distribution. It can be observed that around 46% of toponyms

[1] www.geonames.org

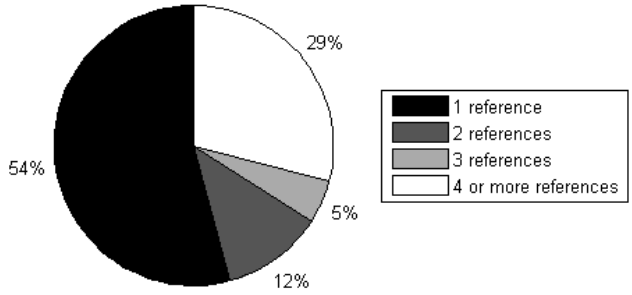


Figure 3.2: Toponym ambiguity in GeoNames: reference frequency distribution.

Table 3.1: Toponym ambiguity in GeoNames: top 10.

Geoname	Number of references
First Baptist Church	2382
The Church of Jesus Christ of Latter Day Saints	1893
San Antonio	1561
Church of Christ	1558
Mill Creek	1530
Spring Creek	1486
San José	1366
Dry Creek	1271
First Presbyterian Church	1229
Santa Rosa	1205

have two or more, 35% three or more, and 29% four or more references. Table 3.1 shows the top ten of the most ambiguous geographic names according to GeoNames gazetteer.

In natural language, humans rely on the context to disambiguate a toponym. Note that in human communication, the context used for disambiguation is broad: not only the surrounding text matters, but also the author and recipient, their background knowledge, the activity they are currently involved in, even the information the author has about the background knowledge of the recipient, and much more.

Although entity extraction and disambiguation are highly dependent, almost all efforts focus on improving the effectiveness of either one but not both. Hence, almost none examine their interdependency. It is the aim of this chap-

ter to examine exactly this. We studied not only the positive and the negative effect of the extraction process on the disambiguation process, but also the potential of using the result of disambiguation to improve extraction. We call this potential for mutual improvement, the *reinforcement effect* (see Figure 3.3).

To examine the *reinforcement effect*, we conducted experiments on a collection of holiday home descriptions from the Eurocottage[2] portal. These descriptions contain general information about the holiday home including its location and its neighborhood (See Figure 3.6 for some examples).

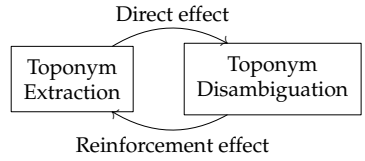


Figure 3.3: The *reinforcement effect* between the toponym extraction and disambiguation processes.

The task we focus on is to extract the toponyms from the description and use them to infer the country where the holiday property is located. We use country inference as a way to disambiguate the extracted toponyms. A set of heuristics have been developed to extract toponyms from the text. Three different approaches for toponym disambiguation are compared. We investigate how the effectiveness of disambiguation is affected by the effectiveness of extraction by comparing with results based on manually extracted toponyms. We investigate the reverse measuring, the effectiveness of extraction when filtering out those toponyms found to be highly ambiguous, and in turn, measure the effectiveness of disambiguation after filtering this set of highly ambiguous toponyms.

The rest of the chapter is organized as follows. Sections 3.3 and 3.4 present the approaches we used for toponym extraction and disambiguation respectively. In Section 3.6, we describe the experimental setup, present its results, and discuss some observations and their consequences. Finally, conclusions and future research directions are presented in Section 3.7.

3.3 Toponyms Extraction

3.3.1 GATE Toolkit

We use GATE [15] for toponym extraction. GATE (General Architecture for Text Engineering) is an open source framework developed at the University of Sheffield since 1995 for building systems that process human language [15]. GATE is distributed with an IE system called ANNIE (A Nearly New

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

Information Extraction system) which is composed of reusable processing components for common NLP tasks such as a tokenizer, sentence splitter, POS tagger, gazetteer, finite state transducer, orthomatcher, and coreference resolver. ANNIE relies on the JAPE language (Java Annotation Patterns Engine) for specifying rules for annotating phrases in text documents.

3.3.2 JAPE Rules

JAPE is a Java Annotation Patterns Engine. It provides finite state transduction over annotations based on regular expressions. It is a version of The Common Pattern Specifications Language (CPSL) [16].

A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left-hand-side (LHS) of the rules consist of an annotation pattern description. The right-hand-side (RHS) consists of annotation manipulation statements. Annotations matched on the LHS of a rule may be referred to on the RHS by means of labels that are attached to pattern elements. Consider the example in listing 3.1:

Listing 3.1: JAPE Rule Example.

```
Phase: Jobtitle
Input: Lookup
Options: control = appelt debug = true

Rule: Jobtitle1
(
 {Lookup.majorType == jobtitle}
 (
  {Lookup.majorType == jobtitle}
 )?
)
:jobtitle
-->
 :jobtitle.JobTitle = {rule = "JobTitle1"}
```

The LHS is the part preceding the '–>' and the RHS is the part following it. The LHS specifies a pattern to be matched to the annotated GATE document, whereas the RHS specifies what is to be done to the matched text. In this example, we have a rule entitled 'Jobtitle1', which will match text annotated with

a 'Lookup' annotation with a 'majorType' feature of 'jobtitle', followed optionally by further text annotated as a 'Lookup' with 'majorType' of 'jobtitle'. Once this rule has matched a sequence of text, the entire sequence is allocated a label by the rule, and in this case, the label is 'jobtitle'. On the RHS, we refer to this span of text using the label given in the LHS; 'jobtitle'. We say that this text is to be given an annotation of type 'JobTitle' and a 'rule' feature set to 'JobTitle1'.

JAPE grammar begins by giving it a phase name, e.g. 'Phase: Jobtitle'. JAPE grammars can be cascaded, and so each grammar is considered to be a 'phase'. It is also required to provide a list of the annotation types we will use in the grammar. In our example, we use 'Input: Lookup' because the only annotation type is used on the LHS are Lookup annotations. If no annotations are defined, all annotations will be matched.

Then, several options are set:

- Control; in this case, 'appelt'. This defines the method of rule matching.
- Debug. When set to true, if the grammar is running in Appelt mode and there is more than one possible match, the conflicts will be displayed on the standard output.

A wide range of functionality can be used with JAPE, making it a very powerful system. More details can be found in this tutorial [47].

3.3.3 Extraction Rules

For toponym extraction, we develop handcrafted rules for extraction as suggested in [48]. The rules are specified in GATE 's JAPE language. They are based on heuristics on the orthography features of tokens and other annotations. Listing 3.2 contains the toponym extraction rules used in our experiments.

Listing 3.2: Toponyms extraction JAPE rules.

```
Phase:firstpass
Input:  Token Split Lookup
Options: control = appelt
Rule: LocationsTokens
Priority: 10
(
 (
  ({Token,!Token.string==":",!Token.kind=="number",!Token.string==".",!Split})
  (

   ({Token.orth == upperInitial,!Lookup.majorType=="date"})[1,2]
   (
    ({Token.string == "-"})[0,1]
   )

   ({Token.orth == upperInitial,!Lookup.majorType=="date"})[0,2]
  ):toponym_1
 )
 |
 (
  ({Token,!Token.string==":",!Token.kind=="number",!Token.string==".",!Split})
  (

   ({Token.orth == upperInitial,!Lookup.majorType=="date"})[1,2]
   (
    ({Token.string == "-"})[0,1]
    |
```

```
    ({Token.orth == lowercase, Token.string!="and",Token.length<=3})[0,1]
   )

   ({Token.orth == upperInitial,!Lookup.majorType=="date"})[1,2]
  ):toponym_2
)
|
(
 (
  ({Token,Token.string==":"})
  |
  ({Token,Token.string=="."})
  |
  ({Split})
 )
 (
  ({Token.orth == upperInitial,!Lookup.majorType=="date"})[1,2]
  (
   ({Token.string == "-"})[0,1]
   |
   ({Token.orth == lowercase, Token.string!="and",Token.length<=3})[0,1]
  )

  ({Token.orth == upperInitial,!Lookup.majorType=="date"})[1,2]
 ):toponym_3
)
|
(
```

```
  ({Split})
  (
   ({Token.orth == upperInitial,!Lookup.majorType=="date"})
   ({Token.string == "-"})[0,1]
   ({Token.orth == upperInitial,!Lookup.majorType=="date"})[1,2]
  ):toponym_4
 )
 |
 (
  ({Token.string=~"(of|from|at|to|near)"})
  (
   ({Token.orth == upperInitial,!Lookup.majorType=="date"})
   ({Token.string == "-"})[0,1]
   ({Token.orth == upperInitial,!Lookup.majorType=="date"})[1,2]
  ):toponym_5
 )
 |
 (
  ({Token.string=~"(\"|\')"})
  (
   ({Token.orth == upperInitial,!Lookup.majorType=="date"})
   ({Token.string == "-"})[0,1]
   ({Token.orth == upperInitial,!Lookup.majorType=="date"})[1,2]
  ):toponym_6
  ({Token.string=~"(\"|\')"})
 )
)
-->
```

```
:toponym_1.LocationsTokens={rule=LocationsTokens_1},
:toponym_2.LocationsTokens={rule=LocationsTokens_2},
:toponym_3.LocationsTokens={rule=LocationsTokens_3},
:toponym_4.LocationsTokens={rule=LocationsTokens_4},
:toponym_5.LocationsTokens={rule=LocationsTokens_5},
:toponym_6.LocationsTokens={rule=LocationsTokens_6}
```

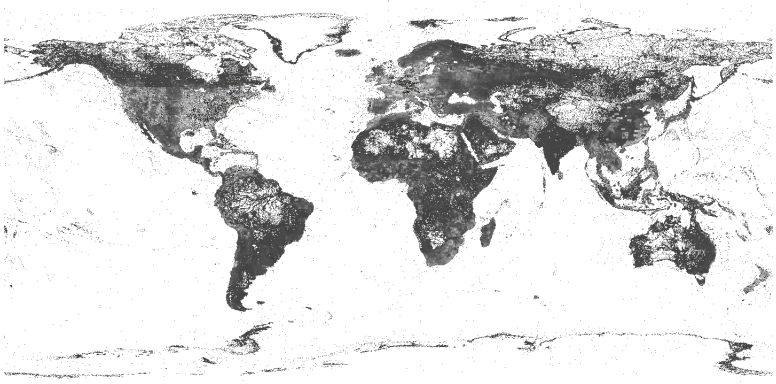


Figure 3.4: The world map drawn with the GeoNames longitudes and latitudes.

3.3.4 Entity matching

We use the GeoNames geographical database for entity matching. It consists of 7.5 million unique entities of which 2.8 million are populated places with in total 5.5 million alternative names. All entities are categorized into 9 classes defining the type of place (e.g., country, region, lake, city, and road). Figure 3.4 shows the coverage of GeoNames as a map drawn by placing a point at the coordinates of each entity.

3.4 Toponyms Disambiguation

We compare three approaches for toponym disambiguation, one representative example for each of the categories described in Section 2.4.1. All require the text to contain toponym annotations. Hence, disambiguation can be seen as a classification problem assigning the toponyms to their most probable country. The notation we used for describing the approaches can be found in Table 3.2.

3.4.1 Bayes Approach

This is a supervised learning approach for toponym disambiguation based on Naive Bayes (NB) theory. NB is a probabilistic approach widely used for text classification. It uses the joint probabilities of terms and categories to estimate the probabilities of categories given a document [49]. It is naive in the

Table 3.2: Notation used for describing the toponym disambiguation approaches.

D	the set of all documents. $D = \{d_l \in D \mid l = 1 \dots n\}$
T	the set of toponyms appearing in the document d. $T = \{t_i \in d \mid i = 1 \dots m\}$
G	GeoNames gazetteer. $G = \{r_{ix} \mid r_{ix}$ is geographical location$\}$ Where i is the toponym index and x is the reference index. Each reference r_{ix} is represented by a set of characteristics: its country, longitude, latitude, and its class. r_{ix} is a reference for t_i, if t_i is string-wise equal to r_{ix} or one of its alternatives.
$R(t_i)$	the set of references for toponym t_i. $R(t_i) = \{r_{ix} \in G \mid t_i$ is string-wise equal to r_{ix} or to one of its alternatives$\}$
R	the set of all sets $R(t_i)$. $\forall t_i \in T$.
C_i	the set of countries of $R(t_i)$. $C_i = \{c_{ix} \mid c_{ix}$ is the country of the reference $r_{ix}\}$

sense that it makes the assumption that all terms are conditionally independent of each other given a category. Because of this independence assumption, the parameters for each term can be learned separately which simplifies and speeds up computations compared to non-naive Bayes classifiers. Toponym disambiguation can be seen as a text classification problem where extracted toponyms are considered as terms and the country associated with the text as a class.

There are two common event models for NB text classification: the multinomial and multivariate Bernoulli model [50]. Here, we use the multinomial model as suggested by the same reference. In both models, classification of toponyms is performed by applying Bayes' rule:

$$P(C = c_j \mid d_i) = \frac{P(d_i \mid c_j)P(c_j)}{P(d_i)} \tag{3.1}$$

where d_i is a test document (as a list of extracted toponyms) and c_j is a country. We assign that country c_j to d_i that has the highest $P(C = c_j \mid d_i)$, i.e., the highest posterior probability of country c_j given test document d_i. To be able to calculate $P(C = c_j \mid d_i)$, the prior probability $P(c_j)$ and the likelihood $P(d_i \mid c_j)$ have to be estimated from a training set. Note that the evidence $P(d_i)$ is the same for each country, so we can eliminate it from the computation. The prior probability for countries, $P(c_j)$, can be estimated as follows:

$$P(c_j) = \frac{\sum_{i=1}^{N} y(d_i, c_j)}{N} \tag{3.2}$$

where N is the number of training documents and $y(d_i, c_j)$ is defined as:

$$y(d_i, c_j) = \begin{cases} 1 & \text{if } d_i \in c_j \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

So, the prior probability of country c_j is estimated by the fraction of documents in the training set belonging to c_j. $P(d_i \mid c_j)$ parameters are estimated using the multinomial model. In this model, a document d_i is a sequence of extracted toponyms. The Naive Bayes assumption is that the probability of each toponym is independent of its context, position, and length of the document. So, each document d_i is drawn from a multinomial distribution of toponyms with a number of independent trials equal to the length of d_i. The likelihood probability of a document d_i given its country c_j can hence be approximated as:

$$P(d_i \mid c_j) = P(t_1, t_2, \ldots, t_n \mid c_j) \approx \prod_{k=1}^{n} P(t_k \mid c_j) \tag{3.4}$$

where n is the number of toponyms in document d_i, and t_k is the k^{th} toponym occurring in d_i. Thus, the estimation of $P(d_i \mid c_j)$ is reduced to estimating each $P(t_k \mid c_j)$ independently. $P(t_k \mid c_j)$ can be estimated with Laplacian smoothing:

$$P(t_k \mid c_j) = \frac{\Theta + tf_{kj}}{(\Theta \times |T|) + \sum_{l=1}^{|T|} tf_{lj}} \tag{3.5}$$

where tf_{kj} is the term frequency of toponym t_k belonging to country c_j. The summation term in the denominator stands for the total number of toponym occurrences belonging to c_j. Θ in the numerator and $\Theta \times |T|$ in the denominator are used to avoid zero probabilities. Θ is set to 0.0001 according to [51].

Using this approach, all the Bayes parameters for classifying a test document to its associated country can be estimated using a training set.

3.4.2 Popularity Approach

This is an unsupervised approach based on the intuition that, as each toponym in a document may refer to many alternatives, the more of those appear in a

Table 3.3: The feature classes of GeoNames along with the weights we use for each class.

GeoNames Feature Classes (GFC)	**Weight** $wgfc$
Administrative Boundary Features	3
Hydrographic Features	1
Area Features	1
Populated Place Features	3
Road / Railroad Features	1
Spot Features	1
Hypsographic Features	1
Undersea Features	1
Vegetation Features	1

certain country, the more probable it is that the document belongs to that country. For example, it is common to find lakes, rivers or mountains with the same name as a neighboring city. We also take into consideration the GeoNames Feature Class (GFC) of the reference. As shown in Table 3.3, we assign a weight to each of the 9 GFCs representing its contribution to the country of the toponym, basically choosing a higher weight for cities, populated places, regions, etc. We define the *popularity* of a country c for a certain document d to be the average over all toponyms of d of the sum of the weights of the references of those toponyms in c:

$$Pop_d(c) = \frac{1}{|d|} \sum_{t_i \in d} \sum_{r_{ix} \in R(t_i)]c} wgfc(r_{ix}) \tag{3.6}$$

where $R(t_i)]c = \{r_{ix} \in R(t_i) \mid c_{ix} = c\}$ is the restriction of the set of references $R(t_i)$ to those in country c, and $wgfc$ is the weight of the GeoNames Feature Class as specified in Table 3.3. For disambiguating the country of a document, we choose the country with the highest popularity.

3.4.3 Clustering Approach

The clustering approach is an unsupervised disambiguation approach based on the assumption that toponyms appearing in same document are likely to refer to locations close to each other *distance-wise*. For our holiday home descriptions, it appears quite safe to assume this. For each toponym t_i, we have, in general, multiple entity candidates. Let $R(t_i) = \{r_{ix} \in$ GeoNames gazetteer$\}$

be the set of reference candidates for toponym t_i. Additionally each reference r_{ix} in GeoNames belongs to a country c_j. By taking one entity candidate for each toponym, we form a cluster. A cluster, hence, is a possible combination of entity candidates, or in other words, one possible entity candidate of the toponyms in the text. In this approach, we consider all possible clusters, compute the average distance between the candidate locations in the cluster, and choose the cluster $Cluster_{min}$ with the lowest average distance. We choose the most often occurring country in $Cluster_{min}$ for disambiguating the country of the document. In effect, the abovementioned assumption states that the entities that belong to $Cluster_{min}$ are the true representative entities for the corresponding toponyms as they appeared in the text. Equations 3.7 through 3.11 show the steps of the described disambiguation procedure.

$$Clusters = \{\{r_{1x}, r_{2x}, \ldots, r_{mx}\} \mid \forall t_i \in d \bullet r_{ix} \in R(t_i)\} \tag{3.7}$$

$$Cluster_{min} = \underset{Cluster_k \in Clusters}{\arg\min} \text{ average distance of } Cluster_k \tag{3.8}$$

$$Countries_{min} = \{c_j \mid r_{ix} \in Cluster_{min} \wedge r_{ix} \in c_j\} \tag{3.9}$$

$$c_{winner} = \underset{c_j \in Countries_{min}}{\arg\max} \; freq(c_j) \tag{3.10}$$

where

$$freq(c_j) = \sum_{i=1}^{n} \begin{cases} 1 & \text{if } r_{ix} \in c_j \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

Illustrative Example To illustrate our clustering approach, we plot the candidate references of the toponyms of the holiday property description shown in figure 3.6c. Figures 3.5a and 3.5b show the candidate references of each toponym with a different color. For example, the candidates of the toponym '*Steinbach*' have red color. The correct reference of the mentioned toponyms are characterized with a dotted icon. The cluster $Cluster_{min}$ is shown with an oval in figure 3.5b. We can see that $Cluster_{min}$ contains all the correct representatives of the mentioned toponyms. Given the candidates belonging to $Cluster_{min}$, we could easily infer '*Belgium*' to be the c_{winner} of that property.

(a)

(b)

Figure 3.5: Map plot of candidate entities for toponym of property description shown in figure 3.6c.

3.5 The Reinforcement Effect

Examining the results of disambiguation, we discovered that there were many false positives among the automatically extracted toponyms, i.e. words extracted as a toponym and having a reference in GeoNames, that are in fact not toponyms. These words affect the disambiguation result, because the matching entries in GeoNames belong to many different countries. A possible improvement for the extraction process, hence, is filtering out extracted toponyms that belong to documents (properties descriptions) that have been classified under different countries. The intuition is that these toponyms, whether they are actual toponyms in reality or not, confuse the disambiguation process.

3.6 Experimental Results

In this section, we present the results of experiments with the presented methods of extraction and disambiguation applied on a collection of holiday properties descriptions. The goal of the experiments is to investigate the influence of extraction effectiveness on disambiguation effectiveness and vice versa, and ultimately to show that they can *reinforce* each other.

3.6.1 Dataset

The dataset we use for our experiments is a collection of traveling agent holiday properties descriptions from the Eurocottage portal. The descriptions not only contain information about the property itself and its facilities, but also a description of its location, neighboring cities and opportunities for sightseeing. The dataset includes the country of each property which we use to validate our results. We consider this type of text as a semi-formal text. It is not formal as it lacks a proper English sentence structure of subject, verb and object. And also it is not totally informal as it preserves same style of writing for describing the properties contents and neighborhood. Figure 3.6 shows three examples for a holiday property descriptions.

2-room apartment 55 m2: living/dining room with 1 sofa bed and satellite-TV, exit to the balcony. 1 room with 2 beds (90 cm, length 190 cm). Open kitchen (4 hotplates, freezer). Bath/bidet/WC. Electric heating. Balcony 8 m2. Facilities: telephone, safe (extra). Terrace Club: Holiday complex, 3 storeys, built in 1995 2.5 km from the centre of **Armacao de Pera**, in a quiet position. For shared use: garden, swimming pool (25 x 12 m, 01.04.-30.09.), paddling pool, children's playground. In the house: reception, restaurant. Laundry (extra). Linen change weekly. Room cleaning 4 times per week. Public parking on the road. Railway station "**Alcantarilha**" 10 km. Please note: There are more similar properties for rent in this same residence. Reception is open 16 hours (0800-2400 hrs). Lounge and reading room, games room. Daily entertainment for adults and children. Bar-swimming pool open in summer. Restaurant with Take Away service. Breakfast buffet, lunch and dinner(to be paid for separately, on site). Trips arranged, entrance to water parks. Car hire. Electric cafetiere to be requested in adavance. Beach football pitch. IMPORTANT: access to the internet in the computer room (extra). The closest beach (350 m) is the "**Sehora da Rocha**", **Playa de Armacao de Pera** 2.5 km. Please note: the urbanisation comprises of eight 4 storey buildings, no lift, with a total of 185 apartments. Bus station in **Armacao de Pera** 4 km.

(a) Example 1.

Bargecchia 9 km from **Massarosa**: nice, rustic house "**I Cipressi**", renovated in 2000, in the center of **Bargecchia** 11 km from the center of **Viareggio**, 29 km from the center of **Lucca**, in a central, quiet, sunny position on a slope. Private, terrace (60 m2), garden furniture, barbecue. Steep motor access to the house. Parking in the grounds. Grocers, restaurant, bar 100 m, sandy beach 11 km. Please note: car essential.
3-room house 90 m2 on 2 levels, comfortable and modern furnishings: living/dining room with 1 double sofa bed, open fireplace, dining table and TV, exit to the terrace. Kitchenette (oven, dishwasher, freezer). Shower/bidet/WC. Upper floor: 1 double bedroom. 1 room with 1 x 2 bunk beds, exit to the balcony. Bath/bidet/WC. Gas heating (extra). Small balcony. Terrace 60 m2. Terrace furniture, barbecue. Lovely panoramic view of the sea, the lake and the valley. Facilities: washing machine. Reserved parking space n 2 fenced by the house. Please note: only 1 dog accepted.

(b) Example 2.

Le Doyen cottage is the oldest house in the village of **Steinbach** (built in 1674). Very pleasant to live in, it is situated right in the heart of the **Ardennes**. Close to **Robertville** and **Butchembach**, five minutes from the ski slopes and several lakes.

(c) Example 3.

Figure 3.6: Examples of EuroCottage holiday home descriptions (toponyms in bold).

Table 3.4: Effectiveness of the extraction rules.

Ground truth	Precision	Recall
Full ground truth	72%	78%
Matching ground truth	51%	80%

The dataset consists of 29707 property descriptions associated with the country where they are located. This set has been partitioned into a *training set* of 26610 descriptions for the Bayes supervised approach, and a *test set* containing the remaining 3097 descriptions. The *annotation test set* is a subset of the test set containing 790 descriptions for which we constructed a ground truth by manually annotating all toponyms.

It turned out, however, that not all manually annotated toponyms had a match in the GeoNames database. For example, we annotated phrases like 'Columbus Park' as a toponym, but no entry for this toponym in GeoNames exists. Therefore, we constructed, besides this *full ground truth*, also a *matching ground truth* where all non-matching annotations have been removed.

3.6.2 Initial Effectiveness of Extraction

The objective of the first set of experiments is to evaluate the initial effectiveness of the extraction rules in terms of precision and recall.

Table 3.4 contains the precision and recall of the extraction rules on the annotation test set evaluated against both ground truths. As expected, recall is higher with the matching ground truth, because there are less toponyms to find. Precision is lower, because some true positive toponyms don't have match with GeoNames and hence, are not in the matching ground truth. Furthermore, almost all the false positives extracted toponyms have a matching references in GeoNames.

3.6.3 Initial Effectiveness of Disambiguation

The second set of experiments aims to evaluate the initial effectiveness of the proposed disambiguation approaches and its sensitivity to the effectiveness of the extraction process.

The top part of Table 3.5 contains the precision of country disambiguation, i.e., the percentage of correctly inferred countries using the automatically annotated toponyms. As expected, the supervised approach performs better than

Table 3.5: Precision of country disambiguation.

	Bayes approach	Popularity approach	Clustering approach
On full test set			
Automatically extracted toponyms	94.2%	65.45%	78.19%
On annotation test set			
Automatically extracted toponyms	-	65.4%	78.95%
Manually annotated toponyms	-	75.6%	86%

access	attention	beach	breakfast	chalet	cottage	double
during	floor	garden	golf	holiday	haus	kitchen
market	olympic	panorama	resort	satellite	shops	spring
thermal	villa	village	wireless	world	you	

Figure 3.7: A sample of false positives among extracted toponyms.

both unsupervised approaches.

The bottom part of Table 3.5 aims at showing the influence of the imprecision of the extraction process on the disambiguation process. We compare the disambiguation results of using the automatically extracted toponyms versus the results of using the (better quality) manually annotated toponyms. Bayes approach was not applicable on the annotation test set as the number of documents is not enough to give good probability estimations. We can observe that the results for the automatically extracted toponyms are very similar to those of the full test set, hence we assume that our conclusions are also valid for the test set. We can conclude that both unsupervised approaches significantly benefit from better quality toponyms.

3.6.4 The Reinforcement Effect

Examining the results of disambiguation, we discovered that there were many false positives among the automatically extracted toponyms. A sample of such words is shown in figure 3.7.

These words affect the disambiguation result, because the matching entries in GeoNames belong to different countries. For example, 'Breakfast' may refer to an island in 'Marshall Islands', 'You' may refer to a place in 'Burkina Faso', and 'Double' may refer to an island in 'Antarctica'. To get rid of such false

Table 3.6: Effectiveness of the extraction rules after filtering.

Ground truth	Precision	Recall
Full ground truth	74%	77%
Matching ground truth	53%	79%

Table 3.7: Precision of country disambiguation with filtering.

	Popularity approach	Clustering approach
On annotation test set		
Filtered automatically extracted toponyms	73.5%	84.1%

positives, we filter out those extracted toponyms that belong to documents (properties descriptions) that have been classified under different countries. The intuition is that these toponyms, whether they are actual toponyms in reality or not, confuse the disambiguation process. We set the threshold to five, i.e. words classified under more than five countries in the properties descriptions are filtered out from the extracted toponyms. In this way, 197 toponyms were filtered out. In this experiment, we used the clustering approach for disambiguation.

Note that we used the result of disambiguation for an improvement of extraction. Therefore, this is an example of the *reinforcement effect* in figure 3.3.

To evaluate the effect of this improvement, we repeated the previous experiments but this time by using the set of automatically extracted toponyms after filtration. Tables 3.6 and 3.7 present the repetition of the first and second experiment, respectively.

Comparing Tables 3.6 and 3.4, we can observe a relatively small improvement in the extraction precision by filtering out the 'confusing' words with the cost of some loss in the recall. Nevertheless, if we compare tables 3.7 and 3.5, we observe a significant improvement for the subsequent disambiguation results.

This shows our claim that extraction and disambiguation may reinforce each other. In the next section, we explore this idea somewhat further by presenting observations from deeper analysis and discussing possible ways of exploiting the *reinforcement effect*.

3.6.5 Further Analysis and Discussion

From further analysis of results and causes, we like to mention the following observations and thoughts.

Ambiguous toponyms: The improvement described above was based on filtering out toponyms that belongs to descriptions classified under five or more different countries. The intuition was that these terms ordinarily do not constitute toponyms but general terms that happen to be common topological names as well, such as those of figure 3.7. In total, 197 extracted toponyms were filtered out in this way. We have observed, however, that some of these were in fact true toponyms, for example, '*Amsterdam*', '*France*', and '*Sweden*'. Apparently, these toponyms appear in more than five countries. We believe, however, that filtering them out, had a positive effect anyway as they were harming the disambiguation process.

Multi-token toponyms: Sometimes the structure of the terms constituting a toponym in the text is ambiguous. For example, for '*Lake Como*' it is dubious whether or not '*Lake*' is part of the toponym or not. In fact, it depends on the conventions of the gazetteer which choice produces the best results. Furthermore, some toponyms have a rare structure, such as '*Lido degli Estensi*'. The extraction rules of listing 3.2 failed to extract this as one toponym and instead produced two toponyms: '*Lido*' and '*Estensi*' with harmful consequences for the holiday home country disambiguation.

All-or-nothing: Related to this, we can observe that entity extraction is ordinarily an all-or-nothing activity: one can only annotate either '*Lake Como*' or '*Como*', but not both.

Near-border ambiguity: We also observed problems with near-border holiday homes, because their descriptions often mention places across the border. For example, the description in figure 3.8 has 4 toponyms in The Netherlands, 5 in Germany and 1 in the UK, whereas the holiday home itself is in The Netherlands and not in Germany. Even if an approach like the clustering approach is successful in interpreting the correct references of toponyms, it may still assign the wrong country.

Non-expressive toponyms: Finally, we observed some properties with no or non-expressive toponyms, such as '*North Sea*'. In such cases, it remains hard and error prone to correctly disambiguate the country of the holiday home.

Proposed new approach based on uncertain annotations: We believe that many of the observed problems are caused by an improper treatment of the inherent ambiguities. Natural language has the innate property that it is multiply interpretable. Therefore, none of the processes in information extraction

This charming holiday home is in a small holiday park in the village of **Nutter**^{NL}. The village is in the province of **Overijssel**^{NL}. The holiday home is comfortably furnished and equipped with every modern convenience.
The home is furnished in an **English**^{UK} style and has a romantic atmosphere. You can relax on the veranda in the evenings and enjoy delightful views of the orchard. The surrounding area has much to offer.
There are plenty of excellent walking and cycling routes. Interesting towns such as **Ootmarsum**^{NL} and **Almelo**^{NL} are well worth a visit. Children will enjoy the **German**^{GER} Animal Park in **Nordhorn**^{GER}. If you're prepared to travel a little further afield, you can reach the **Apfelkorn Distillery**^{GER} in **Haselüne**^{GER} in **Germany**^{GER}, in around one hour. It's not to be missed.

Figure 3.8: Example holiday home description illustrating the vulnerability of the clustering approach for near-border homes. '$\mathbf{t}^{c}$' depicts a toponym t in country c.

should be 'all-or-nothing'. In other words, all steps, including entity recognition, should produce *possible* alternatives with associated likelihoods and dependencies. Multiple iterations of recognition, matching, and disambiguation are then aimed at adjusting likelihoods and expanding or reducing alternatives (see figure 3.9).

As we have shown in this chapter, steps in the information extraction process can reinforce each other. With 'uncertain alternatives', reinforcement techniques such as refining extraction rules, establishing lists of exceptional cases, or even learning rules, can be more gradual and refined. One can imagine, for example, that it can be *automatically and gradually learned* that '*Lake Como*' is more likely to be the best naming convention rather than '*Como*', or that '*degli*' may connect two terms into one toponym, or that for country disambiguation, what threshold to use for the number of alternative countries above which such toponyms start to harm the disambiguation process. In this way, the entire process becomes more robust against ambiguous situations and can gradually learn. In other words, we believe there is much potential in making the inherent uncertainty in information extraction explicit.

3.7 Conclusions and Future Directions

Named entity extraction and disambiguation are highly dependent processes. The aim of this chapter is to provide a simple proof of concept to examine this dependency and show how one affects the other, and vice versa. Experiments

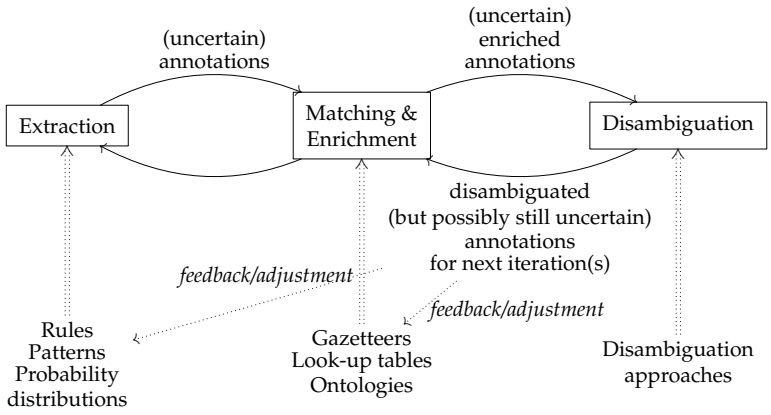


Figure 3.9: Activities and propagation of uncertainty.

were conducted with a set of descriptions of holiday homes with the aim to extract and disambiguate toponyms as a representative example of named entities. Three approaches for disambiguation were applied with the purpose to infer the country where the holiday home is located. We examined how the effectiveness of extraction influences the effectiveness of disambiguation, and reciprocally, how the result of disambiguation can be used to improve extraction. As an example of the latter we filtered out toponyms that were discovered to be highly ambiguous. Results showed that the effectiveness of extraction and, in turn, disambiguation improved, thereby showing that both can reinforce each other. We also analyzed the results more closely and formulated a general approach based on *uncertain annotation* for which we argue that it has much potential for making information extraction more robust against ambiguous situations and allowing it to gradually learn.

For next chapter, we plan to investigate the above mentioned potential. We also plan to examine statistical techniques for extraction, matching, and disambiguation as they seem to fit well in such an approach based on uncertain annotations.

