INFORMATION EXTRACTION FOR CLINICAL NARRATIVES

BY

PRATEEK JINDAL

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Professor Dan Roth, Chair
Professor Carl A. Gunter
Professor ChengXiang Zhai
Professor Wendy Chapman, UCSD

Abstract

Recent US government initiatives have made available a large number of Electronic Health Records (EHRs). These EHRs contain valuable information which can be used in Clinical Decision Support (CDS). So, Information Extraction (IE) from EHRs is a very promising research area. In this thesis, I focus on two tasks namely Mention Detection and Coreference Resolution. A lot of domain knowledge is available regarding clinical narratives. There are also several tools like SpecialistLexicalTools, MetaMap, etc. which help in analyzing clinical narratives. I integrate the domain knowledge and features derived from these tools in the local statistical models. Clinical narratives have a very special format which gives several interconnections between the tasks of mention detection and coreference resolution. A joint formulation for these two tasks has been presented in this thesis. Along with this, there is also a discussion regarding joint formulation for finding the mention types together. Soft constraints have been used while formulating the inference tasks. Softening the constraints is helpful because it allows the constraints to be violated during inference. Joint formulation is based on the fact that only local models are learned in the training phase. Inconsistencies in the decisions based on local models are resolved during the global inference step. I report the best results, to date, on end-to-end coreference resolution. The joint formulation presented in this thesis is very general and would benefit other information extraction tasks as well. I have made the systems described in this thesis publicly available for research use.

To Mercy

Acknowledgments

This research was supported by Grant HHS 90TR0003/01 and by Intelligence Advanced Research Projects Activity (IARPA) Foresight and Understanding from Scientific Exposition (FUSE) program via Department of Interior National Business Center (DoI/NBC) contract number D11PC2015. Its contents are solely the responsibility of the author and do not necessarily represent the official views, either expressed or implied, of the HHS, IARPA, DoI/NBC or the US government.

My stay in UIUC has been very pleasant. UIUC is one of the best universities in US and has people from very diverse backgrounds. It, thus, provides an excellent opportunity for intellectual growth of an individual. During my Ph.D., I made friends with several people and am grateful to all of them to help me become a better person. It is by God's grace that one gets the opportunity to meet with good people. Living with others helped me to look inwards and get closer to my true self.

Adviser plays a very important role in the life of a graduate student. I am fortunate to have a wonderful adviser, Prof. Dan Roth, who is well-versed in the areas of Natural Language Processing and Machine Learning. Prof. Roth has been very encouraging to me and allowed me to grow at a natural pace. One of the things that I liked the most about him is that he regularly interacts with all his students (through group meetings, individual meetings and weekly reports) and provides valuable feedback. He has a very understanding nature and is very pleasant to talk to. Even though Prof. Roth is very successful, he is very sober and easily excuses the minor mistakes of his students (without being annoyed!).

Other than my adviser, my committee members included Prof. Carl Gunter, Prof. Chengxiang Zhai and Prof. Wendy Chapman. All of them provided valuable sugges-

tions during both prelim and final exam. Prof. Gunter is also the director of the SHARPS grant that supported me and was the first one to explain me my role in SHARPS. From time-to-time, he helped me to align my work with the goals of SHARPS and also facilitated my communication with people from ONC. I am thankful to US Government (headed by Mr. Barrack Obama) for supporting the SHARPS grant.

My qual committee members included Prof. Gerald deJong (chair), Prof. David Forsyth and Prof. Julia Hockenmaier. I am thankful to all of them for being very nice with me during the exam. I took two courses with Prof. Hockenmaier and both these courses were very helpful in preparing for the qual exam. I am also thankful to Prof. L.V. Kale who first invited me to UIUC. I worked with him for about an year. He has a very friendly attitude and I learned a lot while working in his lab (Parallel Programming Laboratory).

During my Ph.D., I took 10 courses which were taught by Prof. L.V. Kale, Prof. Dan Roth (2 courses), Prof. Jeff Erickson, Prof. Julia Hockenmaier (2 courses), Prof. Eyal Amir, Prof. Pierre Moulin, Prof. David Forsyth and Prof. Rob Rutenbar. In addition, I took 1 individual study with Prof. Dan Roth. I also audited 2 optimization classes by Dr. Bernard Lidicky and Prof. Angelia Nedich. I am thankful to all these professors for giving valuable lessons during the classes. I particularly liked the teaching style of Prof. Pierre Moulin (Pattern Recognition class) and Dr. Bernard Lidicky (Optimization class).

When I started working in my research group (Cogcomp), I was quite crude and didn't understand many of the things. In this regard, my interactions with Cogcompers have been very helpful. Mark Sammons and Nick Rizzolo were the first ones to help me get started in Cogcomp. Later, I received a lot of help from Quang Do who gave me some good pieces of his code and many useful pointers which served as important building blocks in my research. I am also very thankful to Kai-Wei Chang who helped me to understand some of the difficult concepts regarding machine learning. I shared office space with Jeff Pasternack for more than 2 and a half years. He was very nice with me during all this time for which I am thankful to him. After Jeff moved to California, I have been sharing the space with Max Isenbolt and he is nice with me as well. Other Cogcompers with who I had wonderful discussions include Vivek Srikumar, Gourab

Kundu, Ming-Wei Chang, Vinod Vydiswaran, Rajhans Samdani, Lev Ratinov, Yuancheng Tu, Dan Goldwasser, Wei Lu and Yee-Seng Chan.