CHAPTER 4

Improving Disambiguation by Iteratively Enhancing Certainty of Extraction

4.1 Summary

In the previous chapter, we presented a simple proof of concept to show the interdependency between toponym extraction and disambiguation and how one affects the other, and vice versa. In this chapter, we address the problem that existing disambiguation techniques mostly take as input the extracted named entities without considering the uncertainty and imperfection of the extraction process. We aim to investigate how handling the uncertainty of annotation has much potential for making both extraction and disambiguation more robust. For this purpose we use probabilistic extraction approaches. We show that the extraction confidence probabilities are useful in enhancing the effectiveness of disambiguation. Reciprocally, retraining the extraction models with negative samples automatically derived from the disambiguation results, improves the extraction models. This mutual reinforcement is shown to even have an effect after several automatic iterations.

The contents of this chapter have been published as [52].

4.2 Introduction

The general principle in our work is our conviction that named entity extraction and disambiguation are highly dependent. In the previous chapter, we studied not only the positive and negative effect of the extraction process on the disambiguation process, but also the potential of using the result of disambiguation to improve extraction. We called this potential for mutual improve-

ment, the *reinforcement effect*.

To examine the *reinforcement effect*, we conducted experiments on a collection of holiday home descriptions from the EuroCottage portal. These descriptions contain general information about the holiday home including its location and its neighborhood. As a representative example of toponym extraction and disambiguation, we focused on the task of extracting toponyms from the description and using them to infer the country where the holiday property is located.

In general, we concluded that many of the observed problems are caused by an improper treatment of the inherent ambiguities. Natural language has the innate property that it is multiply interpretable. Therefore, none of the processes in information extraction should be 'all-or-nothing'. In other words, all steps, including entity recognition, should produce possible alternatives with associated likelihoods and dependencies.

In this chapter, we focus on this principle. We turned to probabilistic approaches for toponym extraction. We choose to use HMM and CRF to build probabilistic models for extraction. The advantage of probabilistic techniques for extraction is that they provide alternatives for annotations along with confidence probabilities (confidence for short). Instead of discarding these, as is commonly done by selecting the top-most likely candidate, we use them to enrich the knowledge for disambiguation. The probabilities proved to be useful in enhancing the disambiguation process. We believe that there is much potential in making the inherent uncertainty in information extraction explicit in this way. For example, phrases like '*Lake Como*' and '*Como*' can be both extracted with different confidence. This restricts the negative effect of differences in naming conventions of the gazetteer on the disambiguation process.

Moreover, extraction models are inherently imperfect and generate imprecise confidence. We were able to use the disambiguation result to find negative confusing samples. Retraining the models with these negative samples enhances the confidence of true toponyms and reduces the confidence of false positives. This enhancement of extraction improves as a consequence the disambiguation (the aforementioned *reinforcement effect*). This process can be repeated iteratively, without any human interference, as long as there is improvement in the extraction and disambiguation.

The rest of the chapter is organized as follows. Section 4.3 presents a problem analysis and our general approach to iterative improvement of toponym extraction and disambiguation based on uncertain annotations. The adaptations we made to toponym extraction and disambiguation techniques are described in section 4.4. In section 4.5, we describe the experimental setup,

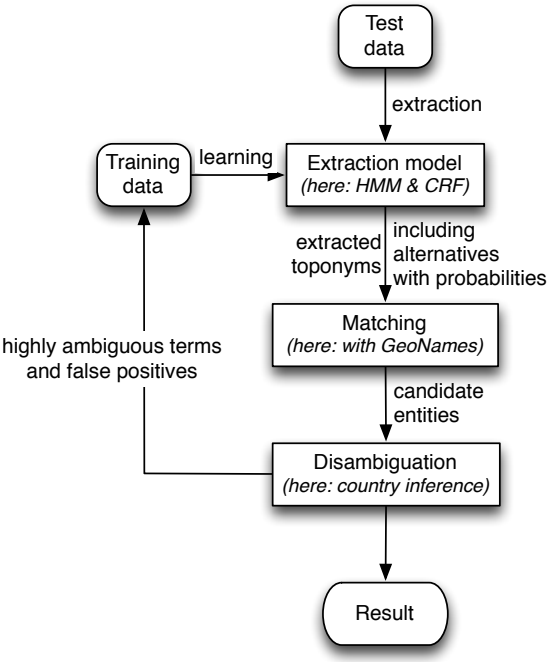


Figure 4.1: General approach.

present its results, and discuss some observations and their consequences. Finally, conclusions and future work directions are presented in section 4.6.

4.3 Problem Analysis and General Approach

The task we focus on is to extract toponyms from EuroCottage holiday home descriptions and use them to infer the country where the holiday property is located. We use this country inference task as a representative example of disambiguating extracted toponyms.

Our initial results from the previous chapter, where we developed a set of hand-coded grammar rules to extract toponyms, showed that effectiveness of disambiguation is affected by the effectiveness of extraction. We also proved the feasibility of a reverse influence, namely how the disambiguation result

can be used to improve extraction by filtering out terms found to be highly ambiguous during disambiguation.

One major problem with the hand-coded grammar rules is its *All-or-nothing* behavior. One can only annotate either '*Lake Como*' or '*Como*', but not both. Furthermore, hand-coded rules don't provide extraction confidences which we believe to be useful for the disambiguation process. We therefore propose an entity extraction and disambiguation approach based on uncertain annotations. The general approach illustrated in figure 4.1 has the following steps:

1. Prepare training data by manually annotating named entities (in our case toponyms) appearing in a subset of documents of sufficient size.
2. Use the training data to build a probabilistic extraction model.
3. Apply the extraction model on test data and training data. Note that we explicitly allow uncertain and alternative annotations with probabilities.
4. Match the extracted named entities against one or more gazetteers.
5. Use the entity candidates for the disambiguation process (in our case we try to disambiguate the country of the holiday home description).
6. Evaluate the extraction and disambiguation results for the training data and determine a list of highly ambiguous named entities and false positives that affect the disambiguation results. Use them to re-train the extraction model by introducing a new class for negative samples.
7. The steps from 2 to 6 are repeated automatically until there is no improvement any more in either the extraction or the disambiguation.

Note that the reason for including the training data in the process, is to be able to determine false positives in the result. From test data one cannot determine a term to be a false positive, but only to be highly ambiguous.

4.4 Extraction and Disambiguation Approaches

In this section, we illustrate the selected techniques for the extraction and disambiguation processes. We also present our adaptations to enhance the disambiguation by handling uncertainty and the imperfection in the extraction process, and how the extraction and disambiguation processes can reinforce each other iteratively.

4.4.1 Toponyms Extraction

For toponym extraction, we trained two probabilistic named entity extraction modules[1], one based on Hidden Markov Models (HMM) and one based on Conditional Random Fields (CRF).

HMM Extraction Module

The goal of HMM is to find the optimal tag sequence $T = t_1, t_2, ..., t_n$ for a given word sequence $W = w_1, w_2, ..., w_n$ that maximizes:

$$P(T \mid W) = \frac{P(T)P(W \mid T)}{P(W)} \tag{4.1}$$

where $P(W)$ is the same for all candidate tag sequences. $P(T)$ is the probability of the named entity (NE) tag. It can be calculated by Markov assumption which states that the probability of a tag depends only on a fixed number of previous NE tags. Here, in this work, we used $n = 4$. So, the probability of a NE tag depends on three previous tags, and then we have,

$$P(T) = P(t_1) \times P(t_2|t_1) \times P(t_3|t_1, t_2) \times P(t_4|t_1, t_2, t_3) \times \ldots \times P(t_n|t_{n-3}, t_{n-2}, t_{n-1}) \tag{4.2}$$

As the relation between a word and its tag depends on the context of the word, the probability of the current word depends on the tag of the previous word and the tag to be assigned to the current word. So $P(W|T)$ can be calculated as:

$$P(W|T) = P(w_1|t_1) \times P(w_2|t_1, t_2) \times \ldots \times P(w_n|t_{n-1}, t_n) \tag{4.3}$$

The prior probability $P(t_i|t_{i-3}, t_{i-2}, t_{i-1})$ and the likelihood probability $P(w_i|t_i)$ can be estimated from training data. The optimal sequence of tags can be efficiently found using the Viterbi dynamic programming algorithm [23].

CRF Extraction Module

HMMs have difficulty with modeling overlapped, non-independent features of the output part-of-speech tag of the word, the surrounding words, and capitalization patterns. Conditional Random Fields (CRF) can model these overlapping, non-independent features [24]. Here we used a linear chain CRF, the simplest model of CRF.

[1] We made use of the *lingpipe* toolkit for development: `http://alias-i.com/lingpipe`

A linear chain Conditional Random Field defines the conditional probability:

$$P(T \mid W) = \frac{\exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j f_j\left(t_{i-1}, t_i, W, i\right)\right)}{\sum_{t,w}\exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j f_j\left(t_{i-1}, t_i, W, i\right)\right)} \tag{4.4}$$

where f is set of m feature functions, λ_j is the weight for feature function f_j, and the denominator is a normalization factor that ensures the distribution p sums to 1. This normalization factor is called the *partition function*. The outer summation of the *partition function* is over the exponentially many possible assignments to t and w. For this reason, computing the *partition function* is intractable in general, but much work exists on how to approximate it [25].

The feature functions are the main components of CRF. The general form of a feature function is $f_j\left(t_{i-1}, t_i, W, i\right)$, which looks at tag sequence T, the input sequence W, and the current location in the sequence (i).

We used the following set of features for the previous w_{i-1}, the current w_i, and the next word w_{i+1}:

- The tag of the word.
- The position of the word in the sentence.
- The normalization of the word.
- The part of speech tag of the word.
- The shape of the word (Capitalization/Small state, Digits/Characters, etc.).
- The suffix and the prefix of the word.

An example for a feature function which produces a binary value for the current word shape is $Capitalized$:

$$f_i\left(t_{i-1}, t_i, W, i\right) = \begin{cases} 1 & \text{if } w_i \text{ is Capitalized} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

The training process involves finding the optimal values for the parameters λ_j that maximize the conditional probability $P(T \mid W)$. The standard parameter learning approach is to compute the stochastic gradient descent of the log of the objective function:

$$\frac{\partial}{\partial\lambda_k}\sum_{i=1}^{n}\log p(t_i|w_i)) - \sum_{j=1}^{m}\frac{\lambda_j^2}{2\sigma^2} \tag{4.6}$$

where the term $\sum_{j=1}^{m} \frac{\lambda_j^2}{2\sigma^2}$ is a Gaussian prior on λ to regularize the training. In our experiments we used the prior variance σ^2=4. The rest of the derivation for the gradient descent of the objective function can be found in [24].

Extraction Modes of Operation

We used the extraction models to retrieve sets of annotations in two ways:

- **First-Best:** In this method, we only consider the first most likely set of annotations that maximizes the probability $P(T \mid W)$ for the whole text. This method does not assign a probability for each individual annotation, but only to the whole retrieved set of annotations.
- **N-Best:** This method returns a top-N of possible alternative hypotheses in order of their estimated likelihoods $p(t_i|w_i)$. The confidence scores are assumed to be conditional probabilities of the annotation given an input token. A very low cut-off probability is additionally applied as well. In our experiments, we retrieved the top-25 possible annotations for each document then we made a cut-off for annotations with probability lower than 0.1.

4.4.2 Toponyms Disambiguation

For the toponym disambiguation task, we only select those toponyms annotated by the extraction models that match a reference in GeoNames. We furthermore use the clustering-based approach presented in section 3.4.3 to disambiguate to which entity an extracted toponym actually refers.

Handling Uncertainty of Annotations

Equation 3.11 gives equal weights to all toponyms. The countries of toponyms with a very low extraction confidence probability are treated equally to toponyms with high confidence; both count fully. To take the uncertainty in the extraction process into account, we adapt equation 3.11 to include the confidence of the extracted toponyms.

$$freq(c_j) = \sum_{i=1}^{n} \begin{cases} p(t_i|w_i) & \text{if } r_{ix} \in c_j \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

In this way, terms which are more likely to be toponyms have a higher contribution in determining the country of the document than less likely ones.

4.4.3 Improving Certainty of Extraction

In the abovementioned improvement, we make use of the extraction confidence to help the disambiguation to be more robust. However, those probabilities are not accurate and reliable all the time. Some extraction models (like HMM in our experiments) retrieve some false positive toponyms with high confidence probabilities. Moreover, some of these false positives have many entity candidates in many countries according to GeoNames (e.g., the term '*Bar*' refers to 58 different locations in GeoNames in 25 different countries; see table 4.6). These false positives affect the disambiguation process.

This is where we take advantage of the *reinforcement effect*. To be more precise, we introduce another class in the *extraction model* called 'highly ambiguous'. We assign to this class those terms in the training set that: (1) are not manually annotated as a toponym already, (2) have a match in GeoNames, and (3) the disambiguation process finds more than τ countries for documents that contain this term, i.e.,

$$|\{c|\exists d \bullet t_i \in d \wedge c = Country_{winner} \text{ for } d\}| \geq \tau \tag{4.8}$$

The threshold τ can be experimentally and automatically determined (see section 4.5.4). The extraction model is subsequently re-trained and the whole process is repeated without any human interference as long as there is improvement in extraction and disambiguation process for the training set.

Observe that terms manually annotated as toponym stay annotated as toponyms. Only terms not manually annotated as toponym but for which the extraction model predicts that they are a toponym anyway, are affected. The intention is that the extraction model learns to avoid prediction of certain terms to be toponyms when they appear to have a confusing effect on the disambiguation.

4.5 Experimental Results

In this section, we present the experimental results of our methods applied to a collection of holiday properties descriptions. The goal of the experiments is to investigate the influence of using annotation confidence on the disambiguation

effectiveness. Another goal is to show how to improve the imperfect extraction model using the outcomes of the disambiguation process and subsequently improving the disambiguation also.

4.5.1 Dataset

The dataset we use for our experiments is a collection of traveling agent holiday property descriptions from the EuroCottage portal as shown in section 3.6.1. We extended the collection used in the previous chapter. Our extended dataset consists of 1579 property descriptions for which we constructed a ground truth by manually annotating all toponyms. We used the collection in our experiments in two ways:

- **Train_Test set**: We split the dataset into a training set and a validation test set with ratio $2:1$, and used the training set for building the extraction models and finding the highly ambiguous toponyms, and the test set for a validation of extraction and disambiguation effectiveness against *new and unseen* data.
- **All_Train set**: We used the whole collection as a training and test set for validating the extraction and the disambiguation results.

The reason behind using the **All_Train set** for training and testing is that the size of the collection is considered small for NLP tasks. We want to show that the results of the **Train_Test set** can be better if there is enough training data.

4.5.2 Effect of Extraction with Confidence Probabilities

The goal of this experiment is to evaluate the effect of uncertainty in the extracted toponyms on the disambiguation results. Both a HMM and a CRF extraction model were trained and evaluated. Both modes of operation (**First-Best** and **N-Best**) were used for inferring the country of the holiday descriptions as described in section 3.4.3. We used the unmodified version of the clustering approach (equation 3.11) with the output of **First-Best** method, while we used the modified version (equation 4.7) with the output of **N-Best** method to make use of the confidence probabilities assigned to the extracted toponyms.

Results are shown in table 4.1. It shows the percentage of holiday home descriptions for which the correct country was successfully inferred. We can clearly see that the **N-Best** method outperforms the **First-Best** method for both

Table 4.1: Effectiveness of the disambiguation process for First-Best and N-Best methods in the extraction phase.

(a) On Train_Test set

	HMM	CRF
First-Best	62.59%	62.84%
N-Best	68.95%	68.19%

(b) On All_Train set

	HMM	CRF
First-Best	70.7%	70.53%
N-Best	74.68%	73.32%

bath	shop	terrace	shower	at
house	the	all	in	as
they	here	to	table	garage
parking	and	oven	air	gallery
each	a	farm	sauna	sandy

(a) Sample of false positive toponyms extracted by HMM.

north	zoo	west	well	travel
tram	town	tower	sun	sport

(b) Sample of false positive toponyms extracted by CRF.

Figure 4.2: False positive extracted toponyms.

the HMM and the CRF models. This supports our claim that dealing with alternatives along with their confidences yields better results.

4.5.3 Effect of Extraction Certainty Enhancement

While examining the results of extraction for both HMM and CRF, we discovered that there were many false positives among the extracted toponyms, i.e., words extracted as a toponym and having a reference in GeoNames, that are in fact not toponyms. Samples of such words are shown in figures 4.2a and 4.2b. These words affect the disambiguation result, if the matching references in GeoNames belong to many different countries.

We applied the proposed technique introduced in section 4.4.3 to reinforce the extraction confidence of true toponyms and to reduce them for highly ambiguous false positive ones. We used the N-Best method for extraction and the modified clustering approach for disambiguation. The best threshold τ for annotating terms as highly ambiguous has been experimentally determined (see

Table 4.2: Effectiveness of the disambiguation process using manual annotations.

Train_Test set	All_Train set
79.28%	78.03%

Table 4.3: Effectiveness of the extraction using Stanford NER.

	Pre.	Rec.	F1
On Train_Test set	0.8385	0.4374	0.5749
On All_Train set	0.8622	0.4365	0.5796

section 4.5.4).

Table 4.2 shows the results of the disambiguation process using the manually annotated toponyms. Table 4.3 show the extraction results of the state-of-the-art Stanford named entity recognition model [2]. Stanford is a NEE system based on CRF model which incorporates long-distance information [53]. It achieves good performance consistently across different domains. Tables 4.4 and 4.5 show the effectiveness of the disambiguation and the extraction processes respectively along iterations of refinement. The *No Filtering* rows show the initial results of disambiguation and extraction before any refinements have been done.

We can see an improvement in HMM extraction and disambiguation results. It starts with lower extraction effectiveness than Stanford model but it outperforms after retraining the model. This support our claim that the reinforcement effect can help imperfect extraction models iteratively. Further analysis and discussion shown in Section 4.5.5.

4.5.4 Optimal cutting threshold

Figures 4.3a, 4.3b, 4.3c and 4.3d show the effectiveness (in terms of Precision, Recall, and F1 measures) of the HMM and CRF extraction models through iterations of refinement versus the possible thresholds τ. Note that the graphs need to be read from right to left; a lower threshold means more terms being annotated as highly ambiguous. At the far right, no terms are annotated as such anymore, hence this is equivalent to no filtering.

We select the threshold with the highest F1 value. For example, the best threshold value is 3 in figure 4.3a. Observe that for HMM, the F1 measure (from

[2] http://nlp.stanford.edu/software/CRF-NER.shtml

Table 4.4: Effectiveness of the disambiguation process after iterative refinement.

(a) On Train_Test set

	HMM	CRF
No Filtering	68.95%	68.19%
1st Iteration	73.28%	68.44%
2nd Iteration	73.53%	68.44%
3rd Iteration	73.53%	-

(b) On All_Train set

	HMM	CRF
No Filtering	74.68%	73.32%
1st Iteration	77.56%	73.32%
2nd Iteration	78.57%	-
3rd Iteration	77.55%	-

Table 4.5: Effectiveness of the extraction process after iterative refinement.

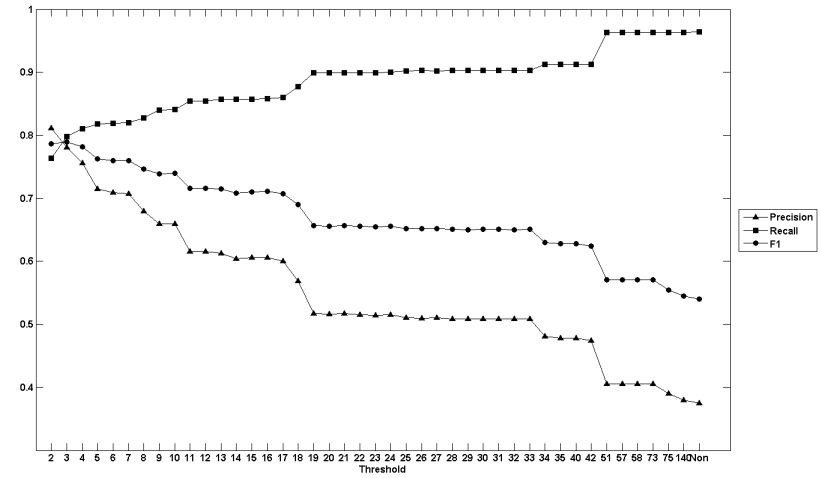
(a) On Train_Test set

	HMM			CRF		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
No Filtering	0.3584	0.8517	0.5045	0.6969	0.7136	0.7051
1st Iteration	0.7667	0.5987	0.6724	0.6989	0.7131	0.7059
2nd Iteration	0.7733	0.5961	0.6732	0.6989	0.7131	0.7059
3rd Iteration	0.7736	0.5958	0.6732	-	-	-

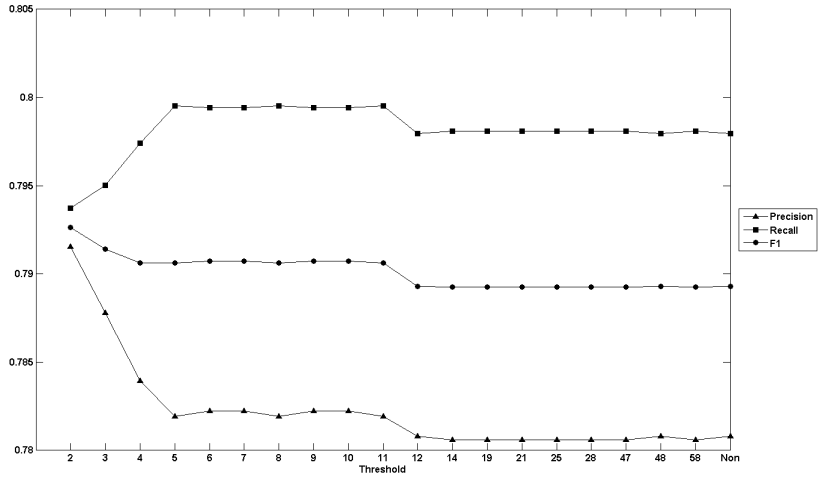
(b) On All_Train set

	HMM			CRF		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
No Filtering	0.3751	0.9640	0.5400	0.7496	0.7444	0.7470
1st Iteration	0.7808	0.7979	0.7893	0.7496	0.7444	0.7470
2nd Iteration	0.7915	0.7937	0.7926	-	-	-
3rd Iteration	0.8389	0.7742	0.8053	-	-	-

right to left) increases, hence the chosen threshold is that one which improves the extraction effectiveness. It does not do so for CRF, which is prominent cause for the poor improvements we saw earlier for CRF.

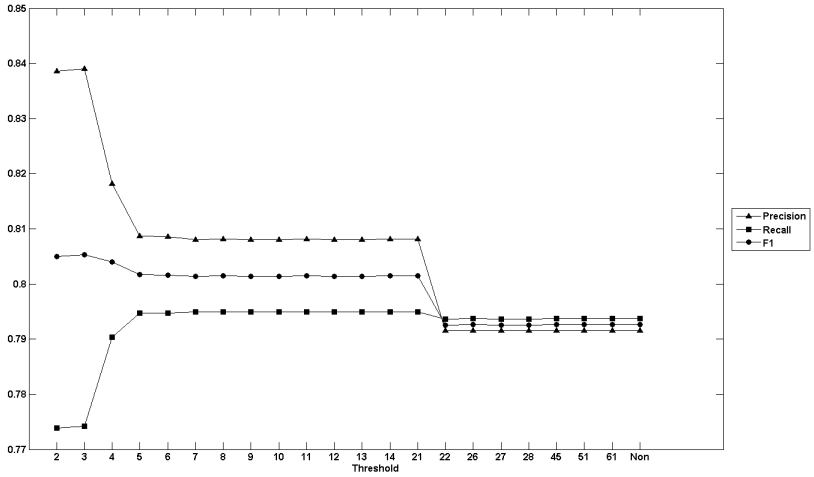


(a) HMM 1st iteration.

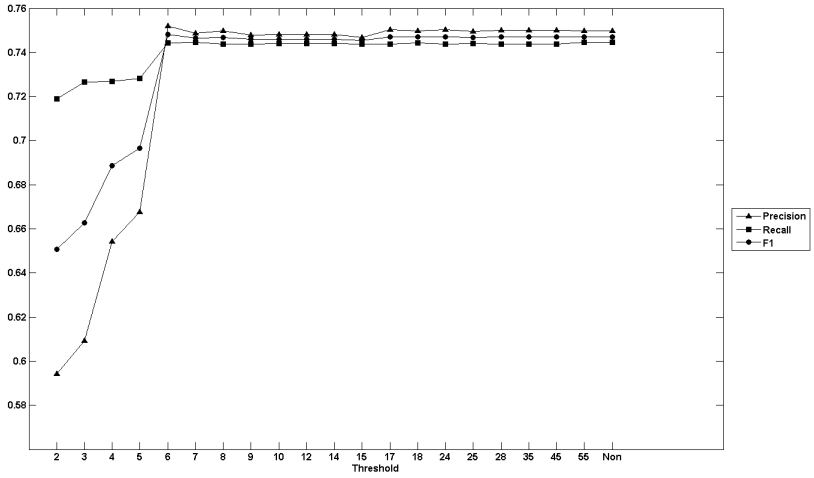


(b) HMM 2nd iteration.

Figure 4.3: The filtering threshold effect on the extraction effectiveness (On All_Train set)[3]



(c) HMM 3rd iteration.



(d) CRF 1st iteration.

Figure 4.3 (continued)

[3]These graphs are supposed to be discrete, but we present it like this to show the trend of extraction effectiveness against different possible cutting thresholds.

4.5.5 Further Analysis and Discussion

For deep analysis of results, we present in table 4.6 detailed results for the property description shown in figure 3.6a. We have the following observations and thoughts:

- From table 4.1, we can observe that both HMM and CRF initial models were improved by considering confidence of the extracted toponyms (see section 4.5.2). However, for HMM, still many false positives were extracted with high confidence scores in the initial extraction model.

- The initial HMM results showed a very high recall rate with a very low precision. In spite of this our approach managed to improve precision significantly through iterations of refinement. The refinement process is based on removing highly ambiguous toponyms resulting in a slight decrease in recall and an increase in precision. In contrast, CRF started with high precision which could not be improved by the refinement process. Apparently, the CRF approach already aims at achieving high precision at the expense of some recall (see table 4.5).

- In table 4.5 we can see that the precision of the HMM outperforms the precision of CRF after iterations of refinement. This results in achieving better disambiguation results for the HMM over the CRF (see table 4.4)

- It can be observed that the highest improvement is achieved on the first iteration. This is where most of the false positives and highly ambiguous toponyms are detected and filtered out. In the subsequent iterations, only few new highly ambiguous toponyms appeared and were filtered out (see table 4.5).

- It can be seen in table 4.6 that initially non-toponym phrases like '*.-30.09.)*' and '*IMPORTANT*' were falsely extracted by HMM. These don't have a GeoNames reference, so were not considered in the disambiguation step, nor in the subsequent re-training. Nevertheless they disappeared from the top-N annotations. The reason for this behavior is that initially the extraction models were trained on annotating for only one type (toponym), whereas in subsequent iterations they were trained on two types (toponym and highly ambiguous non-toponym). Even though the aforementioned phrases were not included in the re-training, their confidences fell below the 0.1 cut-off threshold after the 1st iteration.

4.6 Conclusions and Future Directions

NEE and NED are inherently imperfect processes that moreover depend on each other. The aim of this chapter is to examine and make use of this dependency for the purpose of improving the disambiguation by iteratively enhancing the certainty of extraction, and vice versa. Experiments were conducted with a set of holiday home descriptions with the aim to extract and disambiguate toponyms as a representative example of named entities. HMM and CRF probabilistic approaches were applied for extraction. We compared extraction in two modes, First-Best and N-Best. A clustering approach for disambiguation was applied with the purpose to infer the country of the holiday home from the description.

We examined how handling the uncertainty of extraction influences the effectiveness of disambiguation, and reciprocally, how the result of disambiguation can be used to improve the certainty of extraction. The extraction models are automatically retrained after discovering highly ambiguous false positives among the extracted toponyms. This iterative process improves the precision of the extraction. We argue that our approach that is based on uncertain annotation has much potential for making information extraction more robust against ambiguous situations and allowing it to gradually learn. We provide insight into how and why the approach works by means of an in-depth analysis of what happens to individual cases during the process.

We claim that this approach can be adapted to suit any kind of named entities. It is just required to develop a mechanism to find highly ambiguous false positives among the extracted named entities. Coherency measures can be used to find highly ambiguous named entities. In part III, we plan to apply and adapt our approach for other types of named entities on Twitter messages domain. Furthermore, the approach appears to be fully language independent, therefore we like to prove that this is the case and investigate its effect on texts in multiple and mixed languages. This is shown in the next chapter.

Table 4.6: Deep analysis for the extraction process of the property shown in figure 3.6a (∈: present in GeoNames; #refs: number of references; #ctrs: number of countries).

		GeoNames look-up			Confidence	Disambiguation
	Extracted Toponyms	∈	**#refs**	**#ctrs**	**probability**	**result**
Manually annotated toponyms	Armacao de Pera	√	1	1	-	Correctly Classified
	Alcantarilha	√	1	1	-	
	Sehora da Rocha	×	-	-	-	
	Playa de Armacao de Pera	×	-	-	-	
	Armacao de Pera	√	1	1	-	
Initial HMM model with First-Best extraction method	Balcony 8 m2	×	-	-	-	Misclassified
	Terrace Club	√	1	1	-	
	Armacao de Pera	√	1	1	-	
	.-30.09.)	×	-	-	-	
	Alcantarilha	√	1	1	-	
	Lounge	√	2	2	-	
	Bar	√	58	25	-	
	Car hire	×	-	-	-	
	IMPORTANT	×	-	-	-	
	Sehora da Rocha	×	-	-	-	
	Playa de Armacao de Pera	×	-	-	-	
	Bus	√	15	9	-	
	Armacao de Pera	√	1	1	-	
Initial HMM model with N-Best extraction method	Alcantarilha	√	1	1	1	Correctly Classified
	Sehora da Rocha	×	-	-	1	
	Armacao de Pera	√	1	1	1	
	Playa de Armacao de Pera	×	-	-	0.999849891	
	Bar	√	58	25	0.993387918	
	Bus	√	15	9	0.989665883	
	Armacao de Pera	√	1	1	0.96097006	
	IMPORTANT	×	-	-	0.957129986	

	Lounge	√	2	2	0.916074183	
	Balcony 8 m2	×	-	-	0.877332628	
	Car hire	×	-	-	0.797357377	
	Terrace Club	√	1	1	0.760384949	
	In	√	11	9	0.455276943	
	.-30.09.)	×	-	-	0.397836259	
	.-30.09.	×	-	-	0.368135755	
	.	×	-	-	0.358238066	
	. Car hire	×	-	-	0.165877044	
	adavance.	×	-	-	0.161051997	
HMM model after 1st iteration with N-Best extraction method	Alcantarilha	√	1	1	0.999999999	
	Sehora da Rocha	×	-	-	0.999999914	Correctly
	Armacao de Pera	√	1	1	0.999998522	Classified
	Playa de Armacao de Pera	×	-	-	0.999932808	
Initial CRF model with First-Best extraction method	Armacao	×	-	-	-	
	Pera	√	2	1	-	
	Alcantarilha	√	1	1	-	Correctly
	Sehora da Rocha	×	-	-	-	Classified
	Playa de Armacao de Pera	×	-	-	-	
	Armacao de Pera	√	1	1	-	
Initial CRF model with N-Best extraction method	Alcantarilha	√	1	1	0.999312439	
	Armacao	×	-	-	0.962067016	
	Pera	√	2	1	0.602834683	Correctly
	Trips	√	3	2	0.305478198	Classified
	Bus	√	15	9	0.167311005	
	Lounge	√	2	2	0.133111374	
	Reception	√	1	1	0.105567287	

CHAPTER 5

Multilinguality and Robustness

5.1 Summary

In the previous chapters, we showed that toponym extraction and disambiguation are highly dependent processes. We examined how handling the uncertainty of extraction influences the effectiveness of disambiguation, and reciprocally, how the result of disambiguation can be used to improve the effectiveness of extraction through iterations of refinement. In this chapter we aim to test the robustness of our claim in multiple ways. Robustness is examined on three aspects: language independence, high and low HMM threshold settings, and limited training data. We propose a hybrid toponym extraction approach based on Hidden Markov Models (HMM) and Support Vector Machines (SVM). Hidden Markov Model is used for extraction with high recall and low precision. Then SVM is used to find false positives based on informativeness features and coherence features derived from the disambiguation results. Experimental results conducted with a set of descriptions of holiday homes with the aim to extract and disambiguate toponyms showed that the proposed approach outperform the state-of-the-art methods of extraction and also proved to be robust.

The contents of this chapter have been published as [54].

5.2 Introduction

Most of existing extraction techniques are language-dependent as they use part of speech (POS) tags as an important extraction feature. And it is known that it takes some effort to tune the parameters and the thresholds. Furthermore, machine learning approaches require to be trained on large corpuses. In practice, one would like to have more robustness so that accuracy is not easily ham-

pered. In this chapter, we specifically address robustness against threshold settings, situations with other languages, and situations with limited training data.

In this chapter we propose a hybrid extraction approach based on Hidden Markov Models (HMM) and Support Vector Machines (SVM). An initial HMM is trained and used for extraction. We use a low cutting threshold to achieve high recall resulting in low precision. A clustering based approach for disambiguation is then applied. A set of coherence features are extracted for the extracted toponyms based on the disambiguation results feedback and also on informativeness measures (like Inverse Document Frequency and Gain). A SVM is then trained with the extracted features to classify the HMM extracted toponyms into true positives and false positives resulting in improving the precision and hence the F1 measure. Our hybrid approach outperforms the Conditional Random Fields (CRF), the state-of-the-art method of extraction and Stanford NER, the prominent Named Entity Recognition System. Furthermore, our hybrid approach is shown to be language independent as all the used methods are not based on language dependent techniques like POS tags which is commonly used with the NER systems. Robustness of the proposed approach is experimentally proved by applying different HMM cutting thresholds, evaluating it across multiple languages and also with smaller training sets. More aspects of robustness like evaluating across multiple domains and using different types of named entities are left for future work. To examine our hybrid approach, we conducted experiments on a collection of holiday home descriptions from the EuroCottage portal.