US is so different from India and I am thankful to all my friends who helped me to settle here. They taught me so many things about day-to-day life like cooking, driving, shopping etc. I had the fortune to be in the company of someone or the other for most of my evenings in last 4 years. I really enjoy the discussions with my friends and I hope these would continue in future. Dancing, listening to music and traveling with my friends are some of my favorite activities.

By the mercy of God, I have also received wonderful parents who always supported me in times of need. Of course, it is not possible to exhaustively acknowledge everyone's contributions in my life. By God's arrangement, there are so many living entities who are helping me all the time without even my knowing. The only way in which I can pay back to all of them is by trying to play my own role nicely in His creation.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CRF	Conditional Random Field
EHR	Electronic Health Record
HIE	Health Information Exchange
HMM	Hidden Markov Model
IE	Information Extraction
ILP	Integer Linear Programming
IQP	Integer Quadratic Programming
NLP	Natural Language Processing
POS	Part-of-Speech Label
Timex	Temporal Expression
UMLS	Unified Medical Language System

Publications

A part of the work presented in this thesis has been published in the following papers:

1. P. Jindal and D. Roth. "Extraction of Events and Temporal Expressions from Clinical Narratives". *Journal of Biomedical Informatics (JBI) - 2013*.

2. P. Jindal and D. Roth. "Using Soft Constraints in Joint Inference for Clinical Concept Recognition". *Proceedings of International Conference on Empirical Methods in Natural Language Processing (EMNLP) - 2013*.

3. P. Jindal and D. Roth. "End-to-End Coreference Resolution for Clinical Narratives". *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI) - 2013*. pp 2106-2112.

4. P. Jindal and D. Roth. "Using Domain Knowledge and Domain-Inspired Discourse Model for Coreference Resolution for Clinical Narratives". *Journal of American Medical Informatics Association (JAMIA)* **20(2)** - *2013*. pp 356-362.

5. P. Jindal, D. Roth and C.A. Gunter. "Detecting Privacy-Sensitive Events in Medical Text - 2013". *UIUC CS Technical Report*.

6. P. Jindal and D. Roth. "Using Knowledge and Constraints to Find the Best Antecedent". *Proceedings of International Conference on Computational Linguistics (COLING) - 2012*. pp 1327-1342.

7. P. Jindal and D. Roth. "Learning from Negative Examples in Set-Expansion". *Proceedings of IEEE International Conference on Data Mining (ICDM) - 2011*. pp 1110-1115.

Resources

Based on the work presented in this thesis, following resources have been made publicly available. All these resources are available on the software and demos sections of the website: http://cogcomp.cs.illinois.edu/

1. Medical NER: This is a software package[1] which annotates the input text with the following annotations: `TEST`, `TRE` and `PROB`.

2. Medical Coreference Resolution: This is a software package[2] which finds the coreference chains associated with medical entities.

3. Drug Abuse Detector: This is a software package[3] which finds the instances of drug abuse events in the input text.

4. Medical NER Demo: This is a web demo[4] based on Medical NER software.

5. Drug Abuse Detector Demo: This is a web demo[5] based on Drug Abuse Detector software.

[1] http://cogcomp.cs.illinois.edu/page/software_view/MedNER
[2] http://cogcomp.cs.illinois.edu/page/software_view/MedCoref
[3] http://cogcomp.cs.illinois.edu/page/software_view/MedSHARPS
[4] http://cogcomp.cs.illinois.edu/page/demo_view/mednerdemo
[5] http://cogcomp.cs.illinois.edu/page/demo_view/drugabuserecognizer

Chapter 1

Introduction

Health information technology (HIT) became an active topic of research when President Obama made "*computerization of health care*" a key part of his *American Recovery and Reinvestment Act (ARRA)* of 2009, his economic stimulus package. On Jan. 8th, 2009, he stated the value of HIT both in improving health care and creating jobs as well as set a goal of all Americans having their medical records in electronic form within five years. The recent US government initiatives that promote the use of electronic health records (EHRs) provide opportunities to mine patient notes as more and more health care institutions adopt EHRs.

Information extraction from EHRs is critical for several applications. Computerized Clinical Decision Support (CDS) aims to aid decision making of health care providers and the public by providing easily accessible health-related information at the point and time it is needed. Natural Language Processing (NLP) is instrumental in using free-text information to drive CDS, representing clinical knowledge and CDS interventions in standardized formats and leveraging clinical narrative. Today, a major portion of the patients clinical observations, including radiology reports, operative notes, and discharge summaries are recorded as narrative text (dictated and transcribed, or directly entered into the system by care providers). And in some systems even laboratory and medication records are only available as part of the physician's notes. Moreover, in some cases, the facts that should activate a CDS system can be found only in the free text.

1.1 Corpora

Research on Information Extraction in the general English domain dates back to 1960s and 1970s. However, research on Information Extraction for the clinical text has been more recent. A significant barrier to progress in coreference resolution in the clinical domain has been the lack of a shared annotated corpus to serve as a training and test bed for both rule-based and machine learning methods, with the latter requiring much more data. It has been noted that the biomedical texts differ from newswire. Similarly, clinical text manifests its own patterns as well, as clinical text are generally cursory, not edited, and abound with idiosyncratic shorthands. This further exemplifies the importance of a corpus in the clinical domain. Shared tasks like the i2b2/VA Challenge are addressing this barrier in part [1]. Shared tasks provide annotated datasets to participants and sometimes to nonparticipants (i2b2 datasets are available to others a year after the Challenge). The i2b2 shared task is standardizing its corpus as much as possible - the same records are used from one year to the next with layers of annotation that build on each other, and common input/output specifications are applied every year.

1.2 Scope of this Thesis

In this thesis, I will focus on the following tasks of Information Extraction:

1. Set Expansion
2. Timex (Temporal Expression) Extraction
3. Mention Detection
4. Coreference Resolution

1.3 Thesis Contributions

Domain-Specific Knowledge: Because of very different vocabulary of clinical texts, state-of-the-art tools on general English text don't work well on clinical text. To get good performance on clinical text, it is necessary to incorporate domain-specific knowledge. There are several knowledge sources that are available for medical text. For example, UMLS, or Unified Medical Language System, is a set of files and software that brings together many health and biomedical vocabularies and standards to enable interoperability between computer systems. MetaMap is a configurable program which maps biomedical text to the UMLS Metathesaurus. There are also several biomedical ontologies like MeSH, SNOMED CT etc. In this thesis, we show the use of several such knowledge sources in IE tasks. We also designed clinical descriptors which provides several features which are useful for generalization.

Joint Inference: The specific structure of the clinical narratives provides us special opportunities to improve the IE tasks. It is often the case that different components of information extraction are related to one another. Thus, it is advantageous to model these components jointly to leverage their interactions. This falls in the area of structured prediction problems where the output consists of several interacting variables and an NLP system needs to make global decisions which respect the mutual dependencies between variables. Most of the previous work in clinical NLP has considered only standard techniques to solve IE tasks where different tasks are solved independently. Sometimes, heuristics are used to post-process the results so that inconsistencies can be tackled. Purely statistical models for structured prediction problems tend to violate the constraints of the problem. Incorporating the information related to problem's constraints directly into statistical models is quite difficult because constraints generally involve long-range dependencies. Such long-range dependencies can make the model very expressive and thus, difficult to learn using limited training data. In this thesis, I

have discussed the ways to model the IE tasks jointly. My joint formulation of IE tasks is related to some of the previous works on CCMs [2, 3, 4, 5] which make it possible to effectively use task and domain-specific constraints without complicating the underlying statistical models.

Introduction of Integer Quadratic Programs (IQPs): Previously, in NLP literature, researchers have widely used Integer Linear Programs (ILPs) to model joint inference. In this thesis, we introduce the use of IQPs to solve joint inference. IQPs are more general than ILPs. In principle, it is possible to reduce the IQPs to ILPs. However, such conversion can lead to exponentially large ILPs. Thus, IQPs provide strict modeling advantage over ILPs. We also show that IQPs can be efficiently solved using modern solvers like Gurobi etc. Using IQPs, we integrated soft constraints in the application of mention detection and showed that soft constraints give considerable performance improvement over hard constraints. We were able to do exact inference even while using soft constraints. Previously, for soft constraints, only approximate inference was used.

Best results for Coreference Resolution: We made several advances in the task of coreference resolution. We exploited the discourse structure of clinical narratives to improvise several constraints which gave significant performance improvement for this task. We also showed that different pronouns behave quite differently and thus, it is advantageous to build separate models for resolving different types of pronouns. We managed to get the best results on coreference resolution for both i2b2 and ODIE datasets. We get the best results for both the cases: (a) when gold mentions are already given and (b) for end-to-end coreference resolution.

1.4 Organization of the Thesis

Rest of this thesis is organized as follows:

1. Chapter 2 describes some of the preliminaries. In particular, it gives the background of supervised coreference resolution and also introduces the basic models for structured prediction. It also gives the background of IE in biological domain which is closely related to medical domain.

2. Chapter 3 describes the task of set expansion. The experiments in this chapter were actually carried out on the news domain. However, similar ideas apply to clinical domain as well.

3. Chapter 4 describes the background and state-of-the-art methods for mention detection in clinical domain. It describes in detail the contribution of the features derived from domain-specific knowledge sources. It also describes the joint approach for mention detection in clinical domain.

4. Chapter 5 describes the extraction of temporal expressions.

5. Chapter 6 discusses the privacy concerns regarding the use of clinical narratives.

6. Chapter 7 describes the background and state-of-the-art methods for coreference resolution for both cases where gold mentions are already given and for the case of end-to-end coreference resolution. It describes the contribution of domain-specific knowledge sources in detail.

7. Chapter 8 describes coreference resolution for person mentions.

8. Chapter 9 describes a joint approach for coreference resolution.

9. Chapter 10 provides the conclusions. We also identify several directions for future work.

Chapter 2

Background and Preliminaries

2.1 Domain-Specific Knowledge Sources

Following subsections explain the domain-specific knowledge sources which have been used in this thesis.

2.1.1 UMLS

The purpose of the National Library of Medicine Unified Medical Language System (UMLS) is to facilitate the development of computer systems that behave as if they "understand" the meaning of the language of biomedicine and health. The UMLS provides data for system developers as well as search and report functions for less technical users.