Contributions: We can summarize our contributions as follows: (1) We propose a hybrid toponym extraction approach based on HMM and SVM. (2) The proposed system is proved to be robust against three aspects: different languages, different cutting thresholds, and limited training data. (3) We introduce some features (informativeness and coherence-based) that can be used to enhance the process of toponym extraction.

The rest of the chapter is organized as follows. Section 5.3 presents our proposed approach for toponym extraction and disambiguation. In section 5.4, we describe the experimental setup, present its results, and discuss some observations and their consequences. Finally, conclusions and future directions are presented in section 5.5.

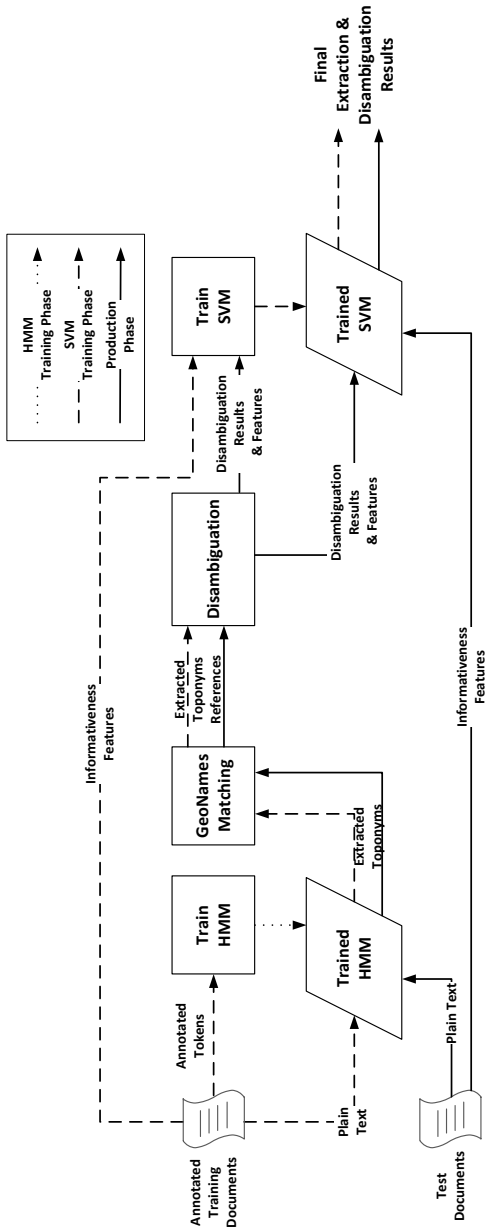


Figure 5.1: Our proposed hybrid approach.

5.3 Hybrid Approach

The hybridness of our proposed approach can be viewed from two points of view. It can be viewed as a hybrid approach of toponym extraction and disambiguation processes. Clues derived from the disambiguation results are used to enhance extraction. Also our system can be viewed as a hybrid machine learning approach for extraction where HMM and SVM are combined to achieve better results. An initial HMM is trained and used for extraction with high recall. A SVM is then trained to classify the HMM extracted toponyms into true positives and false positives resulting in improving the precision and hence the F1 measure.

5.3.1 System Phases

The system illustrated in Figure 5.1 has the following Phases:

Phase 1: HMM Training

1. Training data is prepared by manually annotating all toponyms. Tokens are tagged, following the CoNLL[1] standards, by either a LOCATION or O tag which represents words that are not part of a location phrase.

2. Training data is used to train a HMM[2][3] [55] for toponym extraction. The advantage of statistical techniques for extraction is that they provide alternatives for annotations accompanied with confidence probabilities. Instead of discarding these, as is commonly done by selecting the top-most likely candidate, we use them to enrich the knowledge for disambiguation. The probabilities proved to be useful in enhancing the disambiguation process (see chapter 4).

Phase 2: SVM Training

1. The trained HMM is then used to extract toponyms from the training set. A low cutting threshold is applied with the purpose of achieving high recall. The extracted toponyms are then matched against GeoNames gazetteer. For each toponym, a list of candidate references are fed to the disambiguation process.

[1] http://www.cnts.ua.ac.be/conll2002/ner/

[2] http://alias-i.com/lingpipe/

[3] We used an HmmCharLmEstimator which employs a maximum a posteriori transition estimator and a bounded character language model emission estimator.

2. The disambiguation process tries to find only one representative reference for each extracted toponym based on its coherency with other toponyms mentioned in the same document.

3. Two sets of features (informativeness and coherence-based) are computed for each extracted toponym. Details of the selected features are described in section 5.3.3.

4. The extracted set of features are used to train the SVM classifier [4][5] to distinguish between true positives toponyms and false positives ones.

Phase 3: Production

1. The trained HMM is applied on the test set. The extracted toponyms are matched against GeoNames and their candidate references are disambiguated. Informativeness and coherence features are computed and fed to the trained SVM to find the final results of toponyms extraction process.

2. Disambiguation process can be repeated using the final set of extracted toponyms to get the improvement reflected on the disambiguation results.

The main intuition behind our approach is to make use of more clues than those often used by traditional extraction techniques (like POS, word shape, preceding and succeeding words). We deliberately use set of language-independent features to ensure robustness across multiple languages. To make use of those features we start with high recall and then filter the extracted toponyms based on those features. Even by using a higher cutting threshold, our approach is still able to enhance the precision at the expense of some recall resulting in enhancement of the overall F1 measure. Moreover, the features are found to be highly discriminative, so that only few training samples are required to train the SVM classifier good enough to make correct decisions.

5.3.2 Toponyms Disambiguation

For the toponym disambiguation task, we only select those toponyms annotated by the extraction models that match a reference in GeoNames. We use the clustering approach of presented in section 3.4.3 with the purpose to infer

[4]http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

[5]We used C-support vector classification (C-SVC) type of SVM with RBF kernel.

the country of the holiday home from the description. The clustering approach is an unsupervised disambiguation approach based on the assumption that toponyms appearing in same document are likely to refer to locations close to each other distance-wise. For our holiday home descriptions, it appears quite safe to assume this. For each toponym t_i, we have, in general, multiple entity candidates. Let $R(t_i) = \{r_{ix} \in \text{GeoNames gazetteer}\}$ be the set of reference candidates for toponym t_i. Additionally each reference r_{ix} in GeoNames belongs to a country c_j. By taking one entity candidate for each toponym, we form a cluster. A cluster, hence, is a possible combination of entity candidates, or in other words, one possible entity candidate of the toponyms in the text. In this approach, we consider all possible clusters, compute the average distance between the candidate locations in the cluster, and choose the cluster $Cluster_{min}$ with the lowest average distance. We choose the most often occurring country $Country_{winner}$ in $Cluster_{min}$ for disambiguating the country of the document. In effect the above-mentioned assumption states that the entities that belong to $Cluster_{min}$ are the true representative entities for the corresponding toponyms as they appeared in the text.

5.3.3 Selected Features

Coherence features derived from disambiguation results along with informativeness features are computed for all the extracted toponyms generated by the HMM.

Using informativeness features in NER is introduced by Rennie et al. [56]. They conducted a study on identifying restaurant names from posts to a restaurant discussion board. They found the informativeness scores to be an effective restaurant word filter. Furche et al. [57] introduce a system called AMBER for extracting data from an entire domain. AMBER employs domain specific gazetteers to discern basic domain attributes on a web page, and leverages repeated occurrences of similar attributes to group related attributes into records.

For each extracted toponym the following set of informativeness features are computed:

1. **Inverse Document Frequency (IDF)**: IDF is an informativeness score that embodies the principle that the more frequent a word is, the lower the chance it is a relevant toponym. The IDF score for an extracted toponym t is:

$$IDF = -log\frac{d_t}{D} \tag{5.1}$$

where d_t is the document frequency of the toponym t, and D is the total number of documents.

2. **Residual Inverse Document Frequency (RIDF)**: RIDF is an extension of IDF that has proven effective for NER [56]. RIDF is calculated as the difference between the IDF of a toponym and its expected IDF according to the poisson model. The RIDF score can be calculated by the formula:

$$expIDF = -log(1 - e^{-f_t/D}) \tag{5.2}$$

$$RIDF = IDF - expIDF \tag{5.3}$$

where f_t is the frequency of the toponym across all documents D.

3. **Gain**: Gain is a feature that can be used to identify *important* or *informative* terms. For a toponym t, Gain is derived as:

$$Gain(t) = \frac{d_t}{D}(\frac{d_t}{D} - 1 - log\frac{d_t}{D}) \tag{5.4}$$

4. **Extraction Confidence (EC)**: Extraction confidence (probability) is the HMM conditional probability of the annotation given an input word. The goal of HMM is to find the optimal tag sequence $T = t_1, t_2, ..., t_n$ for a given word sequence $W = w_1, w_2, ..., w_n$ that maximizes:

$$P(T \mid W) = \frac{P(T)P(W \mid T)}{P(W)} \tag{5.5}$$

The prior probability $P(t_i|t_{i-2}, t_{i-1})$ and the likelihood probability $P(w_i|t_i)$ can be estimated from training data. The optimal sequence of tags can be efficiently found using the Viterbi dynamic programming algorithm [23]. The extraction confidence $P(t|w)$ is the probability of being a part of toponym given a token.

Furthermore, the following set of coherence features are computed based on the disambiguation results:

1. **Distance (D)**: The distance feature is the kilo-metric distance between the coordinates of the selected candidate reference r_{ix} for toponym t_i and the coordinates of the inferred country $Country_{winner}$.

$$Distance = Coordinates(r_{ix}) - Coordinates(Country_{winner}) \tag{5.6}$$

2. **Standard Score (SS)**: It is calculated by dividing the distance between the coordinates of the r_{ix} and $Country_{winner}$ over the standard deviation of all selected references distances to $Country_{winner}$.

$$StandardScore = \frac{Coordinates(r_{ix}) - Coordinates(Country_{winner})}{\sigma} \tag{5.7}$$

3. **Number of GeoNames candidate references (#Geo)**: It is simply the number of candidate references for the toponym ti.

$$\#\text{GeoNames Refs} = |r_{ix}| \tag{5.8}$$

4. **Belongingness to the disambiguated country (Bel)**: Indicates whether or not r_{ix} belongs to $Country_{winner}$.

$$\text{Belongingness to } Country_{winner} = \begin{cases} 1 & \text{if } Country(r_{ix}) = Country_{winner} \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

Informativeness features tend to find those false positives that appear multiple times across the collection. Those highly repeated words are more likely to be false positives toponyms. On the other hand, some false positives appear only rarely in the collection. Those toponyms cannot be caught by informativeness features. Here where we make use of coherence-based features. Coherence features tend to find those false positives that are not coherent with other toponyms. The usage of a combination of both sets of features maximizes the extraction effectiveness (F1 measure).

Unlike traditional features commonly used with NER systems like (POS), all our selected features are language independent and thus our approach can be applied to any language as the GeoNames gazetteer has representations for toponyms in different languages. Furthermore we avoid using word shape features as languages like German require the capitalization of all nouns making capitalization a useless feature to extract NE.

5.4 Experimental Results

In this section, we present the results of experiments with the proposed approach applied to a collection of holiday properties descriptions. The goals

of the experiments are to compare our approach with the state-of-the-art approaches and systems and to show its robustness in terms of language independence, high and low HMM threshold settings, and limited training data.

5.4.1 Dataset

The dataset we use for our experiments is a collection of traveling agent holiday property descriptions from the EuroCottage portal. The descriptions not only contain information about the property itself and its facilities, but also a description of its location, neighboring cities and opportunities for sightseeing. Descriptions are also available in German and Dutch. Some of these descriptions are direct translations and some others have independent descriptions of the same holiday cottage. The dataset includes the country of each property which we use to validate our results. Figure 5.2 shows a representative example of a holiday property description in English, German and Dutch. The manually annotated toponyms are written in bold. The dataset consists of 1181 property descriptions for which we constructed a ground truth by manually annotating all toponyms for only the English version. The German and the Dutch versions of descriptions are annotated automatically by matching them against all toponyms that appear in the English version or their translations. For example '*Cologne*' in the English version is translated to '*Köln*' and matched in the German version and translated to '*Keulen*' and matched in the Dutch version. Although this method is not 100% reliable due to slight differences in translated versions, we believe that it is reliable enough as ground truth for showing the language independency of our approach.

We split the dataset into a training set and a validation test set with ratio 2 : 1. We used the training set for training the HMM extraction model and the SVM classifier, and the test set for evaluating the extraction and disambiguation effectiveness for *new and unseen* data.

Olšova Vrata 5 km from **Karlovy Vary**: On the edge of the **Slavkovsky les** nature reserve. Small holiday hamlet next to the hotel which has been a popular destination for **Karlsbad** inhabitants for the past 30 years new, large house with 2 apartments, 2 storeys, built in 2004, surrounded by trees, above **Karlovy Vary**, in a secluded, sunny position, 10 m from the woods edge. Private, patio (20 m2), garden furniture. In the house: table-tennis. Central heating. Breakfast and half-board on request. Motor access to the house (in winter snow chains necessary). Parking by the house. Shop 4 km, grocers 1.5 km, restaurant 150 m, bus stop 550 m, swimming pool 6 km, indoor swimming pool 6 km, thermal baths 6 km, tennis 1 km, golf course 1.5 km, skisport facilities 25 km. Please note: car essential. Airport 1.5 km (2 planes/day). On request: Spa treatments, green fee. Ski resort **Klinovec**, 20 km.

(a) English Description.

Olšova Vrata 5 km von **Karlovy Vary**: Am Rande des Naturschutzgebiets **Slavkovský les**, neben Hotel ein kleiner Ferienweiler - schon vor 30 Jahren als Ausflugziel der Einwohner **Karlsbads** bekannt. Grosses Zweifamilienhaus auf 2 Stockwerken, Baujahr 2004, umgeben von Bäumen. Oberhalb von **Karlovy Vary**, alleinstehende, sonnige Lage, 10 m vom Waldrand. Zur Alleinbenutzung: Sitzplatz (20 m2), Gartenmöbel. Im Hause: Tischtennis, Zentralheizung. Frühstück und Halbpension möglich. Zufahrt bis zum Haus. Im Winter bitte Schneeketten mitbringen. Parkplatz beim Haus. Einkaufsgeschäft 4 km, Lebensmittelgeschäft 1.5 km, Restaurant 150 m, Bushaltestelle 550 m, Freibad 6 km, Hallenbad 6 km, Thermalbad 6 km. Golfplatz 1.5 km, Tennis 1 km, Skisportanlagen 25 km. Bitte beachten: Fahrzeug empfohlen. 1.5 km Flughafen (2 Flugzeuge/Tag). Auf Anfrage: Kuranwendungen, Greenfee. Skigebiet **Klinovec** 20 km.

(b) German Description.

Olšova Vrata 5 km van **Karlovy Vary**: Aan de rand van het natuurreservaat **Slavkovsky les**, naast hotel ook een klein vakantiegehucht - al 30 jaar bekend bij de inwoners van **Karlsbad** als uitstapje nieuw, groot huis met 2 appartementen van 2 verdiepingen, bouwjaar 2004, omgeven door bomen, boven **Karlovy Vary**, geïsoleerde, zonnige ligging, 10 m van de bosrand. Voor alleengebruik, zitje in de tuin (20 m2), tuinmeubelen. In het huis: tafeltennis. Centrale verwarming. Ontbijt en half pension op verzoek. Toegangsweg tot aan het huis (in de winter sneeuwkettingen noodzakelijk). Parkeerplaats bij het huis. Winkel 4 km, levensmiddelenwinkel 1.5 km, restaurant 150 m, bushalte 550 m, openluchtzwembad 6 km, overdekt zwembad 6 km, thermaalbad 6 km, tennis 1 km, golfterrein 1.5 km, ski faciliteiten 25 km. Let op: auto noodzakelijk. Vliegveld, 1.5 km (2 vliegtuigen per dag). Op aanvraag: kuuroordbehandelingen, greenfee. Skigebied **Klinovec**: 20 km.

(c) Dutch Description.

Figure 5.2: Examples of EuroCottage holiday home description in three languages (toponyms in bold).

Table 5.1: Test set statistics through different phases of our system pipeline.

	#Top./Doc.			#Top./Doc. ∈GeoNames			Degree of ambiguity		
	EN	**DE**	**NL**	**EN**	**DE**	**NL**	**EN**	**DE**	**NL**
Ground Truth	5.04	4.62	3.51	3.47	3.10	2.46	7.24	6.15	6.78
HMM(0.1)	12.02	11.31	11.38	6.51	5.72	5.85	8.69	9.27	10.33
HMM(0.1)+SVM	5.24	5.04	3.91	3.59	3.18	2.58	8.43	7.38	7.78

5.4.2 Dataset Analysis

The aim of this experiment is to show some statistics about the test set in all versions through different phases of our system pipeline. Table 5.1 shows the number of toponyms per property description [*#Top./Doc.*], the number of toponyms per property that have references in GeoNames [*#Top./Doc. ∈GeoNames*], and the average degree of ambiguity per toponyms [*Degree of ambiguity*] (i.e. the average number of references in GeoNames for a given toponym). *Ground Truth* represents manual annotations statistics. *HMM(0.1)* represents statistics of the extracted toponyms resulting from applying HMM on the test set with cutting probability threshold 0.1, while *HMM(0.1)+SVM* represents statistics of the extracted toponyms resulting from applying SVM after HMM on the test set.

As can be observed from table 5.1 that HMM extracts many false positives. Examples of those false positives that have references in GeoNames are shown in figure 5.3[6].

It can also be noticed that the English version contains more toponyms per property description. Our method of automatically annotating the German and the Dutch texts misses few annotations. This doesn't harm the evaluation process of the proposed method as our approach works on improving the precision with some loss in recall. Hence, we can claim that precision/recall/F1 measures of our proposed approach applied on German and Dutch versions shown on the section 5.4.4 can be regarded as a lower bound.

5.4.3 SVM Features Analysis

In this experiment we evaluate the selected set of features used for SVM training on the English collection. We want to show the effect of these features

[6]We match the extracted toponyms against names of places, their ascii representation and their alternative representations in GeoNames gazetteer.

bath[34]	shop[1]	terrace[11]	shower[1]	parking[3]
house[5]	garden[24]	sauna[6]	island[16]	farm[5]
villa[49]	here[7]	airport[3]	table[9]	garage[1]

(a) English

bett[1]	bad[15]	strand[95]	meer[15]	foto[11]
bergen[59]	garage[1]	bar[58]	villa[49]	wald[51]
billard[3]	westen[11]	stadt[7]	salon[12]	keller[27]

(b) German

winkel[58]	terras[3]	douche[2]	woon[1]	bergen[59]
kortom[2]	verder[1]	gas[9]	villa[49]	garage[1]
tuin[2]	hal[20]	chalet[8]	binnen[3]	rond[1]

(c) Dutch

Figure 5.3: Examples of false positives (toponyms erroneously extracted by HMM(0.1)) and their number of references in GeoNames.

on the effectiveness of the SVM classifier. The aim of the SVM is to find the false positives toponyms among those extracted by the HMM. Two groups of features are used. Informativeness features and coherence features (features derived from disambiguation results). Table 5.2 shows:

- Extraction and disambiguation results using each of the features individually to train the SVM classifier.
- Information Gain [*IG*] for each feature. *IG* measures the amount of information in bits about the class prediction (in our case true positive toponym or false positive).
- The extraction and disambiguation results using each group of features (Informativeness (*Inf*) and coherence (*Coh*)) and using both combined (*All*).
- Extraction and disambiguation results for only *HMM* with threshold 0.1 (prior to the usage of the SVM).
- Disambiguation results using manually annotated toponyms (*Ground Truth*).

Extraction results are evaluated in terms of precision [*Pre.*], recall [*Rec.*] and [*F1*] measures, while disambiguation results [*Dis.*] are evaluated in terms of the percentage of holiday home descriptions for which the correct country was inferred.

The coherence features can be only calculated for toponyms that belong to GeoNames. This implies that its effect only appears on false positives that belong to GeoNames. To make their effect more clear, we presented two sets of results:

- *All extracted toponyms*: where all toponyms are used to train HMM and SVM regardless of whether they exist in GeoNames or not. Evaluation is done for all extracted toponyms.
- *Only toponyms* $\in$ *GeoNames*: where only toponyms existing in GeoNames are used to train and evaluate HMM and SVM.

By looking at [*IG*] of each feature we can observe that the [*Bel*], [*IDF*] and [*EC*] are highly discriminative features, while [#*Geo*] seems to be a bad feature as it has low effect on the SVM output.

Using manually annotated toponyms for disambiguation, the best possible input one would think, may not produce the best possible disambiguation result. For example, the disambiguation result of HMM(0.1)+SVM(Gain) is higher than that of the ground truth. This is because some holiday cottages are located on the border with other country, so that description mentions cities from other country rather than the country of the cottage. This does not mean that the correct representative candidates for toponyms are missed. Moreover, since our disambiguation result is based on voting, we attribute this effect to chance. The extraction model may produce a false positive toponym which happens to sway the vote to the correct country, in other words, there are cases of correct results for the wrong reasons.

It can be also observed that low recall leads to poor disambiguation results. That is because low recall may result in extracting no toponyms from the property description and hence the country of that property is misclassified.

Table 5.2 shows how using the SVM classifier enhances the extraction and the disambiguation results. The effect of combining both set of features is more clear in the results of [*Only toponyms* $\in$ *GeoNames*]. Precision is improved significantly, and hence the F1 measure, by using the coherence features beside the informativeness ones.

Table 5.3 shows the extracted toponyms for the property shown in figure 5.2a using different methods. Informativeness features tend to find those false

Table 5.2: Extraction and disambiguation results using different features for English version.

	All extracted toponyms				
	IG	**Pre.**	**Rec.**	**F1**	**Dis.**
Ground Truth		**1**	**1**	**1**	79.1349
HMM(0.1)		0.3631	**0.8659**	0.5116	75.0636
HMM(0.1)+SVM(IDF)	0.1459	0.5514	0.8336	0.6637	80.4071
HMM(0.1)+SVM(RIDF)	0.1426	0.5430	0.8472	0.6618	80.4071
HMM(0.1)+SVM(Gain)	0.1013	0.5449	0.8205	0.6549	**80.9160**
HMM(0.1)+SVM(EC)	0.2223	0.7341	0.7489	0.7414	78.3715
HMM(0.1)+SVM(D)	0.0706	0.6499	0.5726	0.6088	74.5547
HMM(0.1)+SVM(SS)	0.0828	0.6815	0.5166	0.5877	68.4478
HMM(0.1)+SVM(#Geo)	0.1008	0.4800	0.6099	0.5372	71.7557
HMM(0.1)+SVM(Bel)	0.3049	**0.8106**	0.4942	0.6140	73.0280
HMM(0.1)+SVM(Inf)		0.7764	0.7756	0.7760	79.8982
HMM(0.1)+SVM(Coh)		0.8106	0.4940	0.6138	73.0280
HMM(0.1)+SVM(All)		0.7726	0.8014	**0.7867**	79.8982

	Only extracted toponyms $\in$ GeoNames				
	IG	**Pre.**	**Rec.**	**F1**	**Dis.**
Ground Truth		**1**	**1**	**1**	79.1349
HMM(0.1)		0.4874	**0.9121**	0.6353	75.0636
HMM(0.1)+SVM(IDF)	0.2652	0.7612	0.8983	0.8241	**81.1705**
HMM(0.1)+SVM(RIDF)	0.2356	0.7536	0.9107	0.8247	80.9160
HMM(0.1)+SVM(Gain)	0.1754	0.6419	0.8656	0.7372	76.3359
HMM(0.1)+SVM(EC)	0.2676	0.8148	0.8243	0.8195	78.3715
HMM(0.1)+SVM(D)	0.1375	0.6563	0.8584	0.7439	77.6081
HMM(0.1)+SVM(SS)	0.1077	0.6802	0.7444	0.7108	68.4478
HMM(0.1)+SVM(#Geo)	0.0791	0.4878	0.9121	0.6356	75.0636
HMM(0.1)+SVM(Bel)	0.3813	0.8106	0.7117	0.7579	73.0280
HMM(0.1)+SVM(Inf)		0.8181	0.8823	0.8490	80.6616
HMM(0.1)+SVM(Coh)		0.8117	0.7451	0.7770	76.3359
HMM(0.1)+SVM(All)		**0.8865**	0.8453	**0.8654**	79.8982

positives that appear multiple times across the collection like {In, Shop}. On the other hand, disambiguation features tend to find those false positives that are not coherent with other toponyms like {Airport}. The usage of a combination of both sets of features maximizes the extraction effectiveness (F1 measure).

Table 5.3: Extracted toponyms for the property shown in figure 5.2a

	HMM(0.1)	HMM(0.1) + SVM(Inf)	HMM(0.1) + SVM(Dis)	HMM(0.1) + SVM(All)
[+]Olšova Vrata	+	+	+	+
[+]Karlovy Vary	+	+	+	+
[+]Slavkovsky les	+	+	+	+
[+]Karlsbad	+	+	+	+
[+]Karlovy Vary	+	+	+	+
[+]Klinovec	+	+	+	+
[-]In	+	-	+	-
[-]Shop	+	-	+	-
[-]Airport	+	+	-	-

5.4.4 Multilinguality, Different Thresolding Robustness and Competitors

In this experiment, we want to show the multiligualitiy and system robustness across different languages and against different threshold settings. Multilinguality is guaranteed by our approach as we only use language independent methods of extraction and filtering. We effectively avoided using POS tags as feature since it is highly language-dependent and for many languages there are no good automatic POS taggers available. Table 5.4 shows the effectiveness of our proposed approach applied on English, German, and Dutch versions in terms of the F1 and the disambiguation results over the state-of-the-art: the CRF, and the Stanford NER models [7]. CRF is considered one of the famous techniques in NER. We trained a CRF on set of features described in section 4.4.1. One of the used features is POS which we were only able to extract for the English version. Stanford is a NER system based on CRF model trained on CoNLL data collection. It incorporates long-distance information [53]. Stanford provides NER models for English and German. Unfortunately, we didn't find a suitable NER system for Dutch to compare with.

It can be observed that the CRF models achieve better precision at the expense of recall. Low recall sometimes leads to extracting no toponyms from the property description and hence the country of that property is misclassified. This results in a poor disambiguation results.

[7] http://nlp.stanford.edu/software/CRF-NER.shtml

Table 5.4: Extraction and disambiguation results for all versions.

	English			
	Pre.	**Rec.**	**F1**	**Dis.**
Ground Truth	**1**	**1**	**1**	79.1349
HMM(0.1)	0.3631	**0.8659**	0.5116	75.0636
HMM(0.1)+SVM(All)	0.7726	0.8014	0.7867	**79.8982**
HMM(0.9)	0.6638	0.7806	0.7175	78.3715
HMM(0.9)+SVM(All)	0.8275	0.7591	**0.7918**	79.3893
Stanford NER	0.8375	0.4365	0.5739	58.2697
CRF(0.9)	**0.9383**	0.6205	0.7470	69.4656

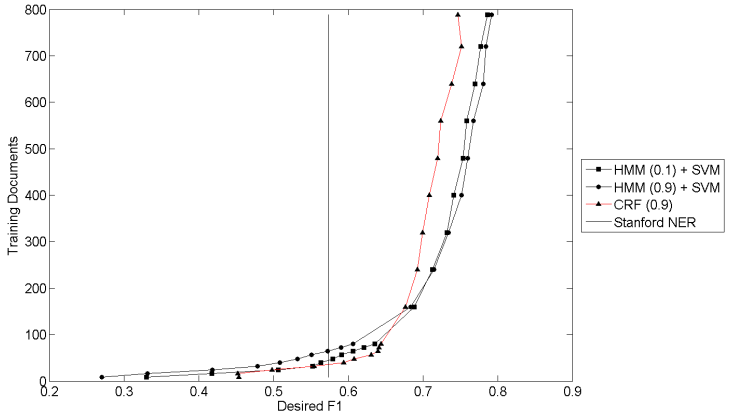
	German			
	Pre.	**Rec.**	**F1**	**Dis.**
Ground Truth	**1**	**1**	**1**	**81.4249**
HMM(0.1)	0.3399	**0.8306**	0.4824	79.3893
HMM(0.1)+SVM(All)	0.6722	0.7321	0.7009	79.6438
HMM(0.9)	0.6169	0.7085	0.6595	77.8626
HMM(0.9)+SVM(All)	**0.7414**	0.6876	**0.7135**	77.3537
Stanford NER	0.5351	0.2723	0.3609	40.4580

	Dutch			
	Pre.	**Rec.**	**F1**	**Dis.**
Ground Truth	**1**	**1**	**1**	**73.0280**
HMM(0.1)	0.2505	**0.8128**	0.3830	68.4478
HMM(0.1)+SVM(All)	0.6157	0.6872	**0.6495**	70.4835
HMM(0.9)	0.4923	0.6713	0.5680	67.1756
HMM(0.9)+SVM(All)	**0.6762**	0.6197	0.6467	67.6845

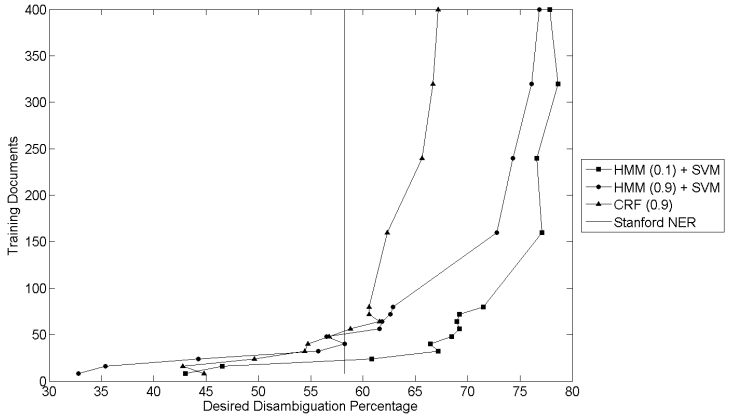
Table 5.4 also shows the robustness of our approach against different HMM thresholding settings. We used two different cutting thresholds (0.1, 0.9) for HMM. It is clear that our approach improves the precision and F1 measure on both cases.

5.4.5 Low Training Data Robustness

Robustness across different languages and using different cutting probability threshold is shown in the previous sections. In this section we want to prove the third aspect of robustness of our system which is its capability to work



(a) F1



(b) Disambiguation

Figure 5.4: The required training data required to achieve desired extraction and disambiguation results.

even with limited training samples. Figures 5.4a and 5.4b shows the required size of training data to achieve a desired result for F1 and disambiguation re-

spectively (applied on the English collection). It can be observed that our approach requires low number of training data to outperform our competitors the CRF and Stanford NER. Only 160 annotated documents are required to achieve 0.7 F1 and 75% correct disambiguation and to outperform the CRF. Much less documents are required to outperform the CRF disambiguation results as we mentioned before that the high precision of CRF systems is accompanied by low recall leading to poor disambiguation results.

5.5 Conclusions and Future Directions

In this chapter, we introduced a hybrid approach for toponym extraction and disambiguation. We used a HMM for extraction and a SVM classifier to classify the HMM output into false positive and true positive toponyms. Informativeness features beside coherence features derived from disambiguation results were used to train the SVM. Experiments were conducted with a set of holiday home descriptions with the aim to extract and disambiguate toponyms. Our system is proved to be robust on three aspects: language differences, high and low HMM threshold settings, and limited training data. It also outperforms the state-of-the-art methods of NER.

In the next part of the thesis, we plan to extend our approach for other types of named entities on the domain of short informal Twitter messages. We claim that this approach is also robust against domain differences and can be adapted to suit any kind of named entities. To achieve this it is required to develop a mechanism to find false positives among the extracted named entities. Coherency measures can be used to find highly ambiguous named entities.

Part III

Named Entities in Informal Text of Tweets

CHAPTER 6

Related Work

6.1 Summary

In the previous part of the thesis, we discussed the interdependency of the extraction and disambiguation processes on toponyms. We showed the robustness of our approach across different languages, different extraction approaches (rule-based and statistical) and different extraction settings. In this part, we want to prove the validity of our claims on other types of named entities and other domain. Informal short text is a domain which would benefit from the proposed *reinforcement effect* approach due to the unreliability of the traditional features like POS tags and capitalization. The structure of this part is as follows: the related work for NEE and NED for both formal and informal text is presented in this chapter. In chapter 7, we present a proof of concept for NEE and NED interdependency on tweets. It describes an unsupervised approach for extraction and disambiguation. Chapter 8 presents a generic open world approach for NED for tweets. Our approach links entity mention to any page on the web to serve as 'homepage' if there is no suitable one in a knowledge base. Finally, chapter 9 presents TwitterNEED, a hybrid supervised approach for NEE and NED for tweets. It makes use of features derived from the disambiguation phase to help improving the extraction of named entities in a supervised way.

6.2 Named Entity Disambiguation

6.2.1 For Formal Text

NED in web documents is a topic that is well covered in literature. Several approaches use Wikipedia or a KB derived from Wikipedia (like DBpedia and

YAGO) as entity store to look-up the suitable entity for a mention.

One of the earliest approaches was proposed by Bunescu et al. [58]. The authors developed a named entity disambiguation system that does disambiguation on two steps. First, it detects whether a proper name refers to a named entity included in the dictionary (detection). Second, it disambiguates between multiple named entities that can be denoted by the same proper name (disambiguation). Furthermore, authors defined a similarity measure that compared the context of a mention to the Wikipedia categories of an entity candidate. Cucerzan [59] proposes a large-scale system for disambiguating named entities based on information extracted from Wikipedia and web search results. The system uses the data associated with the known surface forms identified in a document and all their possible entity disambiguations to maximize the agreement between the context data stored for the candidate entities and the contextual information in the document, and also, the agreement among the category tags of the candidate entities.

The importance of entity-entity coherence measure in disambiguation is introduced by Kulkarni et al. [60]. Similarly, Hoffart et al. [61] combine three measures: the prior probability of an entity being mentioned, the similarity between the contexts of a mention and a candidate entity, as well as the coherence among candidate entities for all mentions together. AIDA [1] [62] is a system built on Hoffart's [61] approach.