There are three UMLS Knowledge Sources:

1. The Metathesaurus, which contains over one million biomedical concepts from over 100 source vocabularies

2. The Semantic Network, which defines 133 broad categories and fifty-four relationships between categories for labeling the biomedical domain

3. The SPECIALIST Lexicon and Lexical Tools, which provide lexical information and programs for language processing

These 3 knowledge sources are shown in Figure 2.1. The UMLS Terminology Services (UTS) provides Internet access to the three UMLS Knowledge Sources and to the UMLS

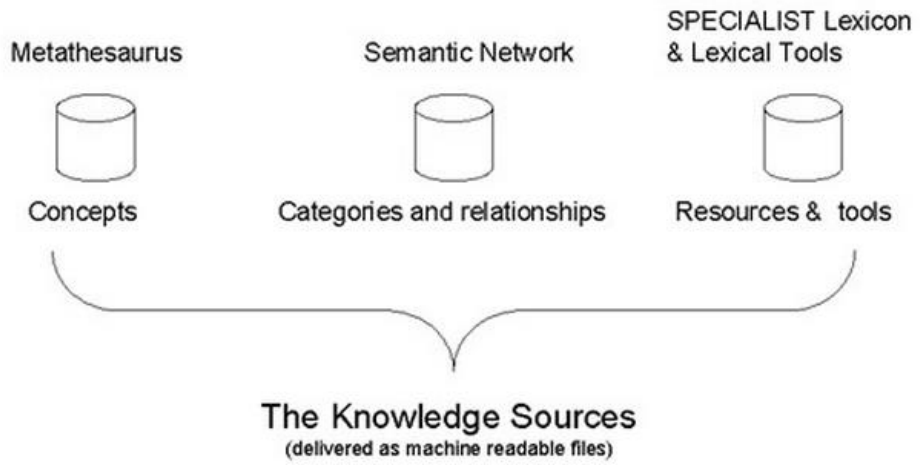


Figure 2.1: This figure shows the three knowledge sources of UMLS.

tools. Users can access the UTS after requesting a UMLS Metathesaurus license and creating a UTS account. MetamorphoSys is a free tool distributed with the UMLS. It is used to create a custom Metathesaurus subset and is needed to install the most current UMLS Knowledge Sources.

2.1.2 MeSH

Medical Subject Headings (MeSH) is a controlled vocabulary produced by the National Library of Medicine. The 2009 version of MeSH contains a total of 25186 subject headings, also known as descriptors. Most of these are accompanied by a short description or definition, links to related descriptors, and a list of synonyms or very similar terms (known as entry terms). Because of these synonym lists, MeSH can also be viewed as a thesaurus.

Descriptor hierarchy: The descriptors or subject headings are arranged in a hierarchy. A given descriptor may appear at several locations in the hierarchical tree. The tree locations carry systematic labels known as tree numbers, and consequently one descriptor can carry several tree numbers. For example, Figure 2.2 shows the hierarchy associated with the descriptor "Myocardial Infarction". The tree numbers of a given descriptor are

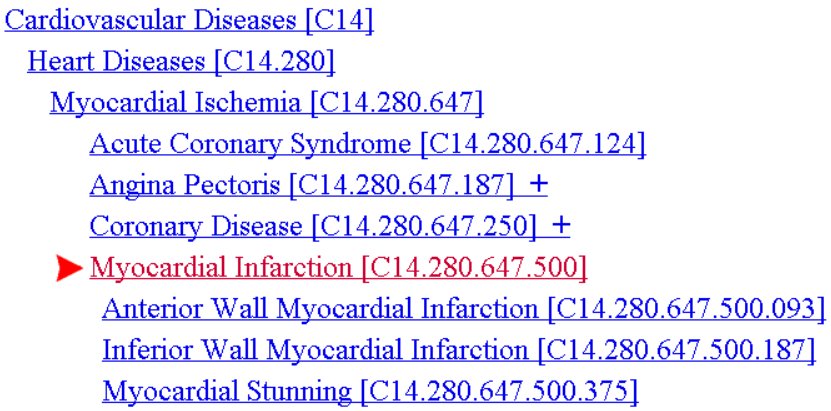


Figure 2.2: This figure shows the concept Myocardial Infarction in MeSH.

subject to change as MeSH is updated. Every descriptor also carries a unique alphanumerical ID that will not change.

2.1.3 SNOMED CT

SNOMED CT (Systematized Nomenclature Of Medicine Clinical Terms), is a systematically organised computer processable collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting. SNOMED CT is considred to be the most comprehensive, multilingual clinical healthcare terminology in the world. The primary purpose of SNOMED CT is to encode the meanings that are used in health information and to support the effective clinical recording of data with the aim of improving patient care. SNOMED CT consists of four primary core components:

1. Concept Codes - numerical codes that identify clinical terms, primitive or defined, organized in hierarchies

2. Descriptions - textual descriptions of Concept Codes

3. Relationships - relationships between Concept Codes that have a related meaning

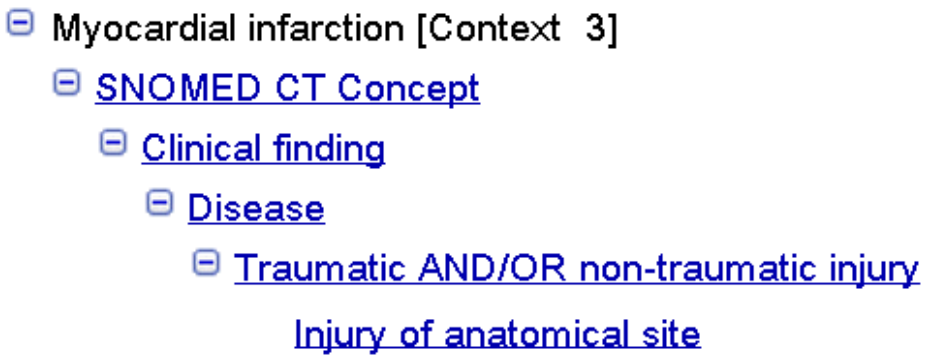


Figure 2.3: This figure shows the concept Myocardial Infarction in SNOMED CT.

4. Reference Sets - used to group Concepts or Descriptions into sets, including reference sets and cross-maps to other classifications and standards.

SNOMED CT "Concepts" are representational units that categorize all the things that characterize health care processes and need to be recorded therein. In 2011, SNOMED CT includes more than 311,000 concepts, which are uniquely identified by a concept ID, i.e. the concept 22298006 refers to Myocardial infarction. All SNOMED CT concepts are organized into acyclic taxonomic (is-a) hierarchies; for example, Figure 2.3 shows the hierarchy associated with the concept "Myocardial Infarction".

2.1.4 MetaMap

MetaMap [6] is a widely available program providing access from biomedical text to the concepts in the unified medical language system (UMLS) Metathesaurus. MetaMap arose in the context of an effort to improve biomedical text retrieval, specifically the retrieval of MEDLINE/PubMed citations. It provided a link between the text of biomedical literature and the knowledge, including synonymy relationships, embedded in the Metathesaurus.

MetaMap's default human-readable output generated from the input text "Patient had a heart attack few decades before." is shown in Figure 2.4. In this example, MetaMap


```
>>>>> Candidates
Meta Candidates (Total=8; Excluded=1; Pruned=0; Remaining=7)
   771   Decade [Quantitative Concept]
   673   Heart attack (Myocardial Infarction) [Disease or Syndrome]
   637   Heart [Body Part, Organ, or Organ Component]
   637   Attack, NOS (Onset of illness) [Temporal Concept]
   637   attack (Attack behavior) [Social Behavior]
   637   Heart (Entire heart) [Body Part, Organ, or Organ Component]
   637   Attack (Observation of attack) [Finding]
   604 E [X]Attacked (Assault) [Injury or Poisoning]
<<<<< Candidates
>>>>> Mappings
Meta Mapping (773):
   673   Heart attack (Myocardial Infarction) [Disease or Syndrome]
   771   Decade [Quantitative Concept]
<<<<< Mappings
```

Figure 2.4: This figure shows an example of MetaMap output.

identified 8 Metathesaurus candidates. The final mapping selected by MetaMap is also shown in the figure.

2.2 Background on Mention Detection

Mention Detection (or Named Entity Recognition (NER)) is a widely studied problem in general English text. Research on mention detection started as early as 1991 [7] on general English text. Initial approaches to NER were primarily rule-based approaches [8]. Since 1996, there has been an increase in the use of machine learning techniques [9, 10, 11, 12] to solve the NER task. Researchers have explored the NER task using different approaches: supervised learning, semi-supervised learning and unsupervised learning. CoNLL-2003 shared task [13] focussed on the following types of named entities: persons, locations and organizations.

In clinical text, NER problem is relatively new. In 2010, i2b2 organized a challenge [14] on concept recognition in clinical text. Participants primarily relied on supervised learn-

ing approach in this task. The models used by various teams for concept extraction can be categorized as follows:

1. CRFs: The most effective concept extraction systems [15, 16, 17] used CRFs. CRF implementations that were used included MALLET [18], CRF++, etc.

2. Semi-Markov HMM: de Bruijn et al. [19] used a semi-markov HMM, trained using passive-aggressive (PA) online updates. Semi-Markov models are Hidden Markov Models that tag multitoken spans of text, as opposed to single tokens. These models do not require a Begin/Inside/Outside (BIO) tagging formalism for Information Extraction tasks; instead, only four tags are needed: outside, problem, treatment, and test. Outside is constrained to tag only single words, while the others can tag spans up to 30 tokens in length.

3. Ensembles: Some teams [20] developed several variations of their systems and then used voting to find the final assignments. Other teams [15] developed hybrid systems which combined rule-based and machine learning approaches.

4. SVMs: One of the teams [21] used SVMs for finding concept types. Several feature selection methods like greedy forward, greedy forward/ backward and genetic algorithms were used to find representative features.

The major strength of the best systems came from feature engineering. Below we describe some of the features used by the best systems:

1. Token Features: punctuation, prefix/stem patterns, word shape information, whether brackets mismatched

2. Syntactic Features: POS of words appearing in a window of fixed size

3. Context Features: Words before/after each word, word bi/tri/quad-grams, skip n-grams, uncased word, pattern-based entity, uncased previous word

4. Sentence Features: sentence-length indicators, casefolding patterns, presence of digits, enumeration tokens at the start, a colon at the end, and whether verbs indicate past or future tense

5. Section features: including headings, assumed to be the most recently seen all-caps line ending with a colon, and subsection headings, assumed to be the most recently seen mixed-case line ending with a colon

6. Document features: including upper-case/lower-case patterns seen across the document, and a document length indicator

7. Semantic features: These consist of the following:

   (a) UMLS: UMLS was used to derive CUI (concept unique identifiers) for concepts. Some systems also used CUIs of concepts' parents. UMLS also provides the semantic type for each concept.

   (b) GENIA: Few systems used NLP tools based on GENIA. GENIA is a corpus which was developed to support biological information extraction. GENIA based lemma and entity types were used as features.

   (c) MetaMap: MetaMap provides a shallow parse for clinical sentences. It also maps clinical text to biomedical vocabularies.

   (d) WordNet: WordNet's synsets and hierarchy of concepts were used as features.

   (e) Wikipedia: Wikipedia provides categories for each concept. These categories and redirect pages of Wikipedia were used as features.

   (f) Brown Clusters: 7-bit hierarchical brown clusters help to solve the sparsity problems.