Ad-hoc (entity oriented) NED represents another direction in NED research. Ad-hoc entities do not exist in a KB such as DBpedia, Freebase or YAGO. Instead of using a KB, given the candidate mentions of all the target entities, entity oriented disambiguation approaches determine which ones are true mentions of a target entity. Examples for such approach are presented in [63] and [64]. In [63], Srinivasan et al. proposed a cross document person name disambiguation system that clusters documents so that each cluster contains all and only those documents referring to the same person. They introduced features based on topic models and also document-level entity profiles sets of information that are collected for each ambiguous person in the entire document. In [64] Wang et al. introduced disambiguation techniques that require no knowledge about the targeted entities except their names. They proposed a graph-based model called MentionRank to leverage the homogeneity constraint and disambiguate the candidate mentions collectively across the document. Leveraging the homogeneity constraint of the entities is done in three ways: context similarity, co-mentioned entities, and cross-document, cross-

[1] https://d5gate.ag5.mpi-sb.mpg.de/webaida/

entity interdependence.

6.2.2 For Informal Short Text

NED in tweets has attracted researchers recently. Most of these researches investigate the problem of entity oriented disambiguation. Within this theme, [65], [66] and [67] focus on the task of filtering Twitter posts containing a given company name, depending on whether the post is actually related with the company or not. They develop a set of features (co-occurrence, Web-based features, Collection-based features) to find keywords for positive and negative cases. Similarly, [68] propose a topic centric entity extraction system where interesting entities pertaining to a topic are mined and extracted from short messages and returned as search results on the topic.

A supervised approach for real time NED in tweets is proposed by [69]. They focused on the problem of continually monitoring the Twitter stream and predicting whether an incoming message containing mentions indeed refers to a predefined entity or not. The authors propose a three-stage pipeline technique. In the first stage, filtering rules (colocations, users, hash tags) are used to identify clearly positive examples of messages truly mentioning the real world entities. These messages are given as input to an Expectation-Maximization method on the second stage, which produces training information to be used during the last stage. Finally, on the last stage they use the training set produced by the previous stage to classify unlabeled messages in real time. Another real time analysis tool proposed by [70]. The authors provide a browser extension which is based on a combination of several third party NLP APIs in order to add more semantics and annotations to Twitter and Facebook microposts.

Similar to our problem discussed on chapter 8, is the problem of entity home page finding which was part of TREC web and entity tracks. The task is to extract target entity and find its home page given an input entity, the type of the target entity and the relationship between the input and the target entity. One of the proposed approaches for this task was [71]. The authors combine content information with other sources as diverse as inlinks, URLs and anchors to find entry page. Another approach for entity home page recognition was introduced by [72]. It selects the features of link or web page content, and constructs entity homepage classifiers by using three kinds of machine learning algorithms of Logistic, SVM, AdaBoost to discover the optimal entity homepage.

Although the TREC problem looks similar to ours, the tweets' short informal nature makes it more tricky to find entity reference page. Moreover, distinguishing entities that could be linked to Wikipedia pages (Wiki-entities) from entities that only have a normal homepage or profile page (Non-Wiki entities), adds another challenge to our problem.

6.3 Named Entity Extraction

Few research efforts studied NEE on tweets. Researchers either used off-the-shelf trained NLP tools known for formal text (like part of speech tagging and statistical methods of extraction) or retrained those techniques to suit informal text of tweets.

In [73], authors built a NLP pipeline to perform NER. The pipeline involves part of speech tagging, shallow parsing, a novel SVM classifier that predicts the informativeness of capitalization in a tweet. Then it trains a CRF model with all the above features for NEE. For classification, LabeledLDA is used with entity types represent classes or topics. Bag of words generated for each entity type and same is done with each extracted mention. Classification done based on comparison of two.

The contextual relationship between the microposts is considered by [74]. The paper proposes merging the microtexts by discovering contextual relationship between the microtexts. A group of microtexts contextually linked with each other is regarded as a microtext cluster. Once this microtext cluster is obtained, authors expect that the performance of NER would be better. The authors provide some suggestions for Contextual closure, Microtext cluster, Semantic closure, Temporal closure, and Social closure. Those closures are used by Maximum Entropy for the NER task.

Similarly, [75] exploits the gregarious property in the local context derived from the Twitter stream. The system first leverages on the global context obtained from Wikipedia and Web N-Gram corpus to partition tweets into valid segments (phrases) using a dynamic programming algorithm. Each tweet segment is a candidate NE. Afterward, a ranking approach tries to rank segments according to their probability of being a NE. The highly-ranked segments have a higher chance of being true NEs. Each segment is represented as a node in a graph, and using the Wikipedia and the context of tweet (adjacent nodes (segments)), a score is assigned to that segment if it is a NE or not.

The Concept Extraction Challenge held as part of the Making Sense of Microposts Workshop (#MSM2013) [7] examines NEE task for tweets. Within this

challenge, set of approaches are used for the extraction task. Antal van Den Bosch et al. [76] proposed a memory-based tagging based on k-NN classification. Two taggers (capitalized and lowercased) were applied and their intersection is taken as a result. Marieke van Erp et al. [77] used textual features like POS, initial capital, suffix along with the decisions made by different of-the-shelf extractors (like AlchemyAPI[2], DBpedia Spotligh[3], Opencalais[4], Wikimeta[5], Stanford NER[6], Ritter Twitter NER [7]) to train SVM to do the extraction task. Similarly, Dlugolinsky et al. [78] Godin et al. [79] combined various existing NER taggers, along with other features to train different classifier. Further details about this challenge and its results are shown in appendix B.

[2] http://www.alchemyapi.com/
[3] thttps://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki
[4] http://www.opencalais.com/
[5] http://www.wikimeta.com/
[6] http://nlp.stanford.edu/ner/
[7] https://github.com/aritter/twitter_nlp

CHAPTER 7

Unsupervised Approach

7.1 Summary

In the previous part, we investigated the potential of *the reinforcement effect* on toponyms as a type of named entities. In this part, we want to examine the robustness of this approach on the domain of short informal messages. Short context messages (like tweets and SMS's) are a potentially rich source of continuously and instantly updated information. Shortness and informality of such messages are challenges for Natural Language Processing tasks. Most efforts done in this direction rely on machine learning techniques which are expensive in terms of data collection and training. In this chapter, we present an unsupervised semantic-driven approach to improve the extraction process by using clues from the disambiguation process. For extraction, we used a simple Knowledge-Base matching technique combined with a clustering-based approach for disambiguation. Experimental results on a self-collected set of tweets (as an example of short context messages) show improvement in extraction results when using unsupervised feedback from the disambiguation process.

The contents of this chapter have been published as [80].

7.2 Introduction

The rapid growth in IT in the last two decades has led to a growth in the amount of information available on the World Wide Web. A new style for exchanging and sharing information is short context. Examples for this style of text are tweets, social networks' statuses, SMS's, and chat messages.

In this part of the thesis, we use Twitter messages as a representative example of short informal context. Twitter is an important source for continuously and instantly updated information. The average number of tweets exceeds 400 million tweet per day sent by over 140 million active users around the world[1]. These numbers are growing exponentially. This huge number of tweets contains a large amount of unstructured information about users, locations, events, etc.

Information Extraction (IE) is the research field which enables the use of such a vast amount of unstructured distributed information in a structured way. IE systems analyze human language text in order to extract information about pre-specified types of events, entities, or relationships. Named entity *extraction* (NEE) is a subtask of IE that seeks to locate and classify atomic elements (mentions) in text belonging to predefined categories such as the names of persons, locations, etc. While named entity *disambiguation* (NED) is the task of exploring which correct person, place, event, etc. is referred to by a mention.

NEE & NED processes on short messages are basic steps of many SMS services such as our motivating application presented in appendix A where users' can use mobile messages to share information. NLP tasks on short context messages are very challenging. The challenges come from the nature of the messages. For example: (a) Some messages have limited length of 140 characters (like tweets and SMS's). (b) Users use acronyms for entire phrases (like LOL, OMG and b4). (c) Words are often misspelled, either accidentally or to shorten the length of the message. (d) Sentences follow no formal structure.

Few research efforts studied NEE on tweets (see chapter 6). Researchers either used off-the-shelf trained NLP tools known for formal text (like part of speech tagging and statistical methods of extraction) or retrained those techniques to suit informal text of tweets. Training such systems requires annotating large datasets which is an expensive task.

NEE and NED are highly dependent processes. In the first part of this thesis, we showed this interdependency in one kind of named entity (toponyms). We proved that the effectiveness of extraction influences the effectiveness of disambiguation, and reciprocally, the disambiguation results can be used to improve extraction. The idea is to have an extraction module which achieves a high recall; clues from the disambiguation process are then used to discover false positives. We called this behavior *the reinforcement effect*.

Contribution: In this chapter, we propose an unsupervised approach to prove the validity of the *reinforcement effect* on short informal text. Our ap-

[1] `https://blog.twitter.com/2012/twitter-turns-six`

proach uses Knowledge-Base (KB) look-up (here we use YAGO [6]) for entity mention extraction. This extraction approach achieves high recall and low precision due to many false positive matches. After extraction, we apply a cluster-based disambiguation algorithm to find coherent entities among all possible candidates. From the disambiguation results we find a set of isolated entities which are not coherent to any other candidates. We consider the mentions of those isolated entities as false positives and therewith improve the precision of extraction. Our approach is considered unsupervised as it doesn't require any training data for extraction or disambiguation.

Furthermore, we propose an idea to solve the problem of lacking context needed for disambiguation by constructing profiles of messages with the same hashtag or messages sent by the same user. Figure 7.1 shows our approach on tweets as an example for short messages.

Assumptions: In this chapter, we made the following assumptions:

1. We consider the KB-based NEE process as a basic predecessor step for NED. This means that we are only concerned with named entities that can be disambiguated. NED cannot be done without a KB to look-up possible candidates of the extracted mentions. Thus, we focus on famous named entities like players, companies, celebrities, locations, etc.

2. We assume the messages to be informative (i.e. contains some useful information about one or more named entities). Dealing with noisy messages is not within our scope.

7.3 Unsupervised Approach

In this chapter, we use YAGO KB for extraction as well as disambiguation processes. YAGO is built on Wikipedia, WordNet, and GeoNames. It contains more than 447 million facts for 9.8 million entities. A fact is a tuple representing a relation between two entities. YAGO has about 100 relations, such as `hasWonPrize`, `isKnownFor`, `isLocatedIn` and `hasInternalWikipediaLinkTo`. Furthermore, it contains relations connecting mentions to entities such as `hasPreferredName`, `means`, and `isCalled`. The `means` relation represents the relation between the entity and all possible mention representations in Wikipedia. For example the mentions {'*Chris Ronaldo*', '*Christiano*', '*Golden Boy*', '*Cristiano Ronaldo dos Santos Aveiro*'} and many more are all related to the entity '*Christiano_Ronaldo*' through the `means` relation.

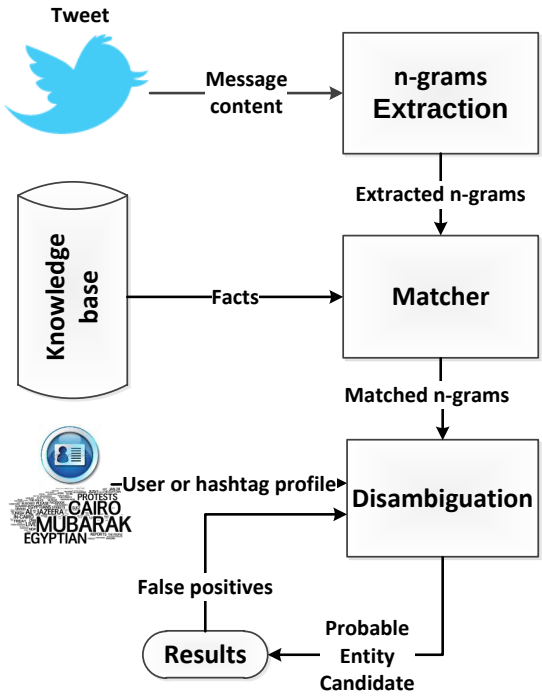


Figure 7.1: Proposed Unsupervised Approach for Twitter NEE & NED.

7.3.1 Named Entity Extraction

The list look-up strategy is an old method of performing NEE by scanning all possible n-grams of a document content against the mentions-entities table of a KB like YAGO or DBpedia [81]. Due to the short length of the messages and the informal nature of the used language, KB look-up is a suitable method for short context NEE.

The advantages of this extraction method are:

1. It prevents the imperfection of the standard extraction techniques (like POS) which perform quite poorly when applied to tweets [73].

2. It can be applied on any language once the KB contains named entity

(NE) representations for this language.

3. It is able to cope with different representations for a NE. For example consider the tweet "*fact: dr. william moulton marston, the man who created wonder woman, also designed an early lie detector*", standard extractors might only be able to recognize either '*dr. william moulton marston*' or '*william moulton marston*' but not both (the one that maximizes the extraction probability). Extraction of only one representation may cause a problem for the disambiguation when matching the extracted mention against the KB which may contain a different representation for the same entity. We followed the longest match strategy for mentions extraction.

4. It is able to find NEs regardless of their type. In the same example, other extractors may not be able to recognize and classify '*wonder woman*' as a NE, although it is the name of a comic character and helps to disambiguate the mention '*william moulton marston*'.

On the other hand, the disadvantages of this method for NEE are:

1. Not retrieving correct NEs which are misspelled or don't match any facts in the KB.

2. Retrieving many false positives (n-grams that match facts in the KB but do not represent a real NE).

This results in a high recall and low precision for the extraction process. In this chapter, we provide a solution for the second disadvantage by using feedback from NED in an unsupervised manner for detecting false positives.

As we are concerned with NED, it is inefficient to annotate all the n-grams space as named entities to achieve recall of 1. To do NED, we still need a KB to look-up for the named entities.

7.3.2 Named Entity Disambiguation

NED is the process of establishing mappings between extracted mentions and the actual entities [61]. For this task, comprehensive gazetteers such as GeoNames or KBs such as DBpedia, Freebase, or YAGO are required to find entity candidates for each mention.

To prove the feasibility of using the disambiguation results to enhance extraction precision, we developed a simple disambiguation algorithm (see Algorithm 1). This algorithm assumes that the correct entities for mentions appearing in the same message should be related to each other in YAGO KB graph.

Algorithm 1 The disambiguation algorithm

input : $M = \{m_i\}$ set of extracted mentions, $R(m_i) = \{e_{ij} \in$ Knowledge base$\}$ set of candidate entities for m_i

output: $Clusters(perm_l) = \{c_j\}$ set of clusters of related candidate entities for permutation $perm_l$ where $|Clusters(perm_l)|$ is the minimum

$$Permutations = \{\{e_{1x}, \ldots, e_{nx}\} \mid \forall 1 \leq i \leq n \exists! x : e_{ix} \in R(m_i)\}$$

foreach *Permutation* $p_l \in Permutations$ **do**
| $Clusters(perm_l) = Agglomerative\_Clustering\{p_l\}$
end
Find $Clusters(perm_l)$ with minimum size

The input of the algorithm is the set of all candidate entities $R(m_i)$ for the extracted mentions m_i. The algorithm finds all possible $permutations$ of the entities. Each permutation includes one candidate entity for each mention. For each permutation $perm_l$, we apply a single-linkage agglomerative clustering to obtain a set of clusters of related entities ($Clusters(perm_l)$) according to YAGO KB. We determine $Clusters(perm_l)$ having minimum size.

The agglomerative clustering starts with each candidate in $perm_l$ as a separate cluster. Then it merges clusters that contains related candidates. Clustering terminates when no more merging is possible.

Two candidates for two different mentions are considered related if there exists a direct or indirect path from one to the other in YAGO KB graph. Direct paths are defined as follows: candidate e_{ij} is related to candidate e_{lk} if there exists a fact of the form <e_{ij}, `some relation`, e_{lk}>. For indirect relations, candidate e_{ij} is related to candidate e_{lk} if there exist two facts of the form <e_{ij}, `some relation`, e_{xy}>and a fact <e_{xy}, `some relation`, e_{lk}>. We refer to the direct and the indirect relation in the experimental results section with *relations of depth 1* and *relations of depth 2*.

We didn't go further than relations with length more than 2, because the time needed to build an entity graph grows exponentially with the increase in the number of levels. In addition, considering relations of a longer path is expected to group all the candidates in one cluster as they are likely to be related to each other through some intermediate entities.

Finding false positives: We select the winning $Clusters(p_l)$ as the one having minimum size. We expect to find one or more clusters that include almost

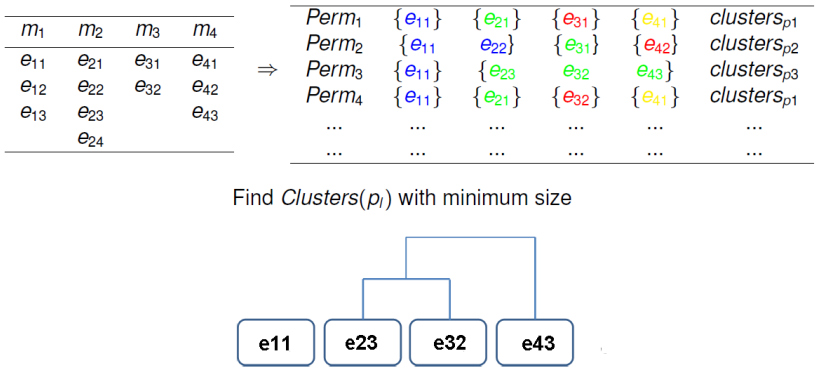


Figure 7.2: Example illustrates the agglomerative clustering disambiguation approach.

Table 7.1: Examples of NED output (Real mentions and their correct entities are shown in Bold)

Tweet	rt @breakingnews: explosion reported at a coptic church in alexandria, egypt; several killed - bbc.com	wp opinion: mohamed elbaradei •egypt's real state of emergency is its repressed democracy
Extracted mentions	**coptic church**, church in, killed, **egypt**, **bbc.com alexandria**, explosion, reported	state of emergency, **egypt**, opinion, real, **mohamed elbaradei**, repressed, democracy
Groups of related candidate entities	{**Coptic_Orthodox_Church_of_Alexandria**, **Alexandria**, **Egypt**, **BBC_News**}, {Churches_of_Rome},{Killed_in_action}, {Space_Shuttle_Challenger_disaster}, {Reported}	{State_of_emergency},{**Mohamed_ElBaradei**, **Egypt**}, {Repressed}, {Democracy_(play)}, {Real_(L'Arc-en-Ciel_album)}

all correct entities of all real mentions and other clusters each containing only one entity. Those clusters with size one contain most probably entities of false positive mentions.

Figure 7.2 shows how the agglomerative clustering algorithm works. The agglomerative clustering applied on permutation ($perm_3$) results in $Clusters(perm_3)$ with minimum size (2 clusters). We consider e_{23}, e_{32}, and e_{43} the correct references of mentions m_2, m_3, and m_4 respectively. While m_1 is considered a false positive as its representative entity candidate e_{11} ends in an individual cluster without being grouped with any other entity reference candidate.

Table 7.1 shows two examples for tweets along with the extracted mentions (using the KB look-up) and the clusters of related candidate entities. It can be observed that the correct candidate of real mentions are grouped in one cluster

Table 7.2: Evaluation of NEE approaches

	Lenient			Average			Strict		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
Stanford	1.0000	0.0076	0.0150	1.0000	0.0076	0.0150	1.0000	0.0076	0.0150
Stanford_lower	0.9091	0.1136	0.2020	0.8321	0.1032	0.1837	0.7538	0.0928	0.1653
Stanford_caseless	0.7818	0.8078	0.7946	0.7248	0.7461	0.7353	0.6673	0.6843	0.6757
KB_lu	0.4532	0.9713	0.6180	0.4178	0.9140	0.5735	0.3839	0.8566	0.5302
KB_lu + rod_1	0.8736	0.4627	0.6050	0.8339	0.4465	0.5816	0.7951	0.4302	0.5583
KB_lu + rod_2	0.5575	0.8528	0.6742	0.5178	0.8059	0.6305	0.4795	0.7591	0.5877

while false positives ended up alone in individual clusters.

Like the KB look-up extractor, this method of disambiguation can be applied on any language once the KB contains NE mentions for this language.

7.4 Experimental Results

Here we present some experimental results to show the effectiveness of using the disambiguation results to improve the extraction precision by discovery of false positives. We also discuss the weak points of our approach and give some suggestions for how to overcome them.

7.4.1 Dataset

We selected and manually annotated a set of 162 tweets that are found to be rich with NEs. This set is collected by searching in an open collection of tweets[2] for named entities that belong to topics like politics, sports, movie stars, etc. Messages are selected randomly from the search results. The set contains 3.23 NE/tweet on average.

Capitalization is a key orthographic feature for extracting NEs. Unfortunately in informal short messages, capitalization is much less reliable than in edited texts [73]. To simulate the worst case of informality of the tweets, we turned the tweets into lower case before applying the extractors.

7.4.2 Experiment

In this experiment we evaluate a set of extraction techniques on our dataset:

[1] http://wis.ewi.tudelft.nl/umap2011/#dataset

Table 7.3: Examples some problematic cases

Case #	Tweet Content
1	rt @wsjindia: india tightens rules on cotton exports http://on.wsj.com/ev2ud9
2	rt @imdb: catherine hardwicke is in talks to direct 'maze runners', a film adaptation of james dashner's sci-fi trilogy. http://imdb.to/

- **Stanford**: Stanford NER [53] trained on normal CoNLL collection.
- **Stanford_lower**: Stanford NER trained on CoNLL collection after converting all text into lower case.
- **Stanford_caseless**: Stanford NER caseless model that ignores capitalization.
- **KB_lu**: KB look-up.
- **KB_lu + rod_1**: KB look-up + considering feedback from disambiguation with *relations of depth 1*.
- **KB_lu + rod_2**: KB look-up + considering feedback from disambiguation with *relations of depth 2*.

The results are presented in table 7.2. The main observation is that the **Stanford** and **Stanford_lower** NER perform badly on our extraction task; while **Stanford_caseless** outperform all the other methods. As expected the KB look-up extractor is able achieve high recall and low precision; and feedback from the disambiguation process improved overall extraction effectiveness (as indicated by the F1 measure) by improving precision at the expense of some recall. Although our approach is unsupervised compared to other supervised approaches, its performance is still not far beyond the best performing supervised approach. Furthermore, our proposed approach is language independent in principle as we use no language features at all while other supervised approaches are all language dependent.

7.4.3 Discussion

In this section we discuss in depth the results and causes.

Capitalization is a very important feature that NEE statistical approaches rely on. Even training Stanford CRF classifier on lower case version of CoNLL does not help to achieve reasonable results.

KB_lu extractor achieves a high recall with low precision due to many false positives. While **KB_lu + rod_1** achieves high precision as it looks only for direct related entities like '*Egypt*' and '*Alexandria*'.

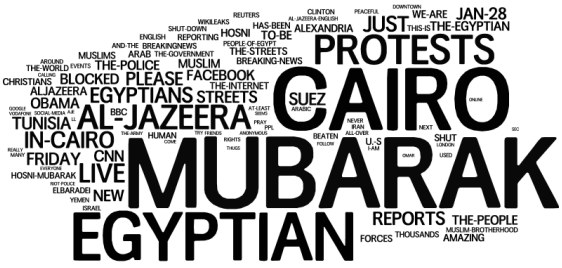
By increasing the scope of finding related entities to depth 2, **KB_lu + rod_2** finds more related entities and hence fails to discover some false positives. This leads to a drop in the recall and an enhancement in both precision and F1 measure (compared with **KB_lu**).

One major problem that harms recall is to have a message with an entity not related to any other NEs or to have only one NE within the message. Table 7.3 case 1 shows a message with only one named entity (india) that ends up alone in a cluster and thus considered false positive. A suggestion to overcome such problem is to expand the context by also considering messages replied to this submission or messages having the same hashtag or messages sent by the same user. It is possible to get enough context needed for the disambiguation process using user or hashtag profiles. Figures 7.3a, 7.3b and 7.3c show the word clouds generated for the hashtags '*Egypt*', '*Superbowl*' and for the user '*LizzieViolet*' respectively. Word clouds for hashtags are generated from the TREC 2011 Microblog Track collection of tweets [3]. This collection covers both the time period of the Egyptian revolution and the US Superbowl. The terms size in the word cloud proportionates the probability that the term is being mentioned in the profile tweets.

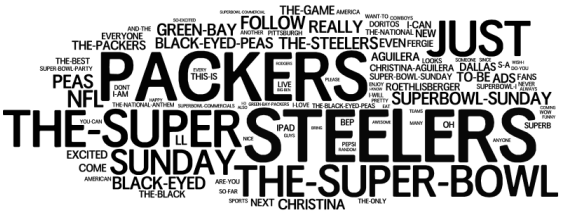
Another problem that harms precision is the entities like the '*United_States*' which is related to many other entities. In table 7.3 case 2, the mention '*talks*' is extracted as named entity. One of its entity candidates is '*Camp_David_Accords*' which is grouped with '*Catherine_Hardwicke*' as they both are related to the entity '*United_States*' (using **KB_lu + rod_2**). Both entities are related to '*United_States*' through relation of type `hasInternalWikipediaLinkTo`.

A suggestion to overcome this problem is to incorporate a weight representing the strength of the relation between two entities. This weight should be inversely proportional to the degree of the intermediate entity node in the KB graph. In our example the relation weight between '*Camp_David_Accords*' and '*Catherine_Hardwicke*' should be very low because they are related together through '*United_States*' which has a very high number of edges connected to its node in the KB graph.

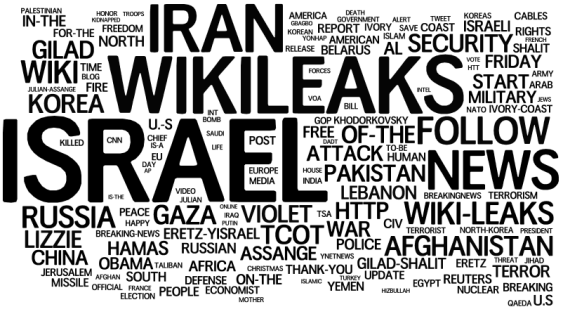
[3] http://trec.nist.gov/data/tweets/



(a) #Egypt



(b) #Superbowl



(c) user LizzieViolet

Figure 7.3: Words clouds for some hashtags and user profiles

7.5 Conclusions and Future Directions

In this chapter, we introduced an approach for unsupervised improvement of Named Entity Extraction (NEE) in short context using clues from Named En-

tity Disambiguation (NED). To show its effectiveness experimentally, we chose an approach for NEE based on knowledge base look-up. This method of extraction achieves high recall and low precision. Feedback from the disambiguation process is used to discover false positives and thereby improve the precision and F1 measure.

In the rest of this part of the thesis, we aim to enhance our results by considering a wider context than a single message for NED. Furthermore, we would like to overcome the limitations of using KBs in the disambiguation process by looking to the web for disambiguating named entities. We would like also to move to supervised approaches for entity extraction and to validate our claims on other datasets.

CHAPTER 8

Generic Open World Disambiguation Approach

8.1 Summary

Social media is a rich source of information. To make use of this information, it is sometimes required to extract and disambiguate named entities. In the previous chapter we presented an unsupervised approach for named entity Extraction (NEE) and disambiguation (NED) that uses entities coherency for disambiguation. In this chapter, we focus only on named entity disambiguation (NED) in Twitter messages. As concluded from the previous chapter, NED in tweets is challenging in two ways. First, the limited length of tweet makes it hard to have enough context while many disambiguation techniques depend on it. The second is that many named entities in tweets do not appear in a knowledge base (KB). We share ideas from information retrieval (IR) and NED to propose solutions for both challenges. For the first problem we make use of the gregarious nature of tweets to get enough context needed for disambiguation. For the second problem we look for an alternative home page if there is no Wikipedia page represents the entity. Given a mention, we obtain a list of Wikipedia candidates from YAGO KB in addition to top ranked pages from Google search engine. We use Support Vector Machine (SVM) to rank the candidate pages to find the best representative entities. Experiments conducted on two datasets show better disambiguation results compared with the baselines and a competitor.

The contents of this chapter have been published as [82].

Table 8.1: Some challenging cases for NED in tweets (mentions are written in bold).

Case #	Tweet Content
1	**Qld** flood victims donate to **Vic** bushfire appeal
2	**Laelith Demonia** has just defeated **liwanu Hird**. Career wins is 575, career losses is 966.
3	Adding **Win7Beta**, **Win2008**, and **Vista** x64 and x86 images to munin. #wds
4	"Even Writers Can Help..An Appeal For **Australian** Bushfire Victims" http://cli.gs/Zs8zL2

8.2 Introduction

Named entity disambiguation (NED) is the task of exploring which correct person, place, event, etc. is referred to by a mention. Wikipedia articles are widely used as entities' references. For example, the mention '*Victoria*' may refer to one of many entities like '`http://en.wikipedia.org/wiki/Victoria_(Australia)`' or '`http://en.wikipedia.org/wiki/Queen_Victoria`'. According to Yago KB [83] the mention '*Victoria*' may refer to one of 188 entities in Wikipedia.

NED in tweets is challenging. Here we summarize the challenges of that problem:

- The limited length (140 characters) of tweets forces the senders to provide dense information. Users resort to acronyms to reserve space. Informal language is another way to express more information in less space. All of these problems make the disambiguation more complex. For example, case 1 in table 8.1 shows two abbreviations ('*Qld*' and '*Vic*'). It is hard to infer their entities without extra information.

- The limited coverage of KB is another challenge facing NED. According to [5], 5 million out of 15 million mentions on the web could not be linked to Wikipedia. This means that relying only on KB for NED leads to around 33% loss in disambiguated entities. This percentage becomes higher on Twitter because of its social nature where people talk more about infamous entities. For example, case 2 in table 1.1 contains two mentions for two users on '*My Second Life*' social network. One would never find their entities in a KB but their profile pages ('`https://my.secondlife.com/laelith.demonia`' and '`https:`

//my.secondlife.com/liwanu.hird') can be easily found by any search engine.

- Named entity (NE) representation in KB implies another NED challenge. Yago KB uses Wikipedia anchor text as possible mention representation for named entities. However, there might be more representations that do not appear in Wikipedia anchor text. Either because of misspelling or because of a new abbreviation of the entity. For example, in case 3 in table 1.1, the mentions '*Win7Beta*' and '*Win2008*' are not representing any entity in YAGO KB although they refer to the entities 'http://en.wikipedia.org/wiki/Windows_7' and 'http://en.wikipedia.org/wiki/Windows_Server_2008' respectively.

- The process of NED involves degrees of uncertainty. For example, case 4 in table 1.1, it is hard to assess whether '*Australian*' should refer to 'http://en.wikipedia.org/wiki/Australia' or 'http://en.wikipedia.org/wiki/Australian_people'. Both might be correct. This is why we believe that it is better to provide a list of ranked candidates instead of selecting only one candidate for each mention.

- A final challenge is the update of the KBs. For example, the page of '*Barack Obama*' on Wikipedia was created on 18 March 2004. Before that date '*Barack Obama* was a member of the Illinois Senate and you could find his profile page on 'http://www.ilga.gov/senate/Senator.asp?MemberID=747'. It is very common on social networks that users talk about some infamous entity who might become later a public figure.

According to a literature survey (see section chapter 6), almost all researchers use KBs entities as references for NED. Some of those researchers assign *null* to mentions with no possible reference entity and others assign an entity to a mention once it is in the dictionary containing all candidates for surface strings even if the correct one is not in the entity repository. Furthermore, researches that studied NED in tweets are mostly entity oriented (i.e. given an entity like '*Apple Inc*', it is required to classify the mention '*Apple*' if it is a correct representative for that entity or not).

In our opinion, for the NED task in tweets, it is necessary to have a generic system that doesn't rely only on the closed world of KBs in the disambiguation process. We also believe that the NED task involves degrees of uncertainty. In this chapter, we propose a generic open world NED approach that shares ideas from NED and IR.

Given a tweet mention, we get a set of possible entity candidates' home pages by querying YAGO KB and Google search engine. We query Google to get possible candidate entities' home pages. We enrich the candidate list by querying YAGO KB to get Wikipedia articles' candidates.

For each candidate, we extract a set of context and URL features. Context features (like language model and tweet-document overlapped terms) measure the context similarity between mention and entity candidates. URL features (like path length and mention-URL string similarity) measure how likely the candidate URL could be a representative to the entity home page. In addition we use the prior probability of the entity from YAGO KB. An SVM is trained on the aforementioned features and used to rank all candidate pages.

Wikipedia pages and home pages are different in their characteristics. Wikipedia pages tend to be long, while home pages tend to have short content. Sometimes it has no content at all but a title and a flash introduction. For this reason we train the SVM to distinguish between three types of entity pages, a Wikipedia page (Wiki entity), a non-Wikipedia home page (Non-Wiki entity), and a non-relevant page.

Furthermore, we suggested an approach to enrich the context of the mention by adding frequent terms from other targeted tweets. Targeted tweets are a set of tweets talking about same event. This approach improves the recognition of Non-Wiki entities.

We conduct experiments on two different datasets of tweets having different characteristics. Our approach achieves better disambiguation results on both sets compared with the baselines and a competitor.

Contributions: This chapter makes the following contributions:

- We propose a generic approach for NED in tweets for any named entity (not entity oriented).
- Mentions are disambiguated by assigning them to either a Wikipedia article or a home page.
- Instead of just selecting one entity for each mention we provide a ranked list of possible entities.
- We improve NED quality in tweets by making use of the gregarious nature of targeted tweets to get enough context needed for disambiguation.

The rest of the chapter is organized as follows. Section 8.3 presents our generic approach for NED in tweets. In section 8.4, we describe the experimental setup, present its results, and discuss some observations and their con-

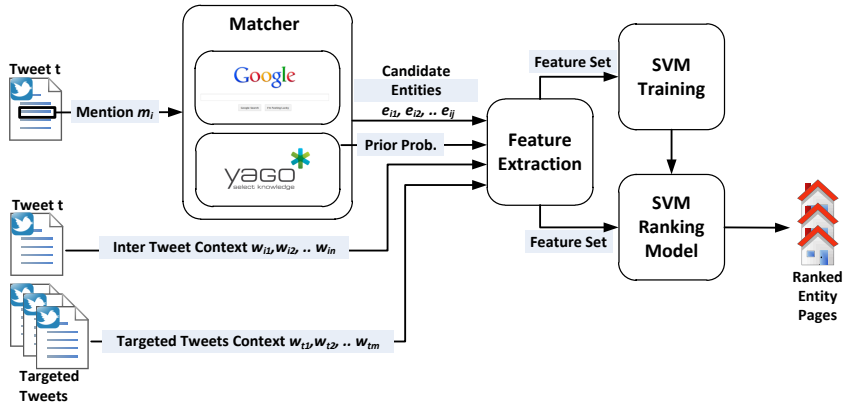


Figure 8.1: System Architecture.

sequences. Finally, conclusions and future research directions are presented in section 8.5.