   (g) Publicly Available Systems: cTAKES, MedLEE, KnowledgeMap and Dictionary-based Semantic Tagger (DST)

2.3 Background on Clinical Coreference Resolution

2.3.1 Medical Nominal Resolution

Here we give an overview of the models used by researchers for clinical coreference resolution.

1. Rule-based Models: Such systems [22] developed rules in accordance with the annotation guidelines. While performing coreference resolution, a precedence order among the rules is followed. If there is a conflict between any two rules regarding the coreference decision, the rule with the higher precedence is selected. These systems always make pairwise decisions. These systems did not give the best results in the shared task. Overall, machine-learning based systems performed better than rule-based systems.

2. Supervised Pairwise Models: Like rule-based models, these systems [23, 24, 25] also make pairwise decisions. However, they use machine-learning techniques to train the pairwise classifier. Some systems [24] consider all possible pairs for coreference whereas other systems [25] consider a subset of all possible pairs. These systems typically use a large number of features. The best performing system [24] in i2b2 shared task was a pairwise classifier. Some of these systems also use an anaphoricity classifier as one of the features while making the coreference decision. Other systems filter the candidate pairs with the anaphoricity classifier before making coreference decisions.

3. Sieve-based Models: These systems [26, 27] are similar in spirit to that of Raghunathan et al. [28]. They make coreference decisions in several stages where the more precise decisions are made first. However, different systems vary in the exact implementation of the respective stages. Sieve-based models gave a reasonably good performance in i2b2 shared task.

End-to-End Coreference Resolution: There has only been a limited work towards end-to-end coreference resolution. One of the important works in this regard is that of Zheng et al. [29, 30]. They use the inbuilt cTAKES NER component to get the candidate mentions. Before performing coreference resolution, they have an intermediate step of candidate consolidation where they try to align the candidate mentions with the mention boundaries as given by a syntactic parser. Finally, coreference resolution is performed with a pairwise model as described above.

2.3.2 Medical Pronominal Resolution

For pronominal resolution, researchers have used both rule-based and machine learning methods. Zheng et al. [30] used Hobbs' algorithm [31] for resolving relative pronouns. Gooch and Roudsari [22] developed regular expressions using JAPE in a GATE framework to determine pleonastic cases. Pleonastics are filtered out during preprocessing and don't participate in coreference. Gooch and Roudsari [22] make use of centering theory [32] in pronominal resolution. Anaphoric pronouns are resolved against the most recent antecedent with the same grammatical role (e.g. subject, object, indirect object etc.). Uncategorized third-person plural pronouns were coreferenced with plural mentions with grammatical role agreement in the absence of intervening plural Person mentions.

Xu et al. [24] adopted a machine learning approach to coreference resolution where they trained a multi-class classifier to predict the type of the antecedent that a pronoun may be referring to. They don't train an anaphoricity classifier separately. However, they include the type null in their multi-class classifier to identify zero-anaphora cases. They used SVM to train multi-class classifier.

2.3.3 Person Resolution

In clinical narratives, coreference resolution for person class is more restricted than that in news text. This is because of the fact that in clinical narratives, the number of people involved are quite few. Clinical report mainly talks about the patient. Then there are few references to the doctors who treated the patient. And finally, there are some mentions of the family members of the patient.

Gooch and Roudsari [22] notes that the clinical reports are de-identified. During de-identification, the person names are replaced with dummy strings. Such replacement makes the problem of coreference resolution somewhat harder because of loss of some information (like gender). To perform coreference resolution, they used several gazetteers (family relations, gender identifiers, role identifiers for doctors etc.). They classified pronouns as belonging to global scope or local scope. For string matching, they used several libraries like SecondString Java Library, Jaro-Winkler [33] and Monge-Elkan [34, 35] metrics.

Xu et al. [24] followed a different approach for person coreference resolution. They trained a binary classifier to predict coreferential pairs. They introduced a new feature called "Patient class" which was used to identify whether a particular mention referred to a patient or not. They used the output of this classifier as a feature in their pairwise classifier.

2.4 Background on Supervised Coreference Resolution

Good surveys on coreference research are available [36, 37, 38, 39]. So, we give here only a brief overview. In 1970s and 1980s, several centering algorithms [32] were proposed for coreference resolution. Examples are focussing [40, 41], centering [42, 43], etc. In 1990s, focus shifted to machine learning approaches because of MUC conferences. In the next few subsections, we present an overview of famous coreference models.

2.4.1 Mention-Pair Model

It was first proposed by Aone and Bennett [44] and McCarthy and Lehnert [45]. In this model, first a pairwise classifier makes decision on each pair. And then a clustering mechanism is used for constructing coreference chains. Some of the important clustering algorithms include correlation clustering [46, 47, 48, 49], graph-partitioning [50] and Bell-Tree [51].

Traditionally, the task of anaphoricity determination has been tackled independently of coreference resolution using a variety of techniques. For example, pleonastic it has been identiiňĄed using heuristic approaches (e.g., Lappin and Leass [52], Kennedy and Boguraev [53]), supervised approaches (e.g., Evans [54], Muller [55], Versley et al. [56]), and distributional methods (e.g., Bergsma et al. [57]); and non-anaphoric definite descriptions have been identified using rule-based techniques (e.g., Vieira and Poesio [58]) and unsupervised techniques (e.g., Bean and Riloff (1999))

2.4.2 Entity-Mention model

Entity-Mention model [59, 60, 61] addresses the expressiveness problem with the mention-pair model.

2.4.3 Ranking Model

Ranking models address the problem of identifying the most probable candidate antecedent. Some examples include Tournament model [62], twin candidate model [63, 64] and cluster ranking model [65].