8.3 Generic Open World Approach

We can conclude from the literature review presented in chapter 6 that almost all NED approaches in tweets are entity oriented. (i.e. given the entity it is required to check if a given mention or given tweet is relevant to the input entity or not). In contrast, we present a generic open world approach for NED for any named entity based on the mention context and with support from targeted tweets if available.

First of all let us formalize the problem. Given a mention m_i that belongs to tweet t, the goal is to find a ranked list of entities' home pages e_{ij} that m_i represents. We make use of the context of the mention $\{w\} = \{m_i, w_1, w_2, ..w_n\}$ to find the best entity candidate. $\{w\}$ is the set of words in the tweet after removing the stop words. A set of features is extracted from each e_{ij} measuring how relative is it to m_i and its context. An SVM is trained over training set of manually annotated mentions and used for ranking of entity pages for unseen mentions.

Figure 8.1 illustrates the whole process of NED in tweets. The system is

composed of the three modules; the matcher, the feature extractor, and the SVM ranker.

8.3.1 Matcher

This module contains two sub-modules: Google API, and YAGO KB. Google API is a service provided by Google to enable developers from using Google products from their applications. YAGO KB is built on Wikipedia. It contains more than 447 million facts for 9.8 million entities. A fact is a tuple representing a relation between two entities. YAGO has about 100 relation types, such as `hasWonPrize`, `isKnownFor`, and `isLocatedIn` . Furthermore, it contains relation types connecting mentions to entities such as `hasPreferredName`, `means`, and `isCalled`. The `means` relation represents the relation between the entity and all possible mention representations in Wikipedia. For example, the mentions {'*Chris Ronaldo*', '*Christiano*', '*Golden Boy*', '*Cristiano Ronaldo dos Santos Aveiro*'} and many more are all related to the entity '`http://en.wikipedia.org/wiki/Cristiano_Ronaldo`' through the `means` relation.

This module takes the mention m_i and looks for its appropriate web pages using Google API. A list of top 18 web pages retrieved by Google is crawled. To enlarge the search space, we query YAGO KB for possible entities for that mention. Instead of taking all candidate entities related to that mention, we just take the set of candidates with top prior probabilities. Prior probability represents the popularity for mapping a name to an entity. YAGO calculates those prior by counting, for each mention that constitutes an anchor text in Wikipedia, how often it refers to a particular entity. We sort the entities in descending order according to their prior probability. We select the top entities satisfying the following condition:

$$\frac{Prior(e_{ij})}{Maximum(Prior(e_{ij}))} > 0.2 \tag{8.1}$$

In this way we consider a set of most probable entities regardless of their count instead of just considering fixed number of top entities.

For all the YAGO selected entities we add their Wikipedia articles to the set of Google retrieved web pages to form our search space for the best candidates for the input mention.

After crawling the candidate pages we apply a wrapper to extract its title, description, keywords and textual content. For this task we used HtmlUnit

Table 8.2: URL features.

Feature Name	Feature Description
URL Length	The length of URL.
Mention-URL Similarity	String similarity between the mention and the URL domain name (for non-Wikipedia pages) or the Wikipedia entity name (for Wikipedia pages) based on Dice Coefficient Strategy [84].
Is Mention Contained	Whether or not the mention is contained in the whole URL.
Google Page Rank	The page order as retrieved by Google. Wikipedia pages added from YAGO are assigned a rank after all Google retrieved pages.
Title Keywords	Whether or not page title contains keywords like ('Official', or 'Home page').
#Slashes	Path length of the page (i.e. number of slashes in the URL).

library[1].

8.3.2 Feature Extractor

This module is responsible for extracting a set of contextual and URL features that give the SVM indicators on how likely the candidate entity page could be a representative to the mention. The tweet is tokenized with a special tweet tokenizer [85]. Similarly, other target tweets (revolving the same event as the mention tweet) are tokenized and top frequent k words are added to the mention context. Only proper nouns and nouns are considered according to the part of speech tags (POS) generated by a special POS tagger designed for tweets [85]. Target tweets can be obtained by considering tweets with the same hashtag. Here, we just use the target tweets as provided in one of the two datasets we used in the experiments.

On the candidate pages side, for each candidate page we extract the following set of features:

- **Language Model (LM):** We used a smoothed unigram LM [86]. We treat

[1]http://htmlunit.sourceforge.net/

the mention along with its tweet keywords as a query and the entity pages as documents. The probability of a document being relevant to the query is calculated as follows:

$$logP(q|d) = \sum_{w \in q,d} log \frac{P_s(w|d)}{\alpha_d P(w|c)} + \sum_{w \in q} logP(w|c) + nlog\alpha_d \tag{8.2}$$

where $q = \{m_i, w_{i1}, ..w_{in}\}$, d is the e_{ij} candidate page, c is the collection of all the candidate pages for m_i, n is the query length and α_d is document length normalization factor, $P(w|c)$ is the collection LM and $P_s(w|d)$ is the Dirichlet conjugate prior [87]. These probabilities can be calculated as follows:

$$P(w|c) = \frac{tf(w,c)}{c_s} \tag{8.3}$$

$$P_s(w|d) = \frac{tf(w,d) + \mu P(w|c)}{|D| + \mu} \tag{8.4}$$

where tf is the term frequency of a word w in a document d or in the entire collection c, c_s is raw collection size (total number of tokens in the collection) and μ is a smoothing parameter that is calculated as the average document length in the collection c.

We calculated a separate LM for each of the entity pages parts (the title, description, keywords, and content).

- **Tweet-Page Overlap:** The difference in length between Wikipedia pages and non-Wikipedia pages in addition to the document length normalization in the LM led to favor short documents (non-Wikipedia pages) over long documents (Wikipedia pages). This is why we looked for another feature that does not favor documents based on its length. The feature Tweet-Page Overlap is inspired by Jaccard distance with disregarding lengths. This feature represents the count of the overlapped words between the query q and the document d. It can be calculated as follows:

$$Overlap(q,d) = |q \cap d|$$

Again 4 versions of this feature are calculated for pages title, description, keywords, and content.

- **Entity Prior Probability:** It is a value provided by YAGO KB as described in section 8.3.1. Only Wikipedia pages have Prior Probabilities. Non-Wikipedia pages are just assigned zero for this feature.

In addition to the context features we also extract a set of URL features shown in table 8.2.

8.3.3 SVM Ranker

After extracting the aforementioned set of features, an SVM classifier [88] with RBF kernel function is trained to rank candidate entities of a mention. The SVM is trained on three types of entity classes; Wikipedia home page, non-Wikipedia home page, and non-relevant page. The reason behind this is that the characteristics of Wikipedia home pages and non-Wikipedia home pages are different, and we don't want the classifier to get confused. In this way, the classifier would use the best set of features for each of the relevant classes. Wikipedia home pages have rich contents and thus context features would be better for calculating how the Wikipedia page is relevant to the mention context. While non-Wikipedia home pages tend to be short and sometimes with almost no content. In this case URL features might be more useful to find the relevant entity page of a mention.

Moreover, we automatically look into the Wikipedia page infobox for a home page URL for the entity. If found, we remove that home page from the candidate list. For example, for the mention '*Barcelona*', if we find among the candidate pages the Wikipedia page '`http://en.wikipedia.org/wiki/FC_Barcelona`' and we find in the infobox of this page that the official site for '*Barcelona*' is '`http://www.fcbarcelona.com/`', we remove the latter page if found among the candidate pages. The idea behind this action is that our training data is annotated by assigning only one entity page for each mention with the priority for Wikipedia pages. We don't want to confuse the classifier by assigning a non-relevant class to a home page for one mention and assigning a relevant class for home page of another mention that doesn't have a Wikipedia entity.

The SVM is trained to provide three probabilities for the three mentioned classes. Due to the imbalance in the training data between the first two classes and the third (only one page is assigned to the mention and the rest is treated as

Table 8.3: Candidate Pages for the mention '*Houston*'.

http://www.houstontx.gov/
http://en.wikipedia.org/wiki/Houston
http://www.visithoustontexas.com/
http://www.chron.com/
http://www.tripadvisor.com/Tourism-g56003-Houston_Texas-Vacations.html
http://www.forbes.com/places/tx/houston/
http://www.nba.com/rockets/
http://www.uh.edu/
http://www.houstontexans.com/
http://www.houston.org/
http://www.citypass.com/houston
http://www.portofhouston.com/
http://www.hillstone.com/
http://wikitravel.org/en/Houston
http://houston.craigslist.org/
http://houston.astros.mlb.com/

non-relevant page), the probabilities of majority class (non-relevant) are dominating. Dealing with the task as a ranking task instead of hard classification enables us to overcome this problem.

For testing and evaluating, we rank the mentions candidate pages according to the highest probabilities of the two relevant classes. Evaluation is done by looking at the quality of finding the correct entity page of the mention at top k rank.

8.3.4 Targeted Tweets

Due to the limitation of tweet context which sometimes affect the disambiguation process, we introduce an improvement by making use of the gregarious nature of tweets. Given a targeted set of tweets (tweets about the same topic), we find the most frequent nouns and add those terms to the context of each tweet in the targeted set. This approach improves the recognition of Non-Wiki entities as will be shown in the next section.

8.4 Experimental Results

8.4.1 Datasets

To validate our approach, we use two Twitter datasets[2]. The two datasets are mainly designed for named entity recognition (NER) task. Thus to build our ground truth we only annotated each NE with one appropriate entity page. We gave higher priority to Wikipedia pages. If Wikipedia has no page for the entity we link it to a home page or profile page. The first dataset (Brian Collection) is the one used in [89]. The dataset is composed of four subsets of tweets; one public timeline subset and three subsets of targeted tweets revolving around economic recession, Australian Bushfires and gas explosion in Bozeman, MT. The other dataset (Mena Collection) is the one used in in chapter 7 which is relatively small in size of tweets but rich in number of NEs. It is composed mainly from tweeted news about players, celebrities, politics, etc. Statistics about the two datasets are shown in table 8.4. The two collections are good representative examples for two types of tweets: the formal news titles tweets (Mena Collection) and the users targeted tweets that discuss some events (Brian Collection).

8.4.2 Experimental Setup

Our evaluation measure is the accuracy of finding the correct entity page of a mention at rank k. We consider only top 5 ranks. The reason behind focusing on recall instead of precision is that we can't consider other retrieved pages as a non-relevant (false positives). In some cases, there may exist more than one relevant page among the candidate pages for a given mention. So that, as we link each mention to only one entity page, it is not fair to consider other pages as a non-relevant pages. For example, table 8.3 shows some candidate pages for the mention '*Houston*'. Although we link this mention to the Wikipedia page `http://en.wikipedia.org/wiki/Houston`, we could not consider other pages (such as `http://www.houstontx.gov/` and `http://wikitravel.org/en/Houston`) that appear in the top k ranks as non-relevant pages.

All our experiments are done through a 4-fold cross validation approach for training and testing the SVM.

[2]Our datasets are available at `https://github.com/badiehm/TwitterNEED`

Table 8.4: Datasets Statistics.

	Brian Col.	Mena Col.
#Tweets	1603	162
#Mentions	1585	510
#Wiki Entities	1233(78%)	483(94%)
#Non-Wiki Entities	274(17%)	19(4%)
#Mentions with no Entity	78(5%)	8(2%)
#Avg Google rank for correct entity	9	5

Table 8.5: Baselines and Upper bounds.

	Brian Col.	Mena Col.
Prior	846(53%)	394(77%)
AIDA	766(48%)	389(76%)
Google 1st rank	269(17%)	197(39%)
YAGO coverage	990(62%)	449(88%)
Google coverage for:		
All entities	1218(77%)	476(93%)
Wiki entities	1077(87%)	462(96%)
Non-Wiki entities	141(51%)	14(74%)

8.4.3 Baselines and Upper bounds

Table 8.5 shows our baselines and upper bounds in terms of the percentage of correctly finding the entity page of a mention. Three baselines are defined. The first is *Prior*, which represents the disambiguation results if we just pick the YAGO entity with the highest prior for a given mention. The second is the *AIDA* disambiguation system. We used the system's RMI to disambiguate mentions. The third is *Google 1st rank* which represents the results if we picked the Google 1st ranked page result for the input mention. It might be surprising that *AIDA* gives worse results than one of its components which is *Prior*. The reason behind this is that *AIDA* matching of mentions is case sensitive and thus could not find entities for lower case mentions. It was not possible to turn all mentions to initials upper case because some mentions should be in all upper case to get matched (like 'USA'). For *Prior*, we do the match case insensitively. *AIDA* and *Prior* are upper bounded by the YAGO coverage for mentions entity. Coverage means how much mention-entity pairs of our ground truth exist

in the KB. Note that more mentions might have a Wikipedia entity but it is not covered in YAGO because it doesn't have the proper surface mention (like 'Win7Beta').

On the other hand, we have an upper bound we cannot exceed. The set of candidates retrieved by Google and enriched through KB does not cover our ground truth completely. Hence, we could not exceed that upper bound.

8.4.4 Feature Evaluation

To evaluate the importance of each of the two feature sets used, we conduct an experiment to measure the effect of each feature set on the disambiguation results. Figure 8.2 shows the disambiguation results on our datasets using each of the introduced feature sets. It also shows the effect of each feature sets on both types of entities, Wiki and Non-Wiki.

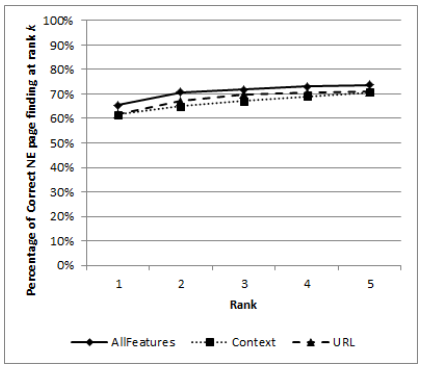
Figures 8.2c and 8.2d show that context features are more effective than URL features in finding Wiki entities. On the other side, figures 8.2e and 8.2f show the superiority of URL features over context features in finding Non-Wiki entities.

Although Wikipedia URLs are normally quite informative, the context features have more data to be investigated and used in the selection and ranking of candidate pages than the URL features. Furthermore, some Wiki URLs are not informative for the given mention. For example, the mention '*Qld*' refers to the Wikipedia entity '`http://en.wikipedia.org/wiki/Queensland`' which is not informative regarding the input mention. This is why context features are more effective than URL features in finding Wiki entities.

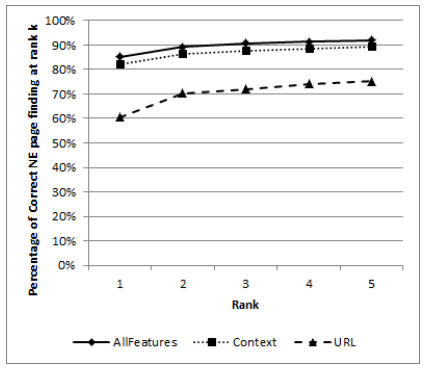
On the other hand, context features are less effective than URL features in finding Non-Wiki entities because many home pages nowadays are either developed in flash or have at least some flash and graphics contents and hence contains less textual content to be used.

All sub figures of figure 8.2 show that usage of both sets of features yields better entity disambiguation results. The only exception is the first two ranks in figure 8.2f. However, it is not an indicator for the failure of our claim as the number of Non-Wiki entities in Mena collection is very small (19 entities).

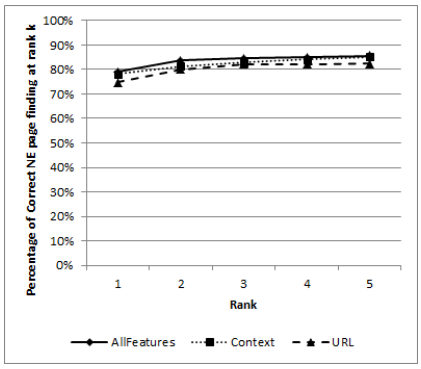
Compared to table 8.5, our approach shows improvements on the disambiguation quality for all entities by about 12% on Brian Collection and by about 8% on Mena Collection over the best baseline (prior) at rank $k = 1$. At rank $k = 5$, the improvements over the best baseline are 21% and 15% respectively.



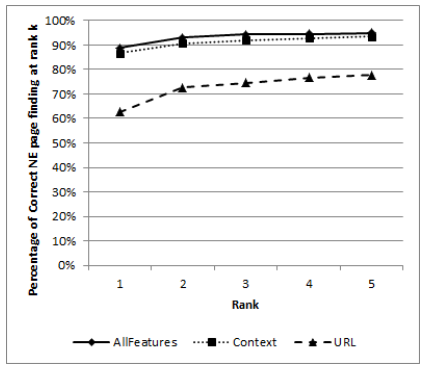
(a) Brian: All Entities



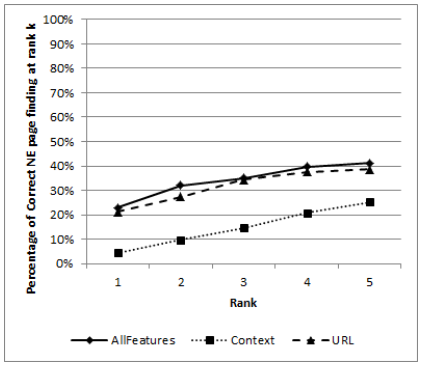
(b) Mena: All Entities



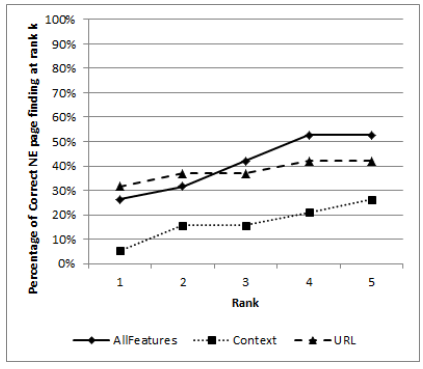
(c) Brian: Wiki Entities



(d) Mena: Wiki Entities



(e) Brian: Non-Wiki Entities



(f) Mena: Non-Wiki Entities

Figure 8.2: Disambiguation results at rank k using different feature sets.

Table 8.6: Top 10 frequent terms in Brian col. targeted tweets.

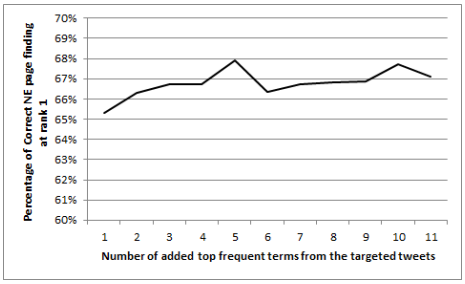
Bozeman Explosion	Australian Bushfires	Economic Recession	Public Timeline
bozeman, montana, bozexplod, mt, twitter, gov, boodles, schweitzer, nw, twitterers	bushfire, sitepoint, appeal, australia, victoria, aussie, coles, brumby, friday, vic	intel, reuters, u.s., fargo, job, san, denver, tuesday, wells, grad	twitter, la, youtube, god, black, mac, tx, iphone, itunes, queen

8.4.5 Targeted Tweets Improvement

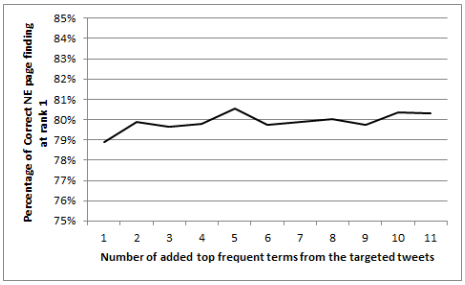
Due to the limitation of tweet context which sometimes affect the disambiguation process, we introduce an improvement by making use of the gregarious nature of tweets. Given a targeted set of tweets (tweets about the same topic), we find the most frequent nouns and add those terms to the context of each tweet in the targeted set. An experiment is performed on Brian collection to study the effect of the frequent terms on the disambiguation results. Table 8.6 shows top 10 frequent terms in each of the targeted sets. Figure 8.3 shows the disambiguation results at rank 1 over different top k frequent terms added from targeted tweets. The overall trend is that disambiguation results of all entities are improved by 2% on average by adding frequent terms to tweet context (see figure 8.3a). Non-Wiki entities in figure 8.3c make better use of the frequent terms and achieve improvement of about 4% to 5% on average. While Wiki entities in figure 8.3b achieve an improvement of about 1% only. The reason behind this is that Non-Wiki entities' pages are much shorter in contents so that an extra term in the tweet context helps more in finding the correct entity page.

8.5 Conclusions and Future Directions

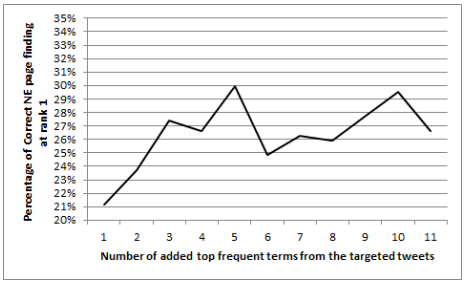
Named entity disambiguation is an important step to make better use of the unstructured information in tweets. NED in tweets is challenging because of the limited size of tweets and the absence of many mentioned entities from KBs. In this chapter, we introduce a generic open world approach for NED in tweets. The proposed approach is generic as it is not entity oriented. It is also open world because it is not limited by the coverage of a KB. We make use of a KB as well as Google search engine to find candidate set of entities' pages for each mention. Two sets of features (context and URL) are presented



(a) Brian: All Entities



(b) Brian: Wiki Entities



(c) Brian: Non-Wiki Entities

Figure 8.3: Disambiguation results over different top k frequent terms added from targeted tweets.

for better finding of Wiki and Non-Wiki entity pages. An SVM is used to rank entities' pages instead of assigning only one entity page for each mention. We are inspired by the fact that NED involves degree of uncertainty. We also in-

troduce a method to enrich a mention's context by adding top frequent terms from targeted tweets to the context of the mention.

Results show that context features are more helpful in finding entities with Wikipedia pages, while URL features are more helpful in finding entities with non-Wikipedia pages. Adding top frequent terms improves the NED results of Non-Wiki entities by about 4% to 5%.

For future improvement, we want to increase the upper bound of candidate pages coverage by re-querying Google search engine for mentions with no suitable candidate pages. In the next chapter, we want to integrate our NED model with a NEE model which makes use of the *reinforcement effect*.

CHAPTER 9

TwitterNEED: A Hybrid Extraction and Disambiguation Approach

9.1 Summary

In the previous chapter we presented our generic open world NED for tweets. It overcomes the problem of KB limitation by looking for entities home pages instead of just linking them to a KB. Furthermore, we proposed a method to enrich the context of the tweet by considering targeted tweets. In this chapter, we present TwitterNEED, a hybrid and robust approach for Named Entity Extraction (NEE) and Disambiguation (NED) for tweets. Although NEE and NED are two topics that are well studied in literature, almost all approaches treat the two problems separately. We believe that the two processes are interdependent. Hence, disambiguation can help in improving the quality of the extraction process with a feedback. This reduces the error propagation on the whole system. Our extraction approach first tries to achieve a high extraction recall by finding all possible uncertain mention candidates. Then Support Vector Machine (SVM) filters the extracted candidates into true positives and false positives using features derived from the disambiguation phase (presented in chapter 8) in addition to other word shape and Knowledge-Base (KB) features. Experiments conducted on different datasets show better combined extraction and disambiguation results compared with several baselines and competitors.

9.2 Introduction

NEE is a subtask of IE that aims to locate phrases (mentions) in the text that represent names of persons, organizations or locations regardless of their type. It differs from the term Named Entity Recognition (NER) which involves both

extraction and classification into set of predefined classes (persons, organizations or locations).

In addition to the NED challenges discussed in section 8.2, we add some challenges facing NEE from tweets:

- The informality nature of tweets makes the extraction process more difficult. For example, consider the tweet "*– **Lady Gaga** - **Speechless** live @ **Helsinki** 10/13/2010 http://www.youtube.com/watch?v=yREociHyijk ... @ladygaga also talks about her Grampa who died recently*", it is hard to extract the mentions using traditional NEE methods because of lack of formal statement. Traditional NEE methods might extract '*Grampa*' as a mention because of it capitalization. Furthermore, it is hard to extract the mention '*Speechless*', which is a name of a song, as it requires further knowledge about songs of '*Lady Gaga*'.

- The process of NEE involves degrees of uncertainty. For example, consider the tweet "*history should show that **bush jr** should be in jail or at least never should have been president*", it is uncertain whether the word '*jr*' should be part of the mention '*bush*' or not. Same for '*Office*' and '*Docs*' in the tweet "*RT @BBCClick: Joy! **MS Office** now syncs with **Google Docs** (well, in beta anyway). We are soon to be one big happy (cont) http://tl.gd/73t94u*" which some extractors may miss.

9.2.1 Hybrid Approach

According to a literature survey (see chapter 6), almost no research tackled the combined problem of NEE and NED. Researchers either focus on NEE or NED but not both. Systems that do NED like [62], either require manual annotations for NE or use some off the shelf extraction models like Stanford NER[1]. In this chapter, we present a combined approach for NEE that makes use of the NED approach presented in chapter 8.

Natural language processing (NLP) tasks are commonly composed of a set of chained sub tasks that form the processing pipeline. The residual error produced in these sub tasks propagates, affecting the final process results. In this thesis we focus on NEE and NED which are two common processes in many NLP applications. We claim that feedback derived from disambiguation would help in improving the extraction and hence the disambiguation. This is the same way we as humans understand text. The ca-

[1] `http://nlp.stanford.edu/software/CRF-NER.shtml`

pability to successfully understand language requires one to acquire a range of tools including syntax, semantics, and an extensive vocabulary. We try to mimic humans way of reasoning to solve the NEE and NED problems. Consider the tweet "– ***Lady Gaga*** - ***Speechless*** *live @* ***Helsinki*** *10/13/2010 http://www.youtube.com/watch?v=yREociHyijk ... @ladygaga also talks about her Grampa who died recently*". One would use his syntax knowledge to recognize '10/13/2010' as a date. Furthermore, his prior knowledge enables him to recognize '*Lady Gaga*' and '*Helsinki*' as a singer name and location name respectively or at least as names if he doesn't know exactly what they refer to. However, the term '*Speechless*' involves some ambiguity as it could be an adjective and also could be a name. A feedback clue from '*Lady Gaga*' would increase one's certainty that it refers to a song. Even without knowing that '*Speechless*' is a song of '*Lady Gaga*', there are sufficient clues to guess with quite high probability that it is a song. The pattern 'live @' in association with disambiguating '*Lady Gaga*' as a singer name and '*Helsinki*' as a location name, will lead to reason '*Speechless*' as a song.

Although the logical order for such system is to do extraction first then the disambiguation, we start with a phase of extraction which aims to achieve high recall (find as much NE candidates as possible). Then we apply disambiguation for all the extracted NE. Finally we filter those extracted NE candidates into true positives and false positives using features derived from the disambiguation phase in addition to other word shape and Knowledge-Base (KB) features. Figure 9.1 illustrates our approach. The potential of this order is that the disambiguation step gives extra information about each NE candidate that may help in the decision whether or not this candidate is a true NE.

For NEE, we believe that this process implies high degree of uncertainty. Our approach is based on finding as much NE candidates (mentions) as possible (achieving high recall) and then filter those candidates. To achieve this high recall, we use a tweet segmentation method [75], in addition to KB lookup. Furthermore, we use a CRF model to generate top k possible annotations for each tweet. We use all those annotations as candidates of NEs. To improve the precision we apply an SVM model to predict if the candidate is a NE or not according to set of features. We use word shape features (like capitalization), Part Of Speech (POS) tags, KB features (like number of possible entities for the given extracted mention), and features derived from disambiguation process (like similarity between the mention context and the disambiguated entity page).

We also consider the best annotation set for the tweet given by the CRF model as true positives. Results obtained from both SVM and CRF are union-

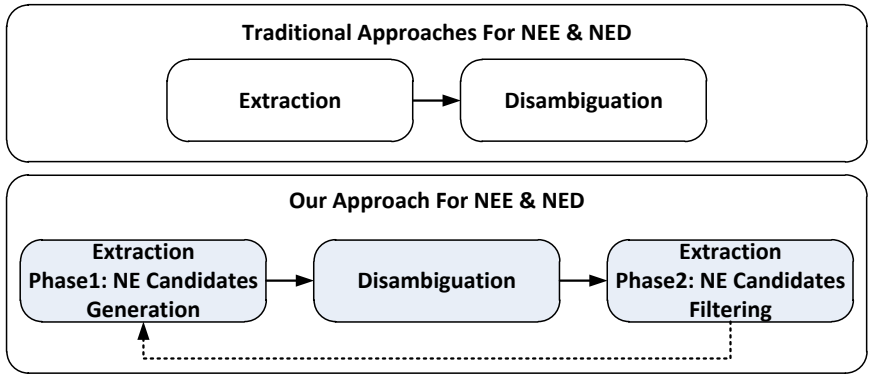


Figure 9.1: Traditional approaches versus our approach for NEE and NED.

ized to give the final extraction results. The idea behind this unionization is that SVM and CRF work in a different way. The former is a distance based classifier that uses numeric features for classification which CRF cannot handle, while the latter is a probabilistic model that can naturally consider state-to-state dependencies and feature-to-state dependencies. On the other hand, SVM does not consider such dependencies. The hybrid approach of both makes use of the strength of each.

The rest of the chapter is organized as follows. Section 9.3 presents our approach for NEE in tweets while the experimental results are presented in section 9.4. Finally, conclusions and future research directions are presented in section 9.5.

9.3 Named Entity Extraction

As stated before, we do the extraction on two phases, one before the disambiguation and one after. The first phase aims to generate as much NE candidates as possible to achieve a high recall. Then the second phase of filtering those candidates comes after the disambiguation process. The whole extraction process is described in figure 9.2.

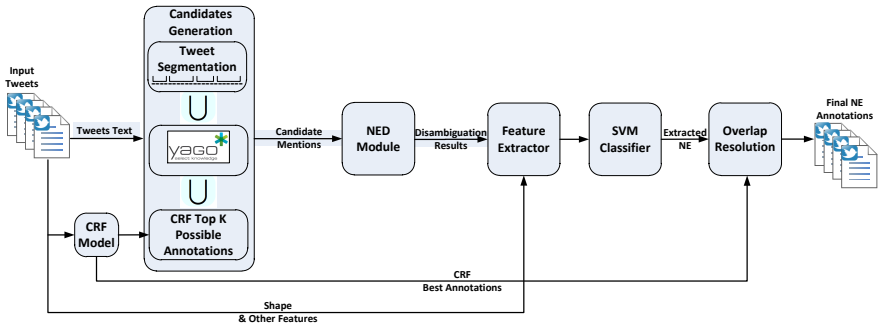


Figure 9.2: Extraction system architecture.

9.3.1 Candidates Generation

For this task we use the following candidates generation methods:

- **Tweet Segmentation**: Tweet text is segmented using the segmentation algorithm described in [75]. Each segment is considered a candidate for a named entity. The segmentation approach splits the tweet text based on a stickiness function. More formally, given a tweet of four words $w_1w_2w_3w_4$, we segment it as $w_1w_2||w_3w_4$ rather than $w_1||w_2w_3w_4$, if $C(w_1w_2) + C(w_3w_4) > C(w_1) + C(w_2w_3w_4)$, where C(.) basically captures the probability being a valid phrase of a segment. Microsoft Web N-Gram [90] and Yago KB (which represents Wikipedia normalization of the stickiness function) are used as described in [75].

- **KB Look-up**: The list look-up strategy is an old method of performing NEE by scanning all possible n-grams of a document content against the mentions-entities table of a KB like YAGO or DBpedia [81]. Due to the short length of the messages and the informal nature of the used language, KB look-up is a suitable method for short context NEE.

- **CRF Alternative Annotations**: CRF is a probabilistic model that is widely used for NER [22]. Despite the successes of CRF, the standard training of CRF can be very expensive due to the global normalization [91]. In our approach, we used an alternative method called *empirical training* [92] to train a CRF model. The maximum likelihood estimation (MLE) of the empirical training has a closed form solution, and it does

not need iterative optimization and global normalization. So empirical training can be radically faster than the standard training. Furthermore, the MLE of the empirical training is also a MLE of the standard training. Hence it can obtain competitive precision to the standard training. Tweet text is tokenized using special tweets tokenizer [85]. For each token, the following features are extracted and used to train the CRF:

- The Part of Speech (POS) tag of the word provided by a special POS tagger designed for tweets [85].
- If the word initial character is capitalized or not.
- If the word characters are all capitalized or not.
- If the word has any capital letters or not.
- If the word characters are all capitalized or not.

We used the IOB representation for the training annotations. Tokens that represents beginning of a NE are annotated as $B-NE$. Tokens that represents inner part of a NE are annotated as $I-NE$. While token that don't belong to a NE are annotated as O. The CRF model is trained and used to provide not only the best annotation set for the tweet text but also top K possible annotation sets.

To obtain the K most probable paths for $p(S|O)$, where O is the observation sequence (tweet tokens) and S is the tag sequence (annotations), a naive implementation is to calculate probabilities for all possible S, sort them by probability and select the K most probable S. But such a naive implementation is space and computation inefficient. To illustrate this, let $O = [o_1, o_2, ..., o_n]$ and $S = [s_1, s_2, ..., s_n]$. Suppose the size of the tag space is $|s|$, then there can be as many as $|s|^n$ possible paths. To exhaust the complete path space becomes impractical with the growth of $|s|$ and n.

In our work, we constrain the *complete* path space to a *promising* path space with the unary and pairwise constraints. According to the Co-occurrence Rate Factorization, $p(S|O)$ can be factorized as follows:

$$p(S|O) = \prod_{i=1}^{n} p(s_i|o_i) \prod_{j=1}^{n-1} \texttt{CR}(s_{j-1}; s_j|o_{j-1}, o_j).$$

We impose the unary constraints to the unary factors $p(s_i|o_i)$. That is for an observation o_i, we only consider the top-K most possible s_i. Similarly,

for the pairwise factors $\texttt{CR}(s_{j-1}; s_j | o_{j-1}, o_j)$, we only consider the top-K most possible $(s_{j-1}; s_j)$ for (o_{j-1}, o_j). So at most there can be $K^{\frac{n}{2}}$ paths in the promising path space. In practice, this works well. But to reduce the complete path space to a promising path space may also lead to excluding the best path from the promising path space even this is very rare on real world datasets. We remedy this by adding the best path to the top-K paths absolutely.

This method of candidates generation enables to handle uncertainties in NE representations. In this way we will be able to extract both '*MS Office*' and '*MS*' as possible candidates. We leave the decision of deciding which one is the correct representation for the next step which makes use of disambiguation and KB clues. As a post processing for the candidates generation step, we remove duplicate candidates. Furthermore, to improve the precision, we applied filtering hypotheses (such as removing segments that are composed of stop words or having verb POS).

9.3.2 Candidates Filtering

After generating the candidates list of NE, we apply our disambiguation approach as described in section 8.3 to disambiguate each extracted NE candidate. After the disambiguation phase, we use SVM classifier with RBF kernel to predict which candidates are a true positive and which ones are not. SVM is a machine learning approach used for classification and regression problems that uses distance-based similarity measures. We use the following set of features for each NE candidate to train the SVM:

- **Shape Features**: The features used to train the CRF model listed in section 9.3.1.
- **Probabilistic Features**:
  - The joint and the conditional probability of the candidate obtained from Microsoft Web N-Gram services.
  - The stickiness of the segment as described in [75].
  - The segment frequency over around 5 million tweets [2].
  - The extraction confidence for the candidate if it was extracted by the CRF.

[2] `http://wis.ewi.tudelft.nl/umap2011/` + TREC 2011 Microblog track collection.

- **KB Features**:
  - If the segment appears in WordNet.
  - If the segment appears as a mention in Yago KB.
- **Disambiguation Features**:
  - All the features described in section 8.3.2 for the top ranked entity page selected for the given NE candidate.
  - If any of the candidate entity pages for the given NE candidate was a Twitter or Facebook or Linkedin or ebay or IMDB page.

9.3.3 Final Set Generation

For this task, we take the union of the best CRF annotation set and SVM results, after removing duplicate extractions, to get the final set of annotations. For overlapped annotations, we select the entity that appears in Yago, then the one having longer length.

9.4 Experimental Results

In this section, we present the results of experiments with the presented methods of extraction applied on four different collections of tweets. The goal of the experiments is to investigate effectiveness of our approach in comparison with a state-of-the-art extraction approach (Stanford NER [53]) and a competitor (Ritter approach [73]). Furthermore, we present a combined evaluation for our both extraction and disambiguation approaches in comparison with a competitor (AIDA [62]) applied on the two datasets described in section 8.4.1.

9.4.1 Datasets

In addition to the two datasets described in section 8.4.1, we use two other Twitter datasets named 'Ritter' and '#MSM' collections. The 'Ritter' collection[3] is a collection presented by [73] and composed of 2394 tweet with 1495 annotated NE mention while the '#MSM' collection[4] is composed of 2815 tweet with 2987 NE mention annotations.

[3] https://github.com/aritter/twitter_nlp
[4] http://oak.dcs.shef.ac.uk/msm2013/

9.4.2 Extraction Evaluation

In this experiment we evaluate a set of extraction techniques on our datasets:

- **Stanford**: Stanford NER [53] model trained on normal CoNLL collection. It is based on CRF model which incorporates long-distance information. It achieves good performance consistently across different domains.
- **Stanford_Caseless**: Stanford NER caseless model (NER models that ignore capitalization).
- **Stanford_CRF**: Stanford CRF model trained and tested on our ground truth collection using 4-fold cross validation.
- **Ritter_NER**: a system that uses a set features including orthographic and dictionary features [73].
- **TwitterNEED**: Represents the different phases of the extraction process; the candidates generation phase **(CG)**, the candidates filtering phase **(SVM_CF)**, the best CRF annotation set **(Best_CRF)** and the final NE set generation **(SVM∪CRF)**.

SVM is trained and tested using 4-fold cross validation. The training folds are used to train the NED and the NEE models while the test fold is used for validation. As Ritter and #MSM collections don't have a ground truth for NED to be used for training, we use instead a disambiguation model trained on the other two collections (Mena and Brian).

Evaluation Criteria: As mentioned before, the process of NEE involves uncertainty. For example, the tweet "*RT @BBCClick: Joy! MS Office now syncs with Google Docs (well, in beta anyway). We are soon to be one big happy (cont) http://tl.gd/73t94u*", annotators may have contrary decisions whether '*Office*' and '*Docs*' are part of the mentions '*MS*' and '*Google*' or not.

This is why we preferred to use the extraction evaluation strategy introduced by GATE[5] which computes three measures for each of the precision, recall, and F1 named strict, lenient, and average. In strict only perfectly matching annotations are counted as correct. While in lenient partially matching annotations are counted as correct. In average, strict and lenient scores are averaged (this is the same as counting a half weight for every partially correct annotation).

[5] `http://gate.ac.uk/`

Table 9.1: Evaluation of NEE approaches

(a) Mena Collection

	Lenient			Average			Strict		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
Stanford	0.8941	0.9275	0.9105	0.8235	0.8461	0.8346	0.7514	0.7647	0.7580
Stanford_Caseless	0.7818	0.8078	0.7946	0.7248	0.7461	0.7353	0.6673	0.6843	0.6757
Stanford_CRF	0.8859	0.8863	0.8853	0.8084	0.8084	0.8078	0.7309	0.7306	0.7303
Ritter_NER	0.9066	0.7055	0.7935	**0.8491**	0.6511	0.7370	0.7899	0.5966	0.6797
TwitterNEED:									
CG	0.4683	**0.9980**	0.6374	0.3815	**0.9676**	0.5473	0.3187	**0.9373**	0.4756
SVM_CF	**0.9330**	0.7647	0.8405	0.8659	0.7343	0.7947	**0.8031**	0.7039	0.7503
Best_CRF	0.9279	0.9078	0.9177	0.8333	0.8137	0.8234	0.7384	0.7196	0.7289
SVM∪CRF	0.8994	0.9471	**0.9226**	0.8344	0.8794	**0.8563**	0.7695	0.8118	**0.7901**

(b) Brian Collection

	Lenient			Average			Strict		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
Stanford	0.7676	0.6877	0.7255	0.6799	0.6038	0.6396	0.5907	0.5199	0.5530
Stanford_Caseless	0.6895	0.6151	0.6502	0.6248	0.5558	0.5883	0.5597	0.4965	0.5262
Stanford_CRF	0.8447	0.7953	0.8190	0.7972	0.7427	0.7688	0.7487	0.6901	0.7180
Ritter_NER	0.6713	0.6005	0.6339	0.6145	0.5414	0.5756	0.5559	0.4824	0.5165
TwitterNEED:									
CG	0.1746	**0.9905**	0.2969	0.1627	**0.9609**	0.2783	0.1517	**0.9312**	0.2609
SVM_CF	**0.9033**	0.7016	0.7898	0.8721	0.6864	0.7682	**0.8418**	0.6713	0.7469
Best_CRF	0.8783	0.8379	0.8576	**0.8308**	0.7855	0.8075	0.7825	0.7331	0.7570
SVM∪CRF	0.8425	0.8738	**0.8579**	0.8056	0.8353	**0.8202**	0.7687	0.7968	**0.7825**

(c) Ritter Collection

	Lenient			Average			Strict		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
Stanford	0.6442	0.6903	0.6665	0.5684	0.6075	0.5873	0.4922	0.5247	0.5079
Stanford_Caseless	0.4381	0.5621	0.4924	0.3807	0.4877	0.4276	0.3231	0.4132	0.3626
Stanford_CRF	0.7600	0.6015	0.6704	0.7017	0.5528	0.6174	0.6429	0.5041	0.5642
Ritter	-	-	-	-	-	-	**0.7300**	0.6100	**0.6700**
TwitterNEED:									
CG	0.1042	**0.9860**	0.1884	0.0946	**0.9326**	0.1718	0.0858	**0.8792**	0.1563
SVM_CF	**0.8189**	0.4920	0.6147	**0.7738**	0.4693	0.5843	0.7296	0.4466	0.5540
Best_CRF	0.7722	0.6742	0.7199	0.7057	0.6148	0.6572	0.6390	0.5554	0.5943
SVM∪CRF	0.7396	0.7336	**0.7366**	0.6843	0.6792	**0.6817**	0.6290	0.6248	0.6269

(d) #MSM Collection

	Lenient			Average			Strict		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
Stanford	0.7341	0.8305	0.7793	0.6728	0.7589	0.7133	0.6112	0.6872	0.6470
Stanford_Caseless	0.7281	0.7788	0.7526	0.6679	0.7122	0.6893	0.6073	0.6456	0.6259
Stanford_CRF	0.8745	0.7615	0.8141	0.8171	0.7090	0.7592	0.7592	0.6565	0.7041
Ritter	0.7195	0.6302	0.6719	0.6982	0.6103	0.6513	0.6767	0.5904	0.6306
TwitterNEED:									
CG	0.2162	**0.9969**	0.3553	0.1936	**0.9652**	0.3225	0.1741	**0.9336**	0.2935
SVM_CF	**0.8840**	0.7358	0.8031	**0.8428**	0.7123	0.7721	**0.8028**	0.6888	0.7414
Best_CRF	0.8252	0.8803	**0.8519**	0.7722	0.8234	0.7970	0.7192	0.7665	0.7421
SVM∪CRF	0.8013	0.9088	0.8517	0.7588	0.8613	**0.8068**	0.7164	0.8139	**0.7620**

It is also important to note that we evaluate only the ability of the different systems to extract mentions of named entities rather extraction and classification (into person, organization, location, etc.). We believe that NED phase could go further by linking the mentions to their correct entities instead of just doing a classification for their entity type.

Discussion: Table 9.1 shows the performance of our TwitterNEED approach in comparison with the other competitors (Stanford and Ritter). From the results shown we can observe the following:

- The evaluation of Ritter approach application on Ritter collection, only shows the strict precision, recall, and F1. This is because [73] used only strict strategy for extraction evaluation and those values listed are copied from Ritter's evaluation. The model provided by Ritter [6] has already been trained on the whole Ritter collection. So that applying the provided model on the collection is not fair.

- TwitterNEED outperforms all the Stanford models even that one trained on our collections in terms of F1 performance.

- TwitterNEED outperforms Ritter_NER on three collections out of four collections. It only fails to outperform Ritter_NER on Ritter's collection. The reason behind this is that Ritter_NER is a pipeline of classifiers (POS, Capitalization reliability, Chunker) trained and used on the same collection. In contrast, our NED model trained on Mena and Brian collections is used on #MSM collection and contributes in improving the extraction performance. This proves that our models are generic and not restricted to the collection used for training.

- We can see that the effect of unionization of the SVM and the CRF output is more clear on the strict results than the lenient results where the improvement over the CRF results is low. The reason is that the SVM is more able to find the exact annotation than the CRF which sometimes misses part of the NE. For example, the tweet "***BBC**: **US** poet wins **Dylan Thomas prize** http://is.gd/ibWvK*", the CRF extracts the mentions '*BBC*', '*US*' and '*Dylan Thomas*'. On the other hand, the SVM is able to correctly classify '*BBC*' , '*US*' and '*Dylan Thomas prize*' as true positive NE. Those NEs that are correctly classified by SVM match the exact manual annotations. This is due to the effective segmentation approach used in the phase of candidates generation plus the usage of the disambiguation features driven

[6] https://github.com/aritter/twitter_nlp

Table 9.2: Combined evaluation of NEE and NED approaches

	Mena Collection			Brian Collection		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
Stanford + AIDA	**0.7263**	0.5569	0.6304	0.5005	0.2940	0.3704
TwitterNEED	0.6861	**0.7157**	**0.7006**	**0.5455**	**0.5640**	**0.5546**

from the disambiguation process in classifying the segments into true positives NE and false positives ones. The disambiguation module is able to find the correct page for the mention '*Dylan Thomas prize*' which is '`http://en.wikipedia.org/wiki/Dylan_Thomas_Prize`'.

Furthermore, the SVM is able to extract NEs that is missed completely by the CRF. For example, the tweet "*@ABC U destroyed #DWTS for this—->**Palin** Rips **American Idol**: **AP** got an advance copy of **Sarah Palin**'s new book... http://bit.ly/aGnwmh*", the CRF extracts only '*Sarah Palin*' while the SVM extracts '*Sarah Palin*', *American Idol*, and *AP*.

Another example that shows the power of using the disambiguation results and features to improve the extraction is the tweet "*RT @nytonline : **Pamela Anderson** to join **Indian BB** : Former **Baywatch** star **Pamela Anderson** is join the Indian version of **Big Brother**. http://ow.ly/19YFz1*". While CRF extracts only the two mentions of '*Pamela Anderson*', the SVM was able to extract '*Baywatch*' and '*Big Brother*' in addition to the two mentions of '*Pamela Anderson*'. The NED model links the mention '*Big Brother*' to the entity home page '`http://en.wikipedia.org/wiki/Big_Brother_(UK)`' which is the correct entity page for this mention. The similarity between the tweet context and the entity page leads to correctly classifying the segment '*Big Brother*' as a true positive NE. Similarly, '*Baywatch*' is linked to the page '`https://en.wikipedia.org/wiki/Baywatch`' and correctly extracted by the SVM.

9.4.3 Combined Extraction and Disambiguation Evaluation

In this experiment we compare the performance of our TwitterNEED system against AIDA[7]. AIDA mainly is a disambiguation system, however it uses Stanford_NER for automatic NE extraction. We consider the combination of

[7] `https://d5gate.ag5.mpi-sb.mpg.de/webaida/`

Stanford_NER and AIDA disambiguation system as a competitor to our extraction and disambiguation system. For a combined extraction and disambiguation evaluation, we consider the true positives set to include each correct exact mention extraction that is correctly assigned to its entity home page. While false positives set includes: *a*) mentions that are partially extracted; and *b*) extracted mentions that are not part of correct NE at all; and *c*) extracted mentions that match exactly correct NE but not successfully assigned to its entity home page. Finally, false negatives set includes all correct NEs that are completely missed by the extractor. Results in table 9.2 shows the superiority of TwitterNEED over the combined Stanford and AIDA system. Evaluation is done only on the collections which have a ground truth for the entity home pages. Recall would be higher if we considered the top k disambiguated home pages instead of the top one as we do here.

9.5 Conclusions and Future Directions

In this chapter, we present TwitterNEED, an approach for the NEE and NED in tweets. We propose a hybrid approach for NEE in tweets that is based on SVM and CRF. The system is composed of three phases. The first phase aims to generate NE candidates. The second is a generic approach for NED in tweets for any named entity (not entity oriented). Finally the third phase is to filter the NE candidates using features derived from disambiguation and other shape and KB features.

In the future work, we want to investigate our approaches for NEE and NED in constructing KB's for closed domains from social media networks. For example, building a KB for general parliament elections or for some local festivals based on users generated contents on social media networks. Furthermore, we want to include entity-entity similarity features to the disambiguation process. This will require to iteratively repeat the disambiguation and extraction processes as suggested in chapter 4 because of the bad effect of the large number of the false positives extractions on the disambiguation results in the first iteration.

Part IV

Conclusions

CHAPTER 10

Conclusions and Future Work

10.1 Summary

A main challenge of natural language is its ambiguity and vagueness. To automatically resolve ambiguity by computers, typically the grammatical structure of sentences is used, for instance, which groups of words go together (phrases) and which words are the subject or object of a verb. However, when we move to informal language widely used in social medias, the language becomes more ambiguous and thus more challenging for automatic understanding. Social media content represents a big part of all textual content appearing on the Internet. These streams of user generated content (UGC) provide an opportunity and challenge for media analysts to analyze huge amount of new data and use them to infer and reason with new information. An example of a main sector for social media analysis is the area of customer feedback through social media. With so many feedback channels, organizations can mix and match them to best suit corporate needs and customer preferences. Another beneficial sector is social security. Automatic monitoring and gathering of information posted in social media could be helpful to take actions to prevent violent, and destructive behaviors.

Information Extraction (IE) is the research field that enables the use of such a vast amount of unstructured distributed data in a structured way. Named Entity Extraction (NEE) is a sub task of IE that aims to locate phrases (mentions) in the text that represent names of persons, organizations or locations regardless of their type. While Named Entity Disambiguation (NED) is the task of exploring which correct person, place, event, etc. is referred to by a mention.

The main goal of this thesis is to obtain more insight into how computers can truly understand natural languages by mimicking human ways of language understanding especially for domains that lack formal sentence struc-

ture. The proposed methods open the doors for more sophisticated applications based on users' contributions on social media. We propose a robust combined framework for NEE and NED in semi and informal text. This framework applies a reinforcement approach which makes use of disambiguation results feedback to improve extraction quality. The achieved robustness of NE extraction obtained from this principle has been proven for several aspects: (a) it is independent on the used combination of the extraction and the disambiguation techniques; (b) once a system is developed, it can trivially be extended to other languages; all that is needed is a suitably amount of training data for the new language; (c) it works in a domain-independent manner. It generalizes to any dataset; (d) it is shown to be robust against a shortage of labeled training data, the coverage of KBs, and the informality of the used language. Furthermore, we propose a method of handling the uncertainty involved in extraction to improve the disambiguation results. Finally, we propose a generic approach for NED in tweets for any named entity (not entity oriented). This approach overcomes the problem of limited coverage of KBs. Mentions are disambiguated by assigning them to either a Wikipedia article or a home page. We also introduce a method to enrich the limited entity context.

10.2 Research Questions Revisited

Here, we revisit the research questions introduced in chapter 1. We show the findings and conclusions we came with.

- **How do the imperfection and the uncertainty involved in the extraction process affect the effectiveness of the disambiguation process and how can the extraction confidence probabilities be used to improve the effectiveness of disambiguation?**

  To answer this question, we conducted experiments on a set of holiday properties description with the purpose to extract and disambiguate toponyms in the description text. For extraction, we used different extraction models (rule-based and statistical). For disambiguation, we applied a clustering based disambiguation algorithm with the aim to find the correct reference for a toponym. By comparing the disambiguation results applied on manually annotated toponyms against the results applied on automatically extracted toponyms, we found that the disambiguation results of the manually annotated toponyms are better than those of the

automatically extracted toponyms. The reason behind this is that the disambiguation algorithm (like many state-of-the-art approaches) relies on coherency features (in our case we used distances between toponyms references). False positive and false negative toponyms harm the coherency features and hence mislead the disambiguation. We answered this part of the question in chapters 3 and 4. Furthermore, we came to the same conclusion in chapter 7 where we used entity-relationships in a KB graph to disambiguate tweet mentions extracted by a look-up strategy.

To investigate whether the extraction confidence can be used to improve the effectiveness of disambiguation, we used statistical extraction models because of their ability to measure the extraction confidence probabilities. As mentioned above, coherency features are used to disambiguate extracted toponyms. Instead of trusting the outcomes of the extraction models, we modified our disambiguation algorithm so that not every extracted toponym contributes equally to the disambiguation of the property country but in ratio of its extraction confidence. This method improves the overall disambiguation results. We investigated this part of the question in chapter 4.

- **How can the disambiguation results be used to improve the certainty of extraction and what are the evidences and features that could be derived from disambiguation to improve extraction process?**

  To answer this question, we used clues and features derived from disambiguation results in different ways. In chapter 3, we considered the toponym that is disambiguated and linked to different references across the documents collection, to be highly ambiguous and harmful for disambiguating other toponyms. Removal of such highly ambiguous toponyms improves the extraction results and hence the disambiguation results. Similarly, in chapter 4 we introduced those highly ambiguous toponyms in the same way. However, instead of removing them, we introduced them as negative samples to retrain the statistical extraction models. As a result, certainty of true positive toponyms increased and certainty of false positives decreased. This method enables iterative improvement of extraction and disambiguation processes.

  In chapter 5, we tried a different approach. We first applied a statistical extraction model and used a low cutting threshold with the aim of achieving high recall by considering possible alternatives and candidates for toponyms. SVM classifier is then used to determine which candidates

are true positive toponyms and which candidates are false positives. For this classification task we used informativeness features and coherency features derived from the disambiguation results. Usage of informativeness features leads to detect common and highly frequent words that are falsely extracted as toponyms. While coherency features is better in finding those falsely extracted toponyms that appears infrequently across the documents collection as they showed no coherency with the other extracted toponyms.

On the domain of short messages of tweets, in chapter 7, we used a simple KB look-up strategy for mentions extraction to achieve high recall. A cluster-based disambiguation algorithm has been developed to find coherent entities among the possible candidates. Two entities are considered coherent if there is a direct link joining them in a KB graph. From the disambiguation results, we find the isolated entities which are not coherent with any other candidates. We consider the mentions of those isolated entities as false positives and therewith improve the precision of extraction. In chapter 9, we used same strategy of achieving a high recall in the initial extraction step. Afterwards, the disambiguation approach introduced in chapter 8 is applied on all candidates. Features used for disambiguation (like tweet-entity page similarity) along with other KB and shape features were used to find false positive mentions. These features leads to correctly classify the ambiguous and informally represented entities as true positive.

In conclusion, features that is used for disambiguating named entities can be used also for enhancing the extraction process.

- **How robust is the *reinforcement effect* and whether this concept is valid across domain, approaches, and languages?**

  The answer of this question is spread across the thesis chapters. The thesis discusses the robustness of the *reinforcement effect* across domains in parts II and III. Part II, shows its application on the domain of semi-formal text with the aim to extract and disambiguate toponyms. While part III illustrates the *reinforcement effect* on different types of named entities on the domain of the informal text of tweets. On both domains, the *reinforcement effect* is experimentally proved to be valid. Robustness across different approaches is also shown in different places in the thesis. For extraction, we tried rule-based approach (chapter 3), statistical approaches (chapters 4 and 5), look-up approach (chapter 7) and a hy-

brid approach (chapter 9). For disambiguation, we used unsupervised approaches (part II, and chapter 7) and supervised approach (chapters 7 and 9). In all combinations used, we were able to prove the validity of the *reinforcement effect*. No matter what approaches are used, the *reinforcement effect* is still applicable. Finally, robustness across different languages, against variable system parameters, and against limited training sets is shown in chapter 5. Results showed how our proposed method is highly competitive to the state-of-the-art language dependent approaches and at the same time less sensitive to changing circumstances.

- **How can we overcome the limited coverage of knowledge-bases and how can the limited context of short messages be enriched?**

  We investigated the answer of this question in chapter 8. We introduced a generic open world approach for NED in tweets that make use of a KB as well as Google search engine to find candidate set of pages of entities for each mention. We distinguished between two types of entity pages. Wiki page which represents a Wikipedia page for the entity, and Non-Wiki page which represents a home page or any other representative page for the entity. Two sets of features (context and URL) are presented for better finding of Wiki and Non-Wiki entity page. We also introduced a method to enrich a mention's context by adding top frequent terms from targeted tweets (tweets discussing same topic) to the context of the mention. Results show that context features are more powerful in finding Wiki pages of entities, while URL features are more helpful in finding pages of entities containing less or almost no contents (Non-Wiki pages). The proposed context enrichment method improves the disambiguation results of Non-Wiki entities.

10.3 Future Work

In this thesis, we introduced research on named entities extraction and disambiguation in semi and informal text. For the disambiguation process, we make use of the available knowledge bases. However, knowledge bases have limited coverage and are not always up to date. It will be useful if we make use of our presented approaches to construct KBs or enrich existing ones by analyzing the continuously flowing information of social media.

To construct a KB, we need to go further than extracting and disambiguating named entities to relation extraction. The aim of relation extraction is to

detect and characterize the semantic relations between entities in text. We believe that feedback loops can take place between the relation extraction process and the disambiguation process. Handling uncertainties will be valuable for the task of KB population specially with the increasing chance of having untrustworthy sources of information and the expected erroneous extraction modules.

We want to apply this future work on the domain of social security. We want to extend our contributions on the TEC4SE project[1] by enriching social media posts with more semantics and relationships between entities involved in different events. The will give the decision makers in the security agencies more evidences about possible threats.

[1] http://www.tec4se.nl/

Appendices

APPENDIX A

Neogeography: The Treasure of User Volunteered Text

A.1 Summary

In this appendix, we propose a motivating application for our contributions within this thesis. Our wide objective is to propose a new portable, domain-independent XML-based technology that involves sets of free services that: enable end-users communities to express and share their spatial knowledge; extract specific spatial information; build a database from all the users' contributions; and make use of this collective knowledge to answer users' questions with sufficient level of quality.

The contents of this appendix have been published as [46].

A.2 Introduction

Users are not passive recipients. Not only can they choose the type of information they want to access but also they can even produce the information themselves. The term *Neogeography*, sometimes referred to as *volunteered geographic information (VGI)*, is a special case of the more general web phenomenon of *user-generated content (UGC)*, that has a relation to geographical features of the earth [93]. UGC refers to various kinds of media content, publicly available, that are produced by end-users. Such contents may include digital video, blogging, mobile phone photography, and wikis. UGC can provide citizens, consumers and students with information and knowledge as its contents tend to be collaborative and encourage sharing and joint production of information, ideas, opinions and knowledge among users.

In neogeography, end-users are not only beneficiaries but also contributors of geographic information. Neogeography combines the complex techniques of cartography and GIS and places them within reach of users and developers [94]. In this proposal, our wide objective is to propose a new portable, domain-independent XML-based technology that involves sets of free services that: enable end-users communities to express and share their spatial knowledge; extract specific spatial information; build a database from all the users' contributions; and make use of this collective knowledge to answer users' questions with sufficient level of quality. Users can use free text (such as SMS, blogs, and Wikis) to express both their knowledge and enquiries, or they can use map-assisted questions from their smart phones. Users can benefit from this technology by using a question answering system to retrieve specific information about some place in the form of generated natural language and, if the communication device allows it, simple maps.

A.3 Motivation

The rapid growth in the IT in the last two decades leads to the growth in the amount of information available on the World Wide Web. However, the information accessibility in the developing countries is still growing slowly. In Africa, which has a population of more than one billion, only 15% of the populations has access to the internet [95]. On the other hand, figures released in 2007 reported that Africa is the fastest-growing cell phone market in the world, having increased at a rate of 20 per cent per year since 2007, with a total market of nearly 649 million users in 2011 [96]. The wide spreading of mobile phones coincides with developing applications and services based on wireless telecommunication. SMS text messaging can be an efficient and effective means of information sharing and accessing. The number of SMS sent globally tripled between 2007 and 2010, from an estimated 1.8 trillion to a staggering 6.1 trillion [97].

The proposed system gives communities of workers in developing countries, where governments are hardly covering the basic public services, the ability to help themselves, sharing their information through mobile phones. For example, truck drivers may provide the system with SMS messages about the traffic situation at particular place at a specific time. Structured information about the place, the time and the situation is extracted from these messages, and stored in a spatial DB. Users can benefit from this system by asking about the best way to go to somewhere by sending a SMS question.

Another possible application is on tourism domain. Tourists are naturally motivated to share their experiences via forums, blogs or even Twitter messages. The system can extract useful information from these tweets and represent it in a structured way. This information can be the opinions of users about hotel services. The system should extract information like the hotel name, its location, and the user opinion about it.

After the extraction process, the extracted information should be integrated into a probabilistic DB using a probabilistic framework to deal with the uncertainty that comes with the users' contributions. Contradiction and subjective uncertainty are expected, which requires that the entire process supports handling of probabilistic data.

The users can benefit from this data by submitting queries like "*What are the good hotels within Paris?*" using question answering mechanism. The system then will use the extracted information with the help of existing open linked data to answer those questions.

A.4 Related Work

Within the previously discussed theme many projects have been developed to make use of contributions of users. Wikimapia and OpenStreetMap are good examples of collaborative projects to create a free editable map of the world, while Google Earth and Flicker allow users to upload and place their own captured photos over the earth's map. Other tools like MapQuest and OpenAPI allow users to embed directions to some places in their web site. Users can share their directions, recorded by their GPS devices, using websites like GPSVisualizer and GeoTracing. Another application is *Digital Foot-printing* for tourists using the presence and movements from cell phone network data and the geo-referenced photos they generate [98]. Similarly, TwitterHitter plots the tweets of single Twitter individual or group of individuals and generates an extended network graph view for visualizing connections among individuals in a region [99]. To bring this technology to the developing world, we need however to adapt it to the available communication technology, namely SMS on simple mobile phones.

Other research dealt with text as a source of geographic information. Numerous researches have focused on geo-parsing which tries to resolve geographic names appear in text [100, 101]. *Places mentioned in this book* service provided by Google Books is one of those applications based on such researches. Other researches have tackled the area of analyzing and visualizing the fre-

quencies of terms used in referring to geographical description [102, 103]. Few researches try to model human natural language expression in representation of references to places [104, 105]. Spatio-Temporal Information Extraction is mentioned by some researches for geographic information retrieval purposes [106, 107]. The aim of those researches was to annotate documents with sets of locations and time information extracted from those documents, visualize this extracted information on digital map.

Other research groups worked on geographical ontologies. Within this paradigm, [108] focused on the problem of integrating multiple datasets for constructing geo-ontology for the purpose of developing a spatially-aware search engine, while [109] tried to propose a reference model for developing geographic ontologies. A GeoOntology building algorithm was developed by [110] to extract data from the different data sources (relational databases, XML documents, GML documents, etc.) and transform them into ontology instances. Similarly, [111] describes work done in order to integrate the information extracted from gazetteers, WordNet, and Wikipedia.

A.5 Challenges

This proposal comes at the cross roads of several research areas. These research areas include: information extraction, the semantic web, probabilistic data integration, probabilistic XML databases, and spatial databases. Information Extraction (IE) plays a major role in this proposal. IE systems analyse human language text in order to extract information about pre-specified types of events, entities or relationships. In our case, the users' community keeps providing their knowledge about conditions within particular geographic regions in a dynamic, free-text manner and our task is to extract valuable information from this mass of text and use it to populate a pre-specified templates. This requires the extraction of the W4 questions of: who, where, when and what from textual descriptions. Information Extraction from text sources is, by nature, challenging in many ways:

- Information contained in text is often partial, subject to evolution over time, in conflict with other sources, and sometimes untrustworthy.

- Recognizing the co-reference of entities and events when they are described or referred to in different textual sources.

- The lack of clear guidelines for evaluating the correctness of the output generated by an extraction algorithm.
- Information about spatial data adds another challenge of resolving the spatial vagueness. Some places have the same names and sometimes the spatial information is not well defined, or changes from time to time.
- Different textual sources imply different ways of writing, and expression.
- IE systems are always built for a specific domain. Research is required on the automatic acquisition of template filler patterns, which will enable systems for much larger domains.

Uncertainty in data is another challenge point. Uncertainty may come in different ways:

- Uncertainty in the extraction process, i.e. the precision level expected from the IE system in resolving facts or geographical names.
- Uncertainty in the source of information, i.e. the possibility that the data provided is completely or partially incorrect.
- The contradictions between the extracted information and the information previously extracted and stored in the probabilistic database.
- The validation of the information over time. Geographical information is dynamic information and always changing over time.

Semantic web and linked data must have precedence when we are dealing with global neogeographic systems. The semantic web adds another challenge of linking the rapidly growing number of existing web data sources to find the meaningful content [112]. There is a growing interest in designing probabilistic XML-databases to represent uncertain data [113]. Besides, spatial databases support spatial data types in its implementation, providing spatial indexing and spatial join methods. In this proposal, we suggest to make use of both mentioned types of databases by extending the probabilistic XML-databases with capabilities to represent spatial information. Solving these problems calls for ideas from multiple disciplines, such as machine learning, natural language understanding, machine translation, probabilistic data integration, knowledge representation, data management, and linguistic theory related to language semantics.

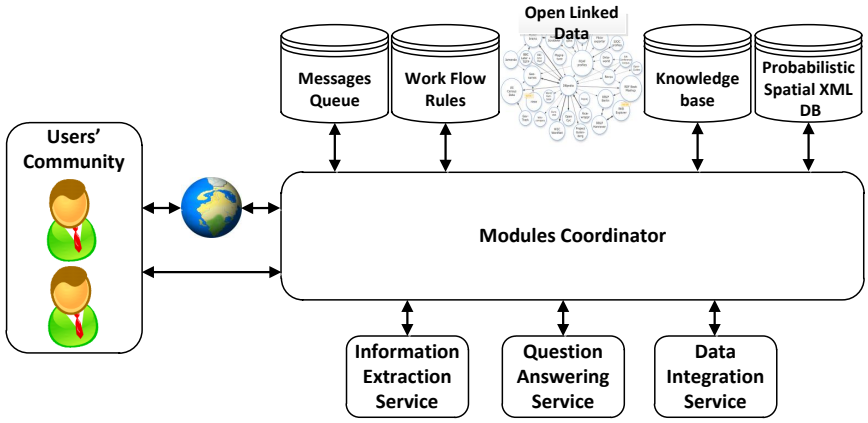


Figure A.1: The proposed system architecture.

A.6 Proposed System Architecture

Figure A.1 shows the proposed system architecture which is composed of the following components and modules:

1. **Modules Coordinator (MC):** This module is the controller of the whole system. It is responsible for controlling the work and data flow between different services. It receives the user contributions and requests, and sends activation messages to the intended services according to set of work flow rules.

2. **Information Extraction (IE) Service:** This is the key service of the system. This module reads input text from the messages queue, checks if the message contains information or a question, and in response sends the type of the message to the **MC** to determine the suitable work flow. In both cases, the **IE** is responsible for processing the text message. If it is an information message, the **IE** service reads the extraction rules from the Knowledge Base (**KB**), tries to extract the required information from the textual data, assigns some certainty factor to the extracted information and then passes this extracted information to the data integration service. In the case of a request message, the IE service then has to under-

stand what this question wants to find and passes the request keywords to the question answering system.

3. **Data Integration (DI) Service:** Data integration task comes in two ways. The first is to integrate the information provided by the **IE** service with the information already existing in the XML Database (**XMLDB**). It tries to find the information in the **XMLDB** that refers to the same geographical place mentioned by the IE, finds the conflicting facts, and tries to resolve such conflicts using the (**KB**) independently of the user by assigning several levels of certainty to each new piece of information. The second is to manage integrating data from Open Linked Data (**OLD**) web ontologies in a consistent and efficient manner to achieve the goals of the project. Data integration over **OLD** also implies uncertainty in the integrated data.

4. **Question Answering (QA) Service:** This service receives the request keywords from the IE service, formulates the XML query, runs this query on the DB, retrieves the results, applies some inference on the results using geo-ontology if needed and sends the results back to the user in the form of natural language generated text.