Commercial Toolkits for coreference resolution include JavaRAP [66], GuiTaR [67], BART [68], CoRTex [69], the Illinois Coreference Package (Bengtson and Roth, 2008), CherryPicker (Rahman and Ng, 2009), Reconcile [70], and Charniak and Elsner's [71] pronoun resolver.

2.4.4 Biomedical Coreference

There are some works on biomedical coreference resolution as well. Examples include [72], [73], [74], [75], [76], [77] and [78].

2.5 Sequence Tagging Models and General Structure Prediction Models

We now consider sequential tagging models and general structure prediction models.

2.5.1 Generative Model: Hidden Markov Model

Generative models specify a joint probability distribution over observations and the corresponding output structures. Many generative models have been proposed for structured prediction tasks [79, 80]. In the following, we review a very popular sequential generative model: the Hidden Markov Model (HMM). A (first-order) HMM is a generative model which models the joint probability of a series of tokens $\mathbf{x}$ and a sequence assignment $\mathbf{y}$. HMMs make an independence assumption that allows one to write the joint probability of $(\mathbf{x}, \mathbf{y})$ as follows:

$$P(\mathbf{x}, \mathbf{y}) = P(y_1) \prod_{i=2}^{T} P(y_i|y_{i-1}) \prod_{i=1}^{T} P(x^i|y^i), \tag{2.1}$$

where x^i is the i-th token in the input sequence, y^i is the i-th token in the output sequence, T is the number of tokens in this sequence, $P(y^1)$ represents the prior probabilities, $P(y^i|y^{i-1})$ represents the transition probabilities and $P(x^i|y^i)$ represents the emission probabilities.

Past works have shown that the prediction problem in HMMs can be viewed as a

Algorithm 1: Structured Perceptron

Input : Number of iteration N, Training Data $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l}$
1 **Output**: $\mathbf{w}_{avg}/(Nl)$
begin
2 $\quad \mathbf{w} \leftarrow 0, \mathbf{w}_{avg} \leftarrow 0$
3 $\quad$ **for** $t = 1 \dots N$ **do**
$\quad\quad$ **for** $i = 1 \dots l$ **do**
$\quad\quad\quad \hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_i)} \mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y})$.
$\quad\quad\quad \mathbf{w} \leftarrow \mathbf{w} + \Phi_{\mathbf{y}_i, \hat{\mathbf{y}}(\mathbf{x}_i)}$
$\quad\quad\quad \mathbf{w}_{avg} \leftarrow \mathbf{w}_{avg} + \mathbf{w}$

linear model over "local" features [81, 82]. That is, one can show that

$$\arg\max_{y} P(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{y}} \log P(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{y}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}), \tag{2.2}$$

where $\mathbf{w}$ is a weight vector and Φ represents a feature function. Therefore, we can convert the probability tables of an HMM into a linear function represented by $\mathbf{w}$ with appropriate feature functions. In this representation, the feature function $\Phi(\mathbf{x}, \mathbf{y})$ is expressed as a set of features which contain "prior features", $\Phi_p(y^1)$, "transition features" , $\Phi_t(y^i, y^{i-1})$, and "emission features", $\Phi_e(x^i, y^i)$ [81]. In other words, there exists a one-to-one mapping between the active features and the associated probability representation, which can be rewritten in the form of a linear function.

2.5.2 Structured Perceptron

The structured perceptron (SP) was first introduced by [82]. The algorithm (Algorithm 1) extends the mistake-driven idea of the Perceptron algorithm (mentioned in Algorithm 2) to the structured output case. In line 3, it finds the best structure for an example using the current weight vector. Then the weight vector is updated with the difference between the feature vectors of the true label and the prediction. Notice that this

Algorithm 2: The Perceptron Learning Algorithm. Note that the feature function only depend on the input here.

Input : Learning rate η, Number of iteration N, Training Data $\mathcal{B} = \{(\mathbf{x}_i, z_i)\}_{i=1}^{l}$
Output: $\mathbf{w}$
begin
1 $\quad \mathbf{w} \leftarrow 0$
2 $\quad$ **for** $t = 1 \dots N$ **do**
$\quad\quad$ **for** $i = 1 \dots l$ **do**
$\quad\quad\quad$ **if** $z_i \mathbf{w}^T \Phi(\mathbf{x}_i) \leq 0$ **then**
$\quad\quad\quad\quad \mathbf{w} \leftarrow \mathbf{w} + z_i \Phi(\mathbf{x}_i)$

is a mistake-driven algorithm, which means that if the current weight vector successfully finds the correct output, the weights will not change. Inspired by the results of [83], the algorithm maintains an averaged weight vector (line 3), which is the final output. This technique has been shown to improve the generalization ability of the final model[83]. However, while structured perceptron algorithm is simple and easy-to-implement, it does not capture the concept of margin and there is no easy method to select N, the number of iterations. In structured perceptron, the prediction function is Eq. (2.3).

$$\arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_i)} \mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y}). \tag{2.3}$$

2.5.3 Conditional Random Field Models

Conditional Random Field (CRF) [84] can be viewed as a probabilistic discriminative model. CRF models the conditional probability by:

$$P(\mathbf{y}_i | \mathbf{x}_i, \mathbf{w}) = \frac{\exp^{\mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y}_i)}}{\sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_i)} \exp^{\mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y})}}. \tag{2.4}$$

The training of a CRF is done by maximizing the conditional log likelihood of the labeled examples. The objective function of a CRF can be written as follows:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^{l} \log \frac{\exp^{\mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y}_i)}}{\sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_i)} \exp^{\mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y})}}. \tag{2.5}$$