5. **Probabilistic Spatial XML Database (XMLDB):** This database is a standard probabilistic **XMLDB** that is extended to handle geospatial data. The information contained in this DB is assigned to some certainty factor that indicates how certain the information is. The data integration module is responsible for assigning this certainty factor.

6. **Knowledge Base (KB):** Holds set of rules needed for the extraction process. These rules are generated from a set of training texts. Also, it handles the probabilistic framework used for assigning probabilities to the possible locations, resolving conflicts between extracted information and those existing in the **XMLDB**.

7. **Open Linked Data (OLD):** All the modules make use of web ontologies to enrich and improve the data.

8. **Message Queue (MQ):** The queue of text messages received from users that need to be processed.

9. **Work Flow Rules (WFR):** These are the rules for activating intended modules on the basis of the type of message being processed.

Table A.1: Templates filled from users contributions.

Field	Template 1	Template 2	Template 3
Hotel_Name	Axel Hotel	movenpick hotel	Berlin hotel
Location	Berlin	Berlin	Berlin
Country	P(Germany)>P(USA)>P(...)	P(Germany)>P(USA)>P(...)	P(Germany)>P(USA)>P(...)
User_Attitude	P(Positive)>P(Negative)	P(Positive)>P(Negative)	P(Positive)>P(Negative)

We present functionality of the system components using an example to see how the system will operate. In this demonstration we deal with the tourism domain. Let us say that users send the following messages (actual tweets) to our system:

"*berlin has some nice hotels i just loved the hetero friendly love that word Axel Hotel in Berlin.*"

"*Good morning Berlin. The sun is out!!!! Very impressed by the customer service at #movenpick hotel in berlin. Well done guys*"

"*In Berlin hotel room, nice enough, weather grim however*"

Once a message is received, it is placed in the Message Queue (**MQ**). A signal is sent to the Module controller (**MC**) indicating that there is a new message is waiting for processing. The **MC** will then activate the Information Extraction (**IE**) module which fetches the message from the **MQ**, and classifies it as an *Informative Message*. A tag is then attached to the message on the **MQ** indicating its type. The **MC** checks the set of Work Flow Rules (**WFR**) according to the type of the message. The **IE** module is activated again by the MC to extract the information that is implied in the message. The IE uses the extraction rules stored in the Knowledge Base (**KB**) to extract the required information resolving the uncertainty about the city and country names with the aid of other geographical signals contained within the message. It is required to extract users' attitude towards some hotel along with its location (city). In this case, the **IE** module will extract the templates shown in table A.1.

A signal is sent back from the **IE** module to the **MC** declaring the end of the extraction process. The **MC** activates the **DI** module which receives the extracted information from the **IE** module. The **DI** module then is responsible of resolving the conflicts between the extracted information and the existing information. In the case that same information already exists in the **XMLDB**, the **DI** module has to modify the certainty factor attached with the existing

information using set of rules in the **KB**. In the case that the extracted piece of information is totally novel information, the **DI** module adds a record to the Hotels table in the **XMLDB** and attach a certainty factor based on the precision of the extraction of the locations names.

Now let us examine the other scenario, that of a user request. The user sends the following request:

"*Can anyone recommend a good, but not ridiculously expensive hotel right in the middle of Berlin?*"

The message is placed in the **MQ** and the **IE** is activated to indicate the type of the new message. The **IE** module marks the message as a *Request Message*. The **WFR** associated with *Request Message* is used by the **MC**. The **IE** extracts the keywords of the request (hotel, Berlin, good, not expensive). Then the **QA** module is activated and receives those keywords and formulates the suitable **XQuery** (assuming the existence of functions like topk, score):

topk(3, for $x in // Hotels
where $x/City == "Berlin" and $x/ User_Attitude == "Positive"
orderby score($x)
return $x)

The **XQuery** is applied on the **XMLDB** and the retrieved records are passed back again to the **QA** module which uses those records to form a natural language answer to the users' request. The answer is forwarded to the **MC**, which in turn forwards it to the user. The answer may be like this:

"*Some good hotels in Berlin are Axel Hotel, movenpick hotel, Berlin hotel.*"

APPENDIX B

Concept Extraction Challenge at #MSM2013

B.1 Summary

In this appendix, we present our contribution to the Making Sense of Microposts Workshop (#MSM2013) Concept Extraction Challenge, hosted in conjunction with the 2013 World Wide Web conference (WWW'13). The task was to extract entity concepts in Micropost data, characterized by a type and a value. For this task, we propose a hybrid approach for Named Entity Extraction (NEE) and Classification (NEC) for tweets. The system uses the power of the Conditional Random Fields (CRF) and the Support Vector Machines (SVM) in a hybrid way to achieve better results. For named entity type classification we use AIDA [62] disambiguation system to disambiguate the extracted named entities and then we use the Wikipedia categories of the disambiguated entities to find the type of the extracted mentions.

The contents of this appendix have been published as [114].

B.2 Introduction

B.2.1 The Task

Microposts are small fragments of social media content that have been published by users. Microposts have been used for a variety of applications (e.g., sentiment analysis, opinion mining, trend analysis), by gleaning useful information, often using third-party concept extraction tools. There has been a great need for such tools in the last few years, along with the creation and adoption of new methods for concept extraction.

The #MSM2013 challenge required participants to build semi-automated systems to identify concepts (entities) within Microposts and extract matching entity types for each concept identified, where concepts are defined as abstract notions of things. In order to focus the challenge we restricted the classification to four entity types:

1. Person (**PER**), e.g. Obama;
2. Organisation (**ORG**), e.g. NASA;
3. Location (**LOC**), e.g. New York;
4. Miscellaneous (**MISC**), consisting of the following: film/movie, entertainment award event, political event, programming language, sporting event and TV show.

Submissions were required to recognize these entity types within each Micropost, and extract the corresponding entity type-value tuples from the Micropost. Consider the following example, taken from the annotated corpus: *"870,000 people in canada depend on #food banks - 25% increase in the last 2 years - please give generously"* The fourth token in this Micropost refers to the location Canada; an entry to the challenge would be required to spot this token and extract it as an annotation, as:

LOC/canada;

For this task, we split the Named Entity Recognition (NER) task into two separate tasks: Named Entity Extraction (NEE) which aims only to detect entity mention boundaries in text; and Named Entity Classification (NEC) which assigns the extracted mention to its correct entity type. For NEE, we used a hybrid approach of CRF and SVM to achieve better results. For NEC, we first apply AIDA disambiguation system [62] to disambiguate the extracted named entities, then we use the Wikipedia categories of the disambiguated entities to find the type of the extracted mentions.

B.2.2 Dataset

The dataset consists of the message fields of each of 4341 manually annotated Microposts, on a variety of topics, including comments on the news and politics, collected from the end of 2010 to the beginning of 2011, with a 60% / 40% split between training and test data. The annotation of each Micropost in the training dataset gave all participants a common base from which to learn extraction patterns. The test dataset contained no annotations; the challenge

task was for participants to provide these. To assess the performance of the submissions the gold standard of the test set is used.

B.3 Proposed Approach

B.3.1 Named Entity Extraction

For this task, we made use of two famous state-of-the-art approaches for NER; CRF and SVM. We trained each of them in a different way as described below. The purpose of training is only for entity extraction rather recognition (extraction and classification). Results obtained from both are unionized to give the final extraction results.

Conditional Random Fields

CRF is a probabilistic model that is widely used for NER [22]. Despite the successes of CRF, the standard training of CRF can be very expensive [91] due to the global normalization. In this task, we used an alternative method called *empirical training* [92] to train a CRF model. The maximum likelihood estimation (MLE) of the empirical training has a closed form solution, and it does not need iterative optimization and global normalization. So empirical training can be radically faster than the standard training. Furthermore, the MLE of the empirical training is also a MLE of the standard training. Hence it can obtain competitive precision to the standard training. Tweet text is tokenized using special tweets tokenizer [85]. For each token, the following features are extracted and used to train the CRF: (a) The Part of Speech (POS) tag of the word provided by a special POS tagger designed for tweets [85]. (b) If the word initial character is capitalized or not. (c) If the word characters are all capitalized or not.

Support Vector Machines

SVM is a machine learning approach used for classification and regression problems. For our task, we used SVM to classify if a tweet segment is a named entity or not. The training process takes the following steps:

1. Tweet text is segmented using the segmentation approach as described in [75]. Each segment is considered a candidate for a named entity. We enriched the segments by looking up a Knowledge-Base (KB) (here we

use YAGO [6]) for possible entity mentions as described in chapter 7. The purpose of this step is to achieve high recall. To improve the precision, we applied filtering hypotheses (such as removing segments that are composed of stop words or having verb POS).

2. For each tweet segment, we extract the following set of features in addition to those features used for training the CRF:
   (a) The joint and the conditional probability of the segment obtained from Microsoft Web N-Gram services [90].
   (b) The stickiness of the segment as described in [75].
   (c) The segment frequency over around 5 million tweets [1].
   (d) If the segment appears in WordNet.
   (e) If the segment appears as a mention in Yago KB.
   (f) AIDA disambiguation system score for the disambiguated entity of that segment (if any).

   The selection of the SVM features is based on the claim that disambiguation clues can help in deciding if the segment is a mention for an entity or not.
3. An SVM with RBF kernel is trained whether the candidate segment represents a mention of NE or not.

We take the union of the CRF and SVM results, after removing duplicate extractions, to get the final set of annotations. For overlapping extractions we select the entity that appears in Yago, then the one having longer length.

B.3.2 Named Entity Classification

The purpose of NEC is to assign the extracted mention to its correct entity type. For this task, we first use the prior type probability of the given mention in the training data. If the extracted mention is out of vocabulary (does not appear in training set), we apply AIDA disambiguation system on the extracted mentions. AIDA provides the most probable entity for the mention. We get the Wikipedia categories of that entity from the KB to form an entity profile. Similarly, we use the training data to build a profile of Wikipedia categories for each of the entity types (PER, ORG, LOC and MISC).

[1] `http://wis.ewi.tudelft.nl/umap2011/` + TREC 2011 Microblog track collection.

Table B.1: Extraction results on training set (cross validation)

	Pre.	Rec.	F1
Twiner Seg.	0.0997	0.8095	0.1775
Yago	0.1489	0.7612	0.2490
Twiner∪Yago	0.0993	**0.8139**	0.1771
Filter(Twiner∪Yago)	0.2007	0.8066	0.3214
SVM	**0.7959**	0.5512	0.6514
CRF	0.7157	0.7634	0.7387
CRF∪SVM	0.7166	0.7988	**0.7555**

Table B.2: Extraction and classification results on training set (cross validation).

	Pre.	Rec.	F1
CRF	0.6440	0.6324	0.6381
AIDA Disambiguation + Entity Categorization	**0.6545**	**0.7296**	**0.6900**

To find the type of the extracted mention, we measure the document similarity between the entity profile and the profiles of the 4 entity types. We assign the mention to the type of the most similar profile.

If the extracted mention is out of vocabulary and is not assigned to an entity by AIDA we try to disambiguate the first token of it. If all those methods failed to find entity type for the mention we just assign PER type.

B.4 Experimental Results

B.4.1 Results on The Training Set

In this section we show our experimental results of the proposed approaches on the training data. All our experiments are done through a 4-fold cross validation approach for training and testing. We used precision, recall and F1 measures as evaluation criteria for those results. Table B.1 shows the NEE results along the extraction process phases. **Twiner Seg.** represents results of the tweet segmentation algorithm described in [75]. **Yago** represents results of the surface matching extraction as described in 7. **Twiner∪Yago** represents results of merging the output of the two aforementioned methods. **Filter(Twiner∪Yago)** represents results after applying filtering hypothesis. The purpose of those

steps is to achieve as much recall as possible with reasonable precision. **SVM** is trained as described in section B.3.1 to find which of the segments represent true NE. **CRF** is trained and tested on tokenized tweets to extract any NE regardless of its type . **CRF∪SVM** is the unionized set of results of both **CRF** and **SVM**. Table B.2 shows the final results of both, extraction with **CRF∪SVM** and entity classification using the method presented in section B.3.2 (**AIDA Disambiguation + Entity Categorization**). It also shows the **CRF** results when trained to recognize (extract and classify) NE. We considered it as our baseline. Our method of separating the extraction and classification outperforms the baseline.

B.4.2 Results on The Test Set

A total of 22 participants joined the challenge. A brief survey on their contributions can be found in section 6.3. Table B.3 reports results of the top 5 participants systems in terms of precision, recall and F1-measure on the test set. Our proposed system achieves the best precision and F1 results and ranked 3rd for recall results with a small margin to the 1st rank.

B.5 Conclusion

In this appendix, we present our approach for the #MSM2013 IE challenge. We split the NER task into two separate tasks: NEE which aims only to detect entity mention boundaries in text; and NEC which assigns the extracted mention to its correct entity type. For NEE we used a hybrid approach of CRF and SVM to achieve better results. For NEC we used AIDA disambiguation system to disambiguate the extracted named entities and then we use the Wikipedia categories of the disambiguated entities to find the type of the extracted mentions.

Table B.3: Top 5 participants results on test set.

(a) Precision.

Participant	PER	ORG	LOC	MISC	ALL
Habib, M. et al. [114]	**0.923**	0.673	**0.877**	0.622	**0.774**
Dlugolinský, S. et al. [78]	0.876	0.603	0.864	**0.714**	0.764
Mendes, P. et al. [115]	0.824	0.648	0.800	0.667	0.735
Van Erp, M. et al. [77]	0.879	0.686	0.844	0.525	0.734
Das, A. et al. [116]	0.809	**0.707**	0.746	0.636	0.724

(b) Recall.

Participant	PER	ORG	LOC	MISC	ALL
Dlugolinský, S. et al. [78]	0.938	**0.614**	0.613	0.287	**0.613**
Van Erp, M. et al. [77]	**0.952**	0.485	**0.739**	0.269	0.611
Habib, M. et al. [114]	0.908	0.611	0.620	0.277	0.604
Cortis, K. [117]	0.859	0.587	0.517	**0.418**	0.595
van Den Bosch, M. et al. [76]	0.926	0.463	0.682	0.122	0.548

(c) F1.

Participant	PER	ORG	LOC	MISC	ALL
Habib, M. et al. [114]	**0.920**	**0.640**	0.738	0.383	**0.670**
Dlugolinský, S. et al. [78]	0.910	0.609	0.721	**0.410**	0.662
Van Erp, M. et al. [77]	0.918	0.568	**0.790**	0.356	0.658
Cortis, K. [117]	0.833	0.611	0.618	0.377	0.610
Godin, F. et al. [79]	0.828	0.486	0.744	0.298	0.589

Bibliography

[1] Social networking reaches nearly one in four around the world. [Online]. Available: http://www.emarketer.com/Article/Social-Networking-Reaches-Nearly-One-Four-Around-World/1009976

[2] N. A. Chinchor, "Proceedings of the Seventh Message Understanding Conference (MUC-7) named entity task definition," in *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Fairfax, VA, April 1998, p. 21 pages, version 3.5, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/. [Online]. Available: http://acl.ldc.upenn.edu/muc7/ne_task.html

[3] M.-A. Abbasi, S.-K. Chai, H. Liu, and K. Sagoo, "Real-world behavior analysis through a social media lens," in *Proceedings of the 5th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, ser. SBP'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 18–26. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29047-3_3

[4] S. Yu and S. Kak, "A survey of prediction using social media," *CoRR*, vol. abs/1203.1647, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1203.html#abs-1203-1647

[5] T. Lin, Mausam, and O. Etzioni, "Entity linking at web scale," in *Proc. of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, 2012, pp. 84–88.

[6] J. Hoffart, F. Suchanek, K. Berberich, E. Kelham, G. de Melo, and G. Weikum, "Yago2: Exploring and querying world knowledge in time, space, context, and many languages," in *Proc. of WWW 2011*, 2011, pp. 229–232.

[7] A. E. C. Basave, A. Varga, M. Rowe, M. Stankovic, and A.-S. Dadzie, "Making sense of microposts (#msm2013) concept extraction challenge," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 1–15. [Online]. Available: http://ceur-ws.org/Vol-1019/

[8] M.-F. Moens, *Information extraction: algorithms and prospects in a retrieval context*. Springer, 2006, vol. 21.

[9] K. Kaiser and S. Miksch, "Information extraction," A Survey. Technical report, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Asgaard-TR-2005-6, Tech. Rep., 2005.

[10] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999, vol. 999.

[11] R. Grishman and B. Sundheim, "Message understanding conference - 6: A brief history," in *Proc. of Int'l Conf. on Computational Linguistics*, 1996, pp. 466–471.

[12] J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson, "Fastus: A system for extracting information from text," in *Proc. of Human Language Technology*, 1993, pp. 133–137.

[13] R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks, "University of Sheffield: Description of the LaSIE system as used for MUC-6," in *Proc. of MUC-6*, 1995, pp. 207–220.

[14] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks, "University of Sheffield: Description of the Lasie-II system as used for MUC-7," in *Proc. of MUC-7*, 1998.

[15] H. Cunningham, "GATE, a general architecture for text engineering," *Computers and the Humanities*, vol. 36, no. 2, pp. 223–254, 2002.

[16] D. E. Appelt and B. Onyshkevych, "The common pattern specification language," in *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*, ser. TIPSTER '98. Stroudsburg, PA, USA: Association for Computational Linguistics, 1998, pp. 23–30. [Online]. Available: http://dx.doi.org/10.3115/1119089.1119095

[17] M. B. Habib and M. van Keulen, "Information extraction, data integration, and uncertain data management: The state of the art," http://eprints.eemcs.utwente.nl/19808/, Centre for Telematics and Information Technology University of Twente, Enschede, Technical Report TR-CTIT-11-06, 2011.

[18] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-performance learning name-finder," in *In Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997, pp. 194–201.

[19] S. Sekine, "Nyu: Description of the japanese ne system used for met-2," in *Proc. of the Seventh Message Understanding Conference (MUC-7*, 1998.

[20] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman, "Nyu: Description of the mene named entity system as used in muc-7," in *In Proceedings of the Seventh Message Understanding Conference (MUC-7*, 1998.

[21] M. Asahara and Y. Matsumoto, "Japanese named entity extraction with redundant morphological analysis," *IPSJ SIG Notes*, vol. 2003, no. 4, pp. 49–56, jan 2003. [Online]. Available: http://ci.nii.ac.jp/naid/110002911617/en/

[22] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proc. of CoNLL 2003*, 2003, pp. 188–191.

[23] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260 – 269, 1967.

[24] H. Wallach, "Conditional random fields: An introduction," Department of Computer and Information Science, University of Pennsylvania, Tech. Rep. MS-CIS-04-21, 2004.

[25] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and Trends in Machine Learning*, 2011.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[27] Y. Yang and X. Liu, "A re-examination of text categorization methods," 1999.

[28] M. D. Lieberman and H. Samet, "Multifaceted toponym recognition for streaming news," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '11. New York, NY, USA: ACM, 2011, pp. 843–852. [Online]. Available: http://doi.acm.org/10.1145/2009916.2010029

[29] B. Pouliquen, M. Kimler, R. Steinberger, C. Ignat, T. Oellinger, K. Blackler, F. Fluart, W. Zaghouani, A. Widiger, A.-C. Forslund, and C. Best, "Geocoding multilingual texts: Recognition, disambiguation and visualisation," in *Proc. of LREC 2006*, 2006, pp. 53–58.

[30] X. Carreras, L. Màrques, and L. Padró, "Named entity extraction using adaboost," in *Proceedings of CoNLL-2002*. Taipei, Taiwan, 2002, pp. 167–170.

[31] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang, "Named entity recognition through classifier combination," in *Proc. of CoNLL-2003*, W. Daelemans and M. Osborne, Eds. Edmonton, Canada, 2003, pp. 168–171.

[32] G. Szarvas, R. Farkas, and A. Kocsor, "A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms," in *Proc. of the 9th international conference on Discovery Science*, ser. DS'06, 2006, pp. 267–278.

[33] A. E. Richman and P. Schone, "Mining wiki resources for multilingual named entity recognition," in *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 1–9. [Online]. Available: http://www.aclweb.org/anthology/P/P08/P08-1001

[34] J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran, "Learning multilingual named entity recognition from wikipedia," *Artificial Intelligence*, vol. 194, no. 0, pp. 151 – 175, 2013, <ce:title>Artificial Intelligence, Wikipedia and Semi-Structured Resources</ce:title>. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370212000276

[35] T. Zhang and D. Johnson, "A robust risk minimization based named entity recognition system," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, ser. CONLL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 204–207. [Online]. Available: http://dx.doi.org/10.3115/1119176.1119210

[36] A. O. Arnold, "Exploiting domain and task regularities for robust named entity recognition," Ph.D. dissertation, Pittsburgh, PA, USA, 2009, aAI3382435.

[37] S. Rued, M. Ciaramita, J. Mueller, and H. Schuetze, "Piggyback: Using search engines for robust cross-domain named entity recognition," in *49th Annual Meeting of the Association for Computational Linguistics (ACL-HLT)*, 2011, pp. 965–975. [Online]. Available: http://www.aclweb.org/anthology/P/P11/P11-1097.pdf

[38] N. Wacholder, Y. Ravin, and M. Choi, "Disambiguation of proper names in text," in *Proc. of ANLC 1997*, 1997, pp. 202–208.

[39] D. Buscaldi and P. Rosso, "A conceptual density-based approach for the disambiguation of toponyms," *Int'l Journal of Geographical Information Science*, vol. 22, no. 3, pp. 301–313, 2008.

[40] D. Smith and G. Crane, "Disambiguating geographic names in a historical digital library," in *Research and Advanced Technology for Digital Libraries*, ser. LNCS, vol. 2163, 2001, pp. 127–136.

[41] E. Rauch, M. Bukatin, and K. Baker, "A confidence-based framework for disambiguating geographic terms," in *Workshop Proc. of the HLT-NAACL 2003*, 2003, pp. 50–54.

[42] J. Overell and S. Ruger, "Place disambiguation with co-occurrence models," in *Proc. of CLEF 2006*, 2006.

[43] D. Smith and G. Mann, "Bootstrapping toponym classifiers," in *Workshop Proc. of HLT-NAACL 2003*, 2003, pp. 45–49.

[44] B. Martins, I. Anastácio, and P. Calado, "A machine learning approach for resolving place references in text," in *Proc. of AGILE 2010*, 2010.

[45] M. B. Habib and M. van Keulen, "Named entity extraction and disambiguation: The reinforcement effect." in *Proceedings of the 5th International Workshop on Management of Uncertain Data, MUD 2011, Seatle, USA*, ser. CTIT Workshop Proceedings Series, vol. WP11-02. Enschede: Centre for Telematics and Information Technology University of Twente, August 2011, pp. 9–16.

[46] M. B. Habib, "Neogeography: The challenge of channelling large and ill-behaved data streams," in *Workshops proc. of ICDE 2011*, 2011.

[47] D. Thakker, T. Osman, and P. Lakin, "Gate jape grammar tutorial," *Nottingham Trent University, UK, Phil Lakin, UK, Version*, vol. 1, 2009.

[48] S. Sekine and C. Nobata, "Definition, dictionaries and tagger for extended named entity hierarchy," in *Proc. of LREC 2004*, 2004, pp. 1977–1980.

[49] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[50] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *Proc. of the AAAI-98 Workshop on Learning for Text Categorization*, 1998, pp. 41–48.

[51] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. Myaeng, "Some effective techniques for naive bayes text classification," *IEEE Trans. on Knowledge and Data Engineering*, vol. 18, pp. 1457–1466, 2006, iSSN 1041-4347.

[52] M. B. Habib and M. van Keulen, "Improving toponym disambiguation by iteratively enhancing certainty of extraction," in *Proceedings of the 4th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2012, Barcelona, Spain*. Spain: SciTePress, October 2012, pp. 399–410.

[53] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *In ACL*, 2005, pp. 363–370.

[54] M. B. Habib and M. van Keulen, "A hybrid approach for robust multilingual toponym extraction and disambiguation," in *Proceedings of the International Conference on Language Processing and Intelligent Information Systems (LP&IIS 2013), Warsaw, Poland*, ser. Lecture Notes in Computer Science. Berlin: Springer Verlag, June 2013.

[55] B. Carpenter, "Character language models for chinese word segmentation and named entity recognition," in *Association for Computational Linguistics*, 2006, pp. 169–172.

[56] J. D. M. Rennie, "Using term informativeness for named entity detection," in *In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 2005, pp. 353–360.

[57] T. Furche, G. Grasso, G. Orsi, C. Schallhart, and C. Wang, "Automatically learning gazetteers from the deep web," in *Proceedings of the 21st International Conference Companion on World Wide Web*, ser. WWW '12 Companion. New York, NY, USA: ACM, 2012, pp. 341–344. [Online]. Available: http://doi.acm.org/10.1145/2187980.2188044

[58] R. C. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *In EACL*, 2006, pp. 9–16.

[59] S. Cucerzan, "Large-scale named entity disambiguation based on Wikipedia data," in *Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 708–716.

[60] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of wikipedia entities in web text," in *Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '09, 2009, pp. 457–466.

[61] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," in *Proc. of EMNLP 2011*, 2011.

[62] M. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum, "Aida: An online tool for accurate disambiguation of named entities in text and tables," vol. 4, no. 12, 2011, pp. 1450–1453.

[63] H. Srinivasan, J. Chen, and R. Srihari, "Cross document person name disambiguation using entity profiles," in *Proceedings of the Text Analysis Conference (TAC) Workshop*, 2009.

[64] C. Wang, K. Chakrabarti, T. Cheng, and S. Chaudhuri, "Targeted disambiguation of ad-hoc, homogeneous sets of named entities," in *Proc. of the 21st international conference on World Wide Web*, ser. WWW '12, 2012, pp. 719–728.

[65] D. Spina, E. Amigó, and J. Gonzalo, "Filter keywords and majority class strategies for company name disambiguation in twitter," in *Proc. of the Second international conference on Multilingual and multimodal information access evaluation*, ser. CLEF'11, 2011, pp. 50–61.

[66] S. R. Yerva, Z. Miklós, and K. Aberer, "Entity-based classification of twitter messages," *IJCSA*, vol. 9, no. 1, pp. 88–115, 2012.

[67] A. D. Delgado, R. Mart'ınez, A. Pérez Garc'ıa-Plaza, and V. Fresno, "Unsupervised Real-Time company name disambiguation in twitter," in *Workshop on Real-Time Analysis and Mining of Social Streams (RAMSS)*, 2012, pp. 25–28.

[68] M. Christoforaki, I. Erunse, and C. Yu, "Searching social updates for topic-centric entities," in *Proc. of the First International Workshop on Searching and Integrating New Web Data Sources - Very Large Data Search (VLDS)*, 2011, pp. 34–39.

[69] A. Davis, A. Veloso, A. S. da Silva, W. Meira, Jr., and A. H. F. Laender, "Named entity disambiguation in streaming data," in *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ser. ACL '12, 2012, pp. 815–824.

[70] T. Steiner, R. Verborgh, J. Gabarró Vallés, and R. Van de Walle, "Adding meaning to social network microposts via multiple named entity disambiguation apis and tracking their data provenance," *International Journal of Computer Information Systems and Industrial Management*, vol. 5, pp. 69–78, 2013.

[71] T. Westerveld, W. Kraaij, and D. Hiemstra, "Retrieving web pages using content, links, urls and anchors," in *Tenth Text REtrieval Conference, TREC 2001*, vol. SP 500, no. 500-25, 2002, pp. 663–672.

[72] L. Li, Z. Yu, J. Zou, L. Su, Y. Xian, and C. Mao, "Research on the method of entity homepage recognition," *Journal of Computational Information Systems (JCIS)*, vol. 5, no. 4, pp. 1617–1624, 2009.

[73] A. Ritter, S. Clark, Mausam, and O. Etzioni, "Named entity recognition in tweets: An experimental study." in *Proc. of EMNLP 2011*, 2011.

[74] J. J. Jung, "Online named entity recognition method for microtexts in social networking services: A case study of twitter," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8066–8070, 2012.

[75] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee, "Twiner: named entity recognition in targeted twitter stream," in *SIGIR*, 2012, pp. 721–730.

[76] A. van Den Bosch and T. Bogers, "Memory-based named entity recognition in tweets," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 40–43. [Online]. Available: http://ceur-ws.org/Vol-1019/paper_03.pdf

[77] M. V. Erp, G. Rizzo, and R. Troncy, "Learning with the web: Spotting named entities on the intersection of NERD and machine learning," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 27–30. [Online]. Available: http://ceur-ws.org/Vol-1019/paper_15.pdf

[78] Štefan Dlugolinský, P. Krammer, M. Ciglan, and M. Laclavík, "MSM2013 IE Challenge: Annotowatch," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 21–26. [Online]. Available: http://ceur-ws.org/Vol-1019/paper_21.pdf

[79] F. Godin, P. Debevere, E. Mannens, W. D. Neve, and R. V. de Walle, "Leveraging existing tools for named entity recognition in microposts," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 36–39. [Online]. Available: http://ceur-ws.org/Vol-1019/paper_25.pdf

[80] M. B. Habib and M. van Keulen, "Unsupervised improvement of named entity extraction in short informal context using disambiguation clues," in *Workshop on Semantic Web and Information Extraction, SWAIE 2012, Galway, Ireland*, ser. CEUR Workshop Proceedings, vol. 925. Germany: CEUR-WS.org, October 2012, pp. 1–10.

[81] D. Nadeau, P. D. Turney, and S. Matwin, "Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity," in *Proc. of 19th Canadian Conference on Artificial Intelligence*, 2006.

[82] M. B. Habib and M. van Keulen, "A generic open world named entity disambiguation approach for tweets," in *Proceedings of the 5th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2013, Vilamoura, Portugal*. Portugal: SciTePress, September 2013, pp. 267–276.

[83] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proc. of the 16th international conference on World Wide Web*, ser. WWW '07, 2007, pp. 697–706.

[84] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[85] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: annotation, features, and experiments," in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, ser. HLT '11, 2011, pp. 42–47.

[86] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Proc. of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '01, 2001, pp. 334–342.

[87] D. J. MacKay and L. C. B. Peto, "A hierarchical dirichlet language model," *Natural Language Engineering*, vol. 1, pp. 1–19, 1994.

[88] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[89] B. Locke and J. Martin, "Named entity recognition: Adapting to microblogging," Senior Thesis, University of Colorado, 2009.

[90] K. Wang, C. Thrasher, E. Viegas, X. Li, and B.-j. P. Hsu, "An overview of microsoft web n-gram corpus and applications," in *Proc. of the NAACL HLT 2010*, 2010, pp. 45–48.

[91] C. Sutton and A. McCallum, "Piecewise training of undirected models," in *Proc. of UAI*, 2005, pp. 568–575.

[92] Z. Zhu, D. Hiemstra, P. M. G. Apers, and A. Wombacher, "Closed form maximum likelihood estimator of conditional random fields," http://eprints.eemcs.utwente.nl/23097/, University of Twente, Technical Report TR-CTIT-13-03, 2013.

[93] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.

[94] A. Turner, *Introduction to neogeography*. O'Reilly Media, Inc., 2006.

[95] (2012) Internet usage statistics for africa. [Online]. Available: http://www.internetworldstats.com/stats1.htm#africa

[96] R. Myslewski. (2013) Increased cell phone coverage tied to uptick in african violence. [Online]. Available: http://www.theregister.co.uk/2013/06/18/increased_cell_phone_coverage_leads_to_increased_african_violence/

[97] (2010) The world in 2010: Ict facts and figure. International Telecommunication Union. [Online]. Available: www.itu.int/net/itunews/issues/2010/10/04.aspx

[98] F. Girardin, F. Calabrese, F. D. Fiore, C. Ratti, and J. Blat, "Digital footprinting: Uncovering tourists with user-generated content," *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 36–43, 2008.

[99] J. J. White and R. E. Roth, "Twitterhitter: Geovisual analytics for harvesting insight from volunteered geographic information," in *Proceedings of GIScience*, vol. 2010, 2010.

[100] B. De Longueville, N. Ostländer, and C. Keskitalo, "Addressing vagueness in volunteered geographic information (vgi)–a case study," *International Journal of Spatial Data Infrastructures Research*, vol. 5, pp. 1725–0463, 2010.

[101] S. E. Overell, "Geographic information retrieval: Classification, disambiguation and modelling," Ph.D. dissertation, Citeseer, 2009.

[102] J. Dykes, R. Purves, A. Edwardes, and J. Wood, "Exploring volunteered geographic information to describe place: visualization of the 'geograph british isles' collection," in *Proceedings of the GIS Research UK 16th Annual Conference GISRUK*, 2008, pp. 256–267.

[103] J. E. A. H. L. M. D. Purves, R. Dykes and J. Wood, "Describing the space and place of digital cities through volunteered geographic information," in *GeoViz Workshop on Contribution of Geovisualization to the concept of the Digital City*, 2009.

[104] I. Mani, J. Hitzeman, and C. Clark, "Annotating natural language geographic references," in *proc. LREC 2008-W13 Workshop on Methodologies and Resources for Processing Spatial Language*. Citeseer, 2008, pp. 11–15.

[105] C. Sallaberry, M. Gaio, J. Lesbegueries, and P. Loustau, "A semantic approach for geospatial information extraction from unstructured documents," *The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*, pp. 93–104, 2007.

[106] B. Martins, H. Manguinhas, and J. Borbinha, "Extracting and exploring the geotemporal semantics of textual resources," in *Semantic Computing, 2008 IEEE International Conference on*. IEEE, 2008, pp. 1–9.

[107] J. Strötgen and M. Gertz, "Timetrails a system for exploring spatiotemporal information in documents," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1569–1572, 2010.

[108] F. J. Lopez-Pellicer, M. J. Silva, M. S. Chaves, and C. Rodrigues, "Geographic ontologies production in grease-ii," 2009.

[109] G. N. Hess, C. Iochpe, and S. Castano, "Towards a geographic ontology reference model for matching purposes." in *GeoInfo*, 2007, pp. 35–47.

[110] W. Liu, H. Gu, C. Peng, and D. Cheng, "Ontology-based retrieval of geographic information," in *Geoinformatics, 2010 18th International Conference on*, 2010, pp. 1–6.

[111] D. Buscaldi, P. Rosso, and P. Peris, "Inferring geographical ontologies from multiple resources for geographical information retrieval." in *GIR*, 2006.

[112] V. R. Benjamins, J. Contreras, O. Corcho, and A. Gomez-perez, "Six challenges for the semantic web," in *In KR2002 Semantic Web Workshop*, 2002.

[113] T. Li, Q. Shao, and Y. Chen, "Pepx: a query-friendly probabilistic xml database," in *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006, pp. 848–849.

[114] M. Habib, M. V. Keulen, and Z. Zhu, "Concept extraction challenge: University of Twente at #msm2013," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 17–20. [Online]. Available: http://ceur-ws.org/Vol-1019/paper_14.pdf

[115] P. Mendes, D. Weissenborn, and C. Hokamp, "DBpedia Spotlight at the MSM2013 challenge," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 57–61. [Online]. Available: http://ceur-ws.org/Vol-1019/paper_30.pdf

[116] A. Das, U. Burman, B. Ar, and S. Bandyopadhyay, "NER from tweets: SRI-JU system @MSM 2013," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 62–66. [Online]. Available: http://ceur-ws.org/Vol-1019/paper_33.pdf

[117] K. Cortis, "ACE: A concept extraction approach using linked open data," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 31–35. [Online]. Available: http://ceur-ws.org/Vol-1019/paper_20.pdf

Author's Publications

2014

M. B. Habib, M. V. Keulen, and Z. Zhu, "Named entity extraction and linking challenge: University of Twente at #microposts2014," in *Proceedings of the #Microposts2014 NEEL Challenge*, 2014.

2013

M. B. Habib and M. van Keulen, "Toponym extraction and disambiguation enhancement using loops of feedback," in *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, ser. Communications in Computer and Information Science, A. Fred, J. Dietz, K. Liu, and J. Filipe, Eds. Springer Berlin Heidelberg, 2013, vol. 415, pp. 113–129.

M. B. Habib and M. van Keulen, "Named entity extraction and disambiguation: the missing link," in *Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval, ESAIR 2013, San Francisco, USA*. New York: ACM, October 2013, pp. 37–40.

M. B. Habib and M. van Keulen, "A generic open world named entity disambiguation approach for tweets," in *Proceedings of the 5th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2013, Vilamoura, Portugal*. Portugal: SciTePress, September 2013, pp. 267–276.

M. B. Habib and M. van Keulen, "A hybrid approach for robust multilingual toponym extraction and disambiguation," in *Proceedings of the International Conference on Language Processing and Intelligent Information Systems (LP&IIS 2013), Warsaw, Poland*, ser. Lecture Notes in Computer Science. Berlin: Springer Verlag, June 2013.

M. B. Habib, M. V. Keulen, and Z. Zhu, "Concept extraction challenge: Uni-

versity of Twente at #msm2013," in *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, 2013, pp. 17–20.

2012

M. B. Habib and M. van Keulen, "Unsupervised improvement of named entity extraction in short informal context using disambiguation clues," in *Workshop on Semantic Web and Information Extraction, SWAIE 2012, Galway, Ireland*, ser. CEUR Workshop Proceedings, vol. 925. Germany: CEUR-WS.org, October 2012, pp. 1–10.

M. B. Habib and M. van Keulen, "Improving toponym disambiguation by iteratively enhancing certainty of extraction," in *Proceedings of the 4th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2012, Barcelona, Spain*. Spain: SciTePress, October 2012, pp. 399–410.

M. B. Habib and M. van Keulen, "Improving toponym extraction and disambiguation using feedback loop," in *Proceedings of the 12th International Conference on Web Engineering (ICWE 2012), Berlin, Germany*, ser. Lecture Notes in Computer Science. Berlin: Springer Verlag, July 2012, pp. 439–443.

2011

M. van Keulen and M. B. Habib, "Handling uncertainty in information extraction," in *of the 7th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2011), Bonn, Germany*, ser. CEUR Workshop Proceedings, vol. 778. Aachen, Germany: CEUR-WS.org, October 2011, pp. 109–112.

M. B. Habib and M. van Keulen, "Named entity extraction and disambiguation: The reinforcement effect." in *Proceedings of the 5th International Workshop on Management of Uncertain Data, MUD 2011, Seatle, USA*, ser. CTIT Workshop Proceedings Series, vol. WP11-02. Enschede: Centre for Telematics and Information Technology University of Twente, August 2011, pp. 9–16.

M. B. Habib, "Neogeography: The challenge of channelling large and ill-behaved data streams," in *The ICDE 2011 Ph.D. Workshop, Hannover, Germany*, E. J. Neuhold and W. Siberski, Eds. USA: IEEE Computer Society, April 2011, pp. 284–287.

M. B. Habib and M. van Keulen, "Information extraction, data in-

tegration, and uncertain data management: The state of the art," http://eprints.eemcs.utwente.nl/19808/, Centre for Telematics and Information Technology University of Twente, Enschede, Technical Report TR-CTIT-11-06, 2011.

2009

T. F. Gharib, M. B. Habib, and Z. T. Fayed, "Arabic text classification using support vector machines," *International Journal of Computers and Their Applications*, vol. 16, no. 4, pp. 192–199, December 2009.

2008

M. B. Habib, "An intelligent system for automated arabic text categorization," Master's thesis, Computer Science Department, Faculty of Computers and Information Sciences, ain Shams University, Cairo, Egypt, February 2008.

2006

M. B. Habib, Z. T. Fayed, and T. F. Gharib, "A hybrid feature selection approach for arabic documents classification," *Egyptian Computer Science Journal*, vol. 28, no. 3, pp. 1–7, September 2006.

M. M. Syiam, Z. T. Fayed, and M. B. Habib, "An intelligent system for arabic text categorization," *International Journal of Intelligent Computing and Information Sciences*, vol. 6, no. 1, pp. 1–19, January 2006.

Summary

Social media content represents a large portion of all textual content appearing on the Internet. These streams of user generated content (UGC) provide an opportunity and challenge for media analysts to analyze huge amount of new data and use them to infer and reason with new information. An example of a main sector for social media analysis is the area of customer feedback through social media. With so many feedback channels, organizations can mix and match them to best suit corporate needs and customer preferences. Another beneficial sector is social security. Automatic monitoring and gathering of information posted in social media can be helpful to take actions to prevent violent and destructive behavior.

A main challenge of natural language is its ambiguity and vagueness. To automatically resolve ambiguity by computers, the grammatical structure of sentences is used, for instance, which groups of words go together (phrases) and which words are the subject or object of a verb. However, when we move to informal language widely used in social media, the language becomes more ambiguous and thus more challenging for automatic understanding.

Information Extraction (IE) is the research field that enables the use of unstructured text in a structured way. Named Entity Extraction (NEE) is a sub task of IE that aims to locate phrases (mentions) in the text that represent names of entities such as persons, organizations or locations regardless of their type. Named Entity Disambiguation (NED) is the task of determining which correct person, place, event, etc. is referred to by a mention.

The main goal of this thesis is to mimic the human way of recognition and disambiguation of named entities especially for domains that lack formal sentence structure. The proposed methods open the doors for more sophisticated applications based on users' contributions on social media. We propose a robust combined framework for NEE and NED in semi-formal and informal text. The achieved robustness has been proven to be valid across languages and domains and to be independent of the selected extraction and disambiguation techniques. It is also shown to be robust against shortness in labeled training

data and against the informality of the used language. We have discovered a reinforcement effect and exploited it a technique that improves extraction quality by feeding back disambiguation results. We present a method of handling the uncertainty involved in extraction to improve the disambiguation results. A generic approach for NED in tweets for any named entity (not entity oriented) is presented. This approach overcomes the problem of limited coverage of KBs. Mentions are disambiguated by assigning them to either a Wikipedia article or a home page. We also introduce a method to enrich the limited entity context.

Samenvatting

De content van sociale media vormt een groot deel van alle tekstuele content op het internet. Deze stromen van user generated content (UGC) vormen een mogelijkheid en uitdaging om enorme hoeveelheden nieuwe data te analyseren, en kunnen gebruikt worden om te redeneren met nieuwe informatie, en om nieuwe informatie te extraheren. Een voorbeeld van een grote tak voor social media analyse is het veld van klantterugkoppeling via social media. Door deze vele terugkoppelingskanalen met elkaar te integreren, kunnen organisaties deze gebruiken om zo goed mogelijk in te spelen op hun eigen behoeften en de voorkeuren van hun gebruikers. Het automatisch monitoren en verzamelen van nieuwe informatie op social media kan nuttig zijn om negatieve gedragingen te voorkomen.

Eén van de grote uitdagingen van natuurlijke taal is de mogelijkheid om deze op verschillende wijzen te interpreteren. Om automatisch de juiste interpretatie te kiezen, wordt de grammaticale opbouw van de zinnen gebruikt, bijvoorbeeld welke woorden bij elkaar horen (zinsdelen), en welke woorden het onderwerp of het lijdend voorwerp van een werkwoord zijn. Als we echter kijken naar de informele taal die wijd verbreid is op social media, neemt het aantal mogelijke interpretaties, en daarmee de uitdaging voor automatische interpretatie, toe. Information extraction (IE) is het onderzoeksgebied dat het mogelijk maakt om ongestructureerde tekst op een gestructureerde manier te gebruiken. Named Entity Extraction (NEE) is een deelgebied binnen IE met als doel zinsnedes te detecteren die namen van entiteiten, zoals personen, organisaties of locaties, representeren. Named Entity Disambiguation (NED) is een ander deelgebied, waarin men zich bezig houdt met het achterhalen welk(e) persoon, plaats of evenement bedoeld is.

Het hoofddoel van het onderzoek beschreven in dit proefschrift, is om het menselijk gedrag van het herkennen en interpreteren van entiteitsnamen na te bootsen, met name in het domein van informele zinsstructuren. De voorgestelde methodes vormen een basis voor verder ontwikkelde toepassingen die gebaseerd zijn op de bijdragen van gebruikers op social media. Wij

introduceren een robuust gecombineerd framework voor NEE en NED in semi-formele en informele tekst. De validateit van de behaalde robuustheid is bewezen in verschillende talen en domeinen, onafhankelijk van de gekozen extractie- en interpretatietechniek. Ook is de robuustheid tegen gebrek aan geannoteerde trainingsdata en de mate van informaliteit van de gebruikte taal bewezen. We hebben een versterkend effect ontdekt en benut door de interpretatieresultaten terug te koppelen. Wij presenteren een methode waarbij de onzekerheid bij het extractieproces gebruikt wordt om de resultaten van het interpretatieproces te verbeteren. Er wordt een generieke aanpak voor NED in tweets gepresenteerd. Met behulp van deze aanpak wordt het probleem van schaarsheid in kennisbanken opgelost. Referenties worden hierbij geïnterpreteerd door ze te koppelen aan een Wikipedia artikel of aan een website. Tot slot introduceren we ook een methode om de gelimiteerde context van een entiteit te verrijken.

SIKS Dissertations List

2014-19 Vincius Ramos (TUE), *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support.*
2014-18 Mattijs Ghijsen (UVA), *Methods and Models for the Design and Study of Dynamic Agent Organizations.*
2014-17 Kathrin Dentler (VU/UVA), *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability.*
2014-16 Krystyna Milian (VU), *Supporting trial recruitment and design by automatically interpreting eligibility criteria.*
2014-15 Nataliya Mogles (VU), *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare.*
2014-14 Yangyang Shi (TUD), *Language Models with Meta-information.*
2014-13 Arlette van Wissen (VU), *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains.*
2014-12 Willem van Willigen (VU), *Look Ma, No Hands: Aspects of Autonomous Vehicle Control.*
2014-11 Janneke van der Zwaan (TUD), *An Empathic Virtual Buddy for Social Support.*
2014-10 Ivan Razo-Zapata (VU), *Service Value Networks.*
2014-09 Philip Jackson (UvT), *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language.*
2014-08 Samur Araujo (TUD), *Data Integration over Distributed and Heterogeneous Data Endpoints.*
2014-07 Arya Adriansyah (TUE), *Aligning Observed and Modeled Behavior.*
2014-06 Damian Tamburri (VU), *Supporting Networked Software Development.*
2014-05 Jurriaan van Reijsen (UU), *Knowledge Perspectives on Advancing Dynamic Capability.*
2014-04 Hanna Jochmann-Mannak (UT), *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation.*
2014-03 Sergio Raul Duarte Torres (UT), *Information Retrieval for Children: Search Behavior and Solutions.*
2014-02 Fiona Tuliyano (RUN), *Combining System Dynamics with a Domain Modeling Method.*
2014-01 Nicola Barile (UU), *Studies in Learning Monotone Models from Data.*
2013-43 Marc Bron (UVA), *Exploration and Contextualization through Interaction and*

Concepts.
2013-42 Leon Planken (TUD), *Algorithms for Simple Temporal Reasoning.*
2013-41 Jochem Liem (UVA), *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning.*
2013-40 Pim Nijssen (UM), *Monte-Carlo Tree Search for Multi-Player Games.*
2013-39 Joop de Jong (TUD), *A Method for Enterprise Ontology based Design of Enterprise Information Systems.*
2013-38 Eelco den Heijer (VU), *Autonomous Evolutionary Art.*
2013-37 Dirk Borner (OUN), *Ambient Learning Displays.*
2013-36 Than Lam Hoang (TUe), *Pattern Mining in Data Streams.*
2013-35 Abdallah El Ali (UvA), *Minimal Mobile Human Computer Interaction.*
2013-34 Kien Tjin-Kam-Jet (UT), *Distributed Deep Web Search.*
2013-33 Qi Gao (TUD), *User Modeling and Personalization in the Microblogging Sphere.*
2013-32 Kamakshi Rajagopal (OUN), *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development.*
2013-31 Dinh Khoa Nguyen (UvT), *Blueprint Model and Language for Engineering Cloud Applications.*
2013-30 Joyce Nakatumba (TUE), *Resource-Aware Business Process Management: Analysis and Support.*
2013-29 Iwan de Kok (UT), *Listening Heads.*
2013-28 Frans van der Sluis (UT), *When Complexity becomes Interesting: An Inquiry into the Information eXperience.*
2013-27 Mohammad Huq (UT), *Inference-based Framework Managing Data Provenance.*
2013-26 Alireza Zarghami (UT), *Architectural Support for Dynamic Homecare Service Provisioning.*
2013-25 Agnieszka Anna Latoszek-Berendsen (UM), *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System.*
2013-24 Haitham Bou Ammar (UM), *Automated Transfer in Reinforcement Learning.*
2013-23 Patricio de Alencar Silv (UvT), *Value Activity Monitoring.*
2013-22 Tom Claassen (RUN), *Causal Discovery and Logic.*
2013-21 Sander Wubben (UvT), *Text-to-text generation by monolingual machine translation.*
2013-20 Katja Hofmann (UvA), *Fast and Reliable Online Learning to Rank for Information Retrieval.*
2013-19 Renze Steenhuizen (TUD), *Coordinated Multi-Agent Planning and Scheduling.*
2013-18 Jeroen Janssens (UvT), *Outlier Selection and One-Class Classification.*
2013-17 Koen Kok (VU), *The PowerMatcher: Smart Coordination for the Smart Electricity Grid.*
2013-16 Eric Kok (UU), *Exploring the practical benefits of argumentation in multi-agent deliberation.*
2013-15 Daniel Hennes (UM), *Multiagent Learning - Dynamic Games and Applications.*
2013-14 Jafar Tanha (UVA), *Ensemble Approaches to Semi-Supervised Learning Learning.*
2013-13 Mohammad Safir (UT), *Service Tailoring: User-centric creation of integrated*

IT-based homecare services to support independent living of elderly.
2013-12 Marian Razavia (VU), *Knowledge-driven Migration to Services.*
2013-11 Evangelos Pournara (TUD), *Multi-level Reconfigurable Self-organization in Overlay Services.*
2013-10 Jeewanie Jayasinghe Arachchig (UvT), *A Unified Modeling Framework for Service Design.*
2013-09 Fabio Gori (RUN), *Metagenomic Data Analysis: Computational Methods and Applications.*
2013-08 Robbert-Jan Mer (VU), *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators.*
2013-07 Giel van Lankveld (UvT), *Quantifying Individual Player Differences.*
2013-06 Romulo Goncalve (CWI), *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience.*
2013-05 Dulce Pumareja (UT), *Groupware Requirements Evolutions Patterns.*
2013-04 Chetan Yadat (TUD), *Coordinating autonomous planning and scheduling.*
2013-03 Szymon Klarman (VU), *Reasoning with Contexts in Description Logics.*
2013-02 Erietta Liarou (CWI), *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing.*
2013-01 Viorel Milea (EUR), *News Analytics for Financial Decision Support.*
2012-51 Jeroen de Jong (TUD), *Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching.*
2012-50 Steven van Kervel (TUD), *Ontologogy driven Enterprise Information Systems Engineering.*
2012-49 Michael Kaisers (UM), *Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions.*
2012-48 Jorn Bakker (TUE), *Handling Abrupt Changes in Evolving Time-series Data.*
2012-47 Manos Tsagkias (UVA), *Mining Social Media: Tracking Content and Predicting Behavior.*
2012-46 Simon Carter (UVA), *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation.*
2012-45 Benedikt Kratz (UvT), *A Model and Language for Business-aware Transactions.*
2012-44 Anna Tordai (VU), *On Combining Alignment Techniques.*
2012-43 Withdrawn , .
2012-42 Dominique Verpoorten (OU), *Reflection Amplifiers in self-regulated Learning.*
2012-41 Sebastian Kelle (OU), *Game Design Patterns for Learning.*
2012-40 Agus Gunawan (UvT), *Information Access for SMEs in Indonesia.*
2012-39 Hassan Fatemi (UT), *Risk-aware design of value and coordination networks.*
2012-38 Selmar Smit (VU), *Parameter Tuning and Scientific Testing in Evolutionary Algorithms.*
2012-37 Agnes Nakakawa (RUN), *A Collaboration Process for Enterprise Architecture Creation.*
2012-36 Denis Ssebugwawo (RUN), *Analysis and Evaluation of Collaborative Modeling*

Processes.
2012-35 Evert Haasdijk (VU), *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics.*
2012-34 Pavol Jancura (RUN), *Evolutionary analysis in PPI networks and applications.*
2012-33 Rory Sie (OUN), *Coalitions in Cooperation Networks (COCOON).*
2012-32 Wietske Visser (TUD), *Qualitative multi-criteria preference representation and reasoning.*
2012-31 Emily Bagarukayo (RUN), *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure.*
2012-30 Alina Pommeranz (TUD), *Designing Human-Centered Systems for Reflective Decision Making.*
2012-29 Almer Tigelaar (UT), *Peer-to-Peer Information Retrieval.*
2012-28 Nancy Pascall (UvT), *Engendering Technology Empowering Women.*
2012-27 Hayrettin Gurkok (UT), *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games.*
2012-26 Emile de Maat (UVA), *Making Sense of Legal Text.*
2012-25 Silja Eckartz (UT), *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application.*
2012-24 Laurens van der Werff (UT), *Evaluation of Noisy Transcripts for Spoken Document Retrieval.*
2012-23 Christian Muehl (UT), *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction.*
2012-22 Thijs Vis (UvT), *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?.*
2012-21 Roberto Cornacchia (TUD), *Querying Sparse Matrices for Information Retrieval.*
2012-20 Ali Bahramisharif (RUN), *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing.*
2012-19 Helen Schonenberg (TUE), *What's Next? Operational Support for Business Process Execution.*
2012-18 Eltjo Poort (VU), *Improving Solution Architecting Practices.*
2012-17 Amal Elgammal (UvT), *Towards a Comprehensive Framework for Business Process Compliance.*
2012-16 Fiemke Both (VU), *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment.*
2012-15 Natalie van der Wal (VU), *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.*
2012-14 Evgeny Knuto (TUE), *Generic Adaptation Framework for Unifying Adaptive Web-based Systems.*
2012-13 Suleman Shahid (UvT), *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions.*
2012-12 Kees van der Sluijs (TUE), *Model Driven Design and Data Integration in Semantic Web Information Systems.*
2012-11 J.C.B. Rantham Prabhakara (TUE), *Process Mining in the Large: Preprocessing,*

Discovery, and Diagnostics.
2012-10 David Smits (TUE), *Towards a Generic Distributed Adaptive Hypermedia Environment.*
2012-09 Ricardo Neisse (UT), *Trust and Privacy Management Support for Context-Aware Service Platforms.*
2012-08 Gerben de Vries (UVA), *Kernel Methods for Vessel Trajectories.*
2012-07 Rianne van Lambalgen (VU), *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions.*
2012-06 Wolfgang Reinhardt (OU), *Awareness Support for Knowledge Workers in Research Networks.*
2012-05 Marijn Plomp (UU), *Maturing Interorganisational Information Systems.*
2012-04 Jurriaan Souer (UU), *Development of Content Management System-based Web Applications.*
2012-03 Adam Vanya (VU), *Supporting Architecture Evolution by Mining Software Repositories.*
2012-02 Muhammad Umai (VU), *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models.*
2012-01 Terry Kakeeto (UvT), *Relationship Marketing for SMEs in Uganda.*
2011-49 Andreea Niculescu (UT), *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality.*
2011-48 Mark Ter Maat (UT), *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent.*
2011-47 Azizi Bin Ab Azi (VU), *Exploring Computational Models for Intelligent Support of Persons with Depression.*
2011-46 Beibei Hu (TUD), *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work.*
2011-45 Herman Stehouwer (UvT), *Statistical Language Models for Alternative Sequence Selection.*
2011-44 Boris Reuderink (UT), *Robust Brain-Computer Interfaces.*
2011-43 Henk van der Schuur (UU), *Process Improvement through Software Operation Knowledge.*
2011-42 Michal Sindlar (UU), *Explaining Behavior through Mental State Attribution.*
2011-41 Luan Ibraimi (UT), *Cryptographically Enforced Distributed Data Access Control.*
2011-40 Viktor Clerc (VU), *Architectural Knowledge Management in Global Software Development.*
2011-39 Joost Westra (UU), *Organizing Adaptation using Agents in Serious Games.*
2011-38 Nyree Lemmens (UM), *Bee-inspired Distributed Optimization.*
2011-37 Adriana Burlutiu (RUN), *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference.*
2011-36 Erik van der Spek (UU), *Experiments in serious game design: a cognitive approach.*
2011-35 Maaike Harbers (UU), *Explaining Agent Behavior in Virtual Training.*
2011-34 Paolo Turrini (UU), *Strategic Reasoning in Interdependence: Logical and Game-*

theoretical Investigations.
2011-33 Tom van der Weide (UU), *Arguing to Motivate Decisions.*
2011-32 Nees-Jan van Eck (EUR), *Methodological Advances in Bibliometric Mapping of Science.*
2011-31 Ludo Waltman (EUR), *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality.*
2011-30 Egon van den Broek (UT), *Affective Signal Processing (ASP): Unraveling the mystery of emotions.*
2011-29 Faisal Kamiran (TUE), *Discrimination-aware Classification.*
2011-28 Rianne Kaptei (UVA), *Effective Focused Retrieval by Exploiting Query Context and Document Structure.*
2011-27 Aniel Bhulai (VU), *Dynamic website optimization through autonomous management of design patterns.*
2011-26 Matthijs Aart Pontier (VU), *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots.*
2011-25 Syed Waqar ul Qounain Jaffry (VU)), *Analysis and Validation of Models for Trust Dynamics.*
2011-24 Herwin van Welbergen (UT), *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior.*
2011-23 Wouter Weerkamp (UVA), *Finding People and their Utterances in Social Media.*
2011-22 Junte Zhang (UVA), *System Evaluation of Archival Description and Access.*
2011-21 Linda Terlouw (TUD), *Modularization and Specification of Service-Oriented Systems.*
2011-20 Qing Gu (VU), *Guiding service-oriented software engineering - A view-based approach.*
2011-19 Ellen Rusman (OU), *The Mind ' s Eye on Personal Profiles.*
2011-18 Mark Ponsen (UM), *Strategic Decision-Making in complex games.*
2011-17 Jiyin He (UVA), *Exploring Topic Structure: Coherence, Diversity and Relatedness.*
2011-16 Maarten Schadd (UM), *Selective Search in Games of Different Complexity.*
2011-15 Marijn Koolen (UvA), *The Meaning of Structure: the Value of Link Evidence for Information Retrieval.*
2011-14 Milan Lovric (EUR), *Behavioral Finance and Agent-Based Artificial Markets.*
2011-13 Xiaoyu Mao (UvT), *Airport under Control. Multiagent Scheduling for Airport Ground Handling.*
2011-12 Carmen Bratosin (TUE), *Grid Architecture for Distributed Process Mining.*
2011-11 Dhaval Vyas (UT), *Designing for Awareness: An Experience-focused HCI Perspective.*
2011-10 Bart Bogaert (UvT), *Cloud Content Contention.*
2011-09 Tim de Jong (OU), *Contextualised Mobile Media for Learning.*
2011-08 Nieske Vergunst (UU), *BDI-based Generation of Robust Task-Oriented Dialogues.*

2011-07 Yujia Cao (UT), *Multimodal Information Presentation for High Load Human Computer Interaction.*
2011-06 Yiwen Wang (TUE), *Semantically-Enhanced Recommendations in Cultural Heritage.*
2011-05 Base van der Raadt (VU), *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.*
2011-04 Hado van Hasselt (UU), *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms.*
2011-03 Jan Martijn van der Werf (TUE), *Compositional Design and Verification of Component-Based Information Systems.*
2011-02 Nick Tinnemeie (UU), *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language.*
2011-01 Botond Cseke (RUN), *Variational Algorithms for Bayesian Inference in Latent Gaussian Models.*
2010-53 Edgar Meij (UVA), *Combining Concepts and Language Models for Information Access.*
2010-52 Peter-Paul van Maanen (VU), *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention.*
2010-51 Alia Khairia Amin (CWI), *Understanding and supporting information seeking tasks in multiple sources.*
2010-50 Bouke Huurnink (UVA), *Search in Audiovisual Broadcast Archives.*
2010-49 Jahn-Takeshi Saito (UM), *Solving difficult game positions.*
2010-47 Chen Li (UT), *Mining Process Model Variants: Challenges, Techniques, Examples.*
2010-46 Vincent Pijpers (VU), *e3alignment: Exploring Inter-Organizational Business-ICT Alignment.*
2010-45 Vasilios Andrikopoulos (UvT), *A theory and model for the evolution of software services.*
2010-44 Pieter Bellekens (TUE), *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain.*
2010-43 Peter van Kranenburg (UU), *A Computational Approach to Content-Based Retrieval of Folk Song Melodies.*
2010-42 Sybren de Kinderen (VU), *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach.*
2010-41 Guillaume Chaslot (UM), *Monte-Carlo Tree Search.*
2010-40 Mark van Assem (VU), *Converting and Integrating Vocabularies for the Semantic Web.*
2010-39 Ghazanfar Farooq Siddiqui (VU), *Integrative modeling of emotions in virtual agents.*
2010-38 Dirk Fahland (TUE), *From Scenarios to components.*
2010-37 Niels Lohmann (TUE), *Correctness of services and their composition.*
2010-36 Jose Janssen (OU), *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification.*
2010-35 Dolf Trieschnigg (UT), *Proof of Concept: Concept-based Biomedical Information*

Retrieval.
2010-34 Teduh Dirgahayu (UT), *Interaction Design in Service Compositions.*
2010-33 Robin Aly (UT), *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval.*
2010-32 Marcel Hiel (UvT), *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems.*
2010-31 Victor de Boer (UVA), *Ontology Enrichment from Heterogeneous Sources on the Web.*
2010-30 Marieke van Erp (UvT), *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval.*
2010-29 Stratos Idreo (CWI), *Database Cracking: Towards Auto-tuning Database Kernels.*
2010-28 Arne Koopman (UU), *Characteristic Relational Patterns.*
2010-27 Marten Voulon (UL), *Automatisch contracteren.*
2010-26 Ying Zhang (CWI), *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines.*
2010-25 Zulfiqar Ali Memon (VU), *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective.*
2010-24 Dmytro Tykhonov , *Designing Generic and Efficient Negotiation Strategies.*
2010-23 Bas Steunebrink (UU), *The Logical Structure of Emotions.*
2010-22 Michiel Hildebrand (CWI), *End-user Support for Access to Heterogeneous Linked Data.*
2010-21 Harold van Heerde (UT), *Privacy-aware data management by means of data degradation.*
2010-20 Ivo Swartjes (UT), *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative.*
2010-19 Henriette Cramer (UvA), *People's Responses to Autonomous and Adaptive Systems.*
2010-18 Charlotte Gerritsen (VU), *Caught in the Act: Investigating Crime by Agent-Based Simulation.*
2010-17 Spyros Kotoulas (VU), *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications.*
2010-16 Sicco Verwer (TUD), *Efficient Identification of Timed Automata, theory and practice.*
2010-15 Lianne Bodenstaff (UT), *Managing Dependency Relations in Inter-Organizational Models.*
2010-14 Sander van Splunter (VU), *Automated Web Service Reconfiguration.*
2010-13 Gianluigi Folino (RUN), *High Performance Data Mining using Bio-inspired techniques.*
2010-12 Susan van den Braak (UU), *Sensemaking software for crime analysis.*
2010-11 Adriaan Ter Mors (TUD), *The world according to MARP: Multi-Agent Route Planning.*
2010-10 Rebecca Ong (UL), *Mobile Communication and Protection of Children.*
2010-09 Hugo Kielman (UL), *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging.*

2010-08 Krzysztof Siewicz (UL), *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments.*
2010-07 Wim Fikkert (UT), *Gesture interaction at a Distance.*
2010-06 Sander Bakkes (UvT), *Rapid Adaptation of Video Game AI.*
2010-05 Claudia Hauff (UT), *Predicting the Effectiveness of Queries and Retrieval Systems.*
2010-04 Olga Kulyk (UT), *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments.*
2010-03 Joost Geurts (CWI), *A Document Engineering Model and Processing Framework for Multimedia documents.*
2010-02 Ingo Wassink (UT), *Work flows in Life Science.*
2010-01 Matthijs van Leeuwen (UU), *Patterns that Matter.*
2009-46 Loredana Afanasiev (UvA), *Querying XML: Benchmarks and Recursion.*
2009-45 Jilles Vreeken (UU), *Making Pattern Mining Useful.*
2009-44 Roberto Santana Tapia (UT), *Assessing Business-IT Alignment in Networked Organizations.*
2009-43 Virginia Nunes Leal Franqueira (UT), *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients.*
2009-42 Toine Bogers (UvT), *Recommender Systems for Social Bookmarking.*
2009-41 Igor Berezhnyy (UvT), *Digital Analysis of Paintings.*
2009-40 Stephan Raaijmakers (UvT), *Multinomial Language Learning: Investigations into the Geometry of Language.*
2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin), *Service Substitution – A Behavioral Approach Based on Petri Nets.*
2009-38 Riina Vuorikari (OU), *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context.*
2009-37 Hendrik Drachsler (OUN), *Navigation Support for Learners in Informal Learning Networks.*
2009-36 Marco Kalz (OUN), *Placement Support for Learners in Learning Networks.*
2009-35 Wouter Koelewijn (UL), *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling.*
2009-34 Inge van de Weerd (UU), *Advancing in Software Product Management: An Incremental Method Engineering Approach.*
2009-33 Khiet Truong (UT), *How Does Real Affect Affect Affect Recognition In Speech?.*
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU), *Architectural Knowledge Management: Supporting Architects and Auditors.*
2009-31 Sofiya Katrenko (UVA), *A Closer Look at Learning Relations from Text.*
2009-30 Marcin Zukowski (CWI), *Balancing vectorized query execution with bandwidth-optimized storage.*
2009-29 Stanislav Pokraev (UT), *Model-Driven Semantic Integration of Service-Oriented Applications.*
2009-28 Sander Evers (UT), *Sensor Data Management with Probabilistic Models.*
2009-27 Christian Glahn (OU), *Contextual Support of social Engagement and Reflection on*

the Web.
2009-26 Fernando Koch (UU), *An Agent-Based Model for the Development of Intelligent Mobile Services.*
2009-25 Alex van Ballegooij (CWI), *RAM: Array Database Management through Relational Mapping".*
2009-24 Annerieke Heuvelink (VUA), *Cognitive Models for Training Simulations.*
2009-23 Peter Hofgesang (VU), *Modelling Web Usage in a Changing Environment.*
2009-22 Pavel Serdyukov (UT), *Search For Expertise: Going beyond direct evidence.*
2009-21 Stijn Vanderlooy (UM), *Ranking and Reliable Classification.*
2009-20 Bob van der Vecht (UU), *Adjustable Autonomy: Controling Influences on Decision Making.*
2009-19 Valentin Robu (CWI), *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets.*
2009-18 Fabian Groffen (CWI), *Armada, An Evolving Database System.*
2009-17 Laurens van der Maaten (UvT), *Feature Extraction from Visual Data.*
2009-16 Fritz Reul (UvT), *New Architectures in Computer Chess.*
2009-15 Rinke Hoekstra (UVA), *Ontology Representation - Design Patterns and Ontologies that Make Sense.*
2009-14 Maksym Korotkiy (VU), *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA).*
2009-13 Steven de Jong (UM), *Fairness in Multi-Agent Systems.*
2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin), *Operating Guidelines for Services.*
2009-11 Alexander Boer (UVA), *Legal Theory, Sources of Law & the Semantic Web.*
2009-10 Jan Wielemaker (UVA), *Logic programming for knowledge-intensive interactive applications.*
2009-09 Benjamin Kanagwa (RUN), *Design, Discovery and Construction of Service-oriented Systems.*
2009-08 Volker Nannen (VU), *Evolutionary Agent-Based Policy Analysis in Dynamic Environments.*
2009-07 Ronald Poppe (UT), *Discriminative Vision-Based Recovery and Recognition of Human Motion.*
2009-06 Muhammad Subianto (UU), *Understanding Classification.*
2009-05 Sietse Overbeek (RUN), *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality.*
2009-04 Josephine Nabukenya (RUN), *Improving the Quality of Organisational Policy Making using Collaboration Engineering.*
2009-03 Hans Stol (UvT), *A Framework for Evidence-based Policy Making Using IT.*
2009-02 Willem Robert van Hage (VU), *Evaluating Ontology-Alignment Techniques.*
2009-01 Rasa Jurgelenaite (RUN), *Symmetric Causal Independence Models.*