The denominator of (2.4) is a summation over all possible structures for this example. Often this is the main computation bottleneck for training a CRF model, given that it contains exponential number of structures. Fortunately, this function can be calculated efficiently if we introduce some restrictions on $\mathcal{Y}$ and Φ. For example, in order to calculate the gradient of the weight vector $\mathbf{w}$, one needs to calculate the term

$$E_{P(\mathbf{y}_i|\mathbf{x}_i,\mathbf{w})}[\Phi(\mathbf{x}_i, \mathbf{y}_i)].$$

If we adopt the Markov assumption as in a HMM model, this term can be computed by the standard forward-backward procedure [85]. However, such restrictions often make it impossible for CRF to capture long distance relationships. Several works have tried to improve the CRF models by capturing the long distance relationships in the supervised setting *at test time* [3, 86]. In a CRF, the prediction function can be expressed as Eq. (2.3) by rewriting the prediction function.

2.6 Constrained Conditional Model

CCMs target structured prediction problems, where given a point $\mathbf{x}$ in an input space $\mathcal{X}$, the goal is to find a label assignment $\mathbf{y}$ in the set of all possible output structures for $\mathbf{x}$, $\mathcal{Y}(\mathbf{x})$. For example, in part-of-speech (POS) tagging, $\mathcal{Y}(\mathbf{x})$ is the set of all possible POS tags for a given input sentence $\mathbf{x}$.

Given a set of feature functions $\Phi = \{\phi_i(\cdot)\}_{i=1}^{n}$, $\phi_i : \mathcal{X} \times \mathcal{Y} \rightarrow R$, which typically encode the *local* properties of a pair $(\mathbf{x}, \mathbf{y})$ (often, the image of ϕ_i is $\{0, 1\}$), the "score"

of a structure $\mathbf{y}$ of a linear model can be represented as

$$f(\mathbf{x}, \mathbf{y}) = \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} w_i \phi_i(\mathbf{x}, \mathbf{y}).$$

The prediction function of this linear model is $\arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} f(\mathbf{x}, \mathbf{y})$.

Constrained Conditional Models provide a general **interface** that allows users to easily combine domain knowledge (which is provided by humans) and statistical models (which are learned from the data). In this chapter, we represent domain knowledge as a (usually small) set of constraints $\Psi = \{\Psi_k\}_{k=1}^{m}$. For each constraint, we are also provided a function $d_{\Psi_k} : \mathcal{X} \times \mathcal{Y} \rightarrow R$ that measures the degree to which the constraint Ψ_k is violated in a pair $(\mathbf{x}, \mathbf{y})$.

A **Constrained Conditional Model** can be represented using two weight vectors: the feature weight vector $\mathbf{w}$ and the constraint penalty vector ρ. The score of an assignment $\mathbf{y} \in \mathcal{Y}$ for an instance $\mathbf{x} \in \mathcal{X}$ can then be obtained by[1]

$$f_{\Phi,\Psi}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} w_i \phi_i(\mathbf{x}, \mathbf{y}) - \sum_{k=1}^{m} \rho_k d_{\Psi_k}(\mathbf{x}, \mathbf{y}). \tag{2.6}$$

A CCM then selects the best structure using the inference problem

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} f_{\Phi,\Psi}(\mathbf{x}, \mathbf{y}), \tag{2.7}$$

as its prediction.

Note that a CCM is not restricted to be trained with any particular learning algorithm. The key goal of a CCM is to allow combining constraints and models in the testing phase. Similarly to other linear models, specialized algorithms may need to be developed to train CCMs. Notice also that the left component in Eq. (2.6) may stand for multiple linear

[1]Recall that n is the number of features and is typically very large, and m is the number of constraints, typically small.

models, trained separately. Unlike standard linear models, we assume the availability of some prior knowledge, encoded in the form of *constraints*. When there is no prior knowledge, there is no difference between CCMs and other linear models.

2.6.1 Benefits of Separating Constraints and Features

In Eq. (2.6), the constraints term (the second term) appears to be similar to the features term (the first time). In fact, using constraints or features to express long distance relationships sometimes can be a design choice. However, it is important to note that both in this work and in many other recent publications [2, 3, 87, 88, 89, 90, 91], people have demonstrated the importance of separating features and constraints. In this section we discuss this issue in details.

- Hard constraints vs. features

  While we simplified our notation in Eq. (2.6), the constraint term is different from the feature term because it can be used to enforce hard constraints. Hence, it is necessary to separate constraints and features.

- Reuse and improving existing models with expressive constraints

  It is often expensive to retrain a complicated NLP system. While sometime choosing features or constraints to express long distance relationships can be a design choice, adding more features often require expensive retraining. Moreover, in [92], they propose use constraints to *combine* two independent trained models. Note that if we model the long distance constraints with features, we need to train these two models *jointly*, which can be much more expensive compared to training them separately by separating constraints from the features.

- Implications on learning algorithms

Separating expressive constraints from models also impacts the learning performance. Many recent works have show many benefits of keeping the existing model and treating the expressive constraints as a supervision resource [93, 88, 89, 90, 91]. As we show in this work, using constraints as a supervision resource can be very effective when there are few labeled examples in the semi-supervised setting.

In the supervised setting, we separate the constraints from features in Eq. (2.6) because the constraints should be trusted most of the time. Therefore, the penalties ρ can be fixed or handled separately. For example, if we are confident about our knowledge, rather than learning the $\{\rho_j\}$, we can directly set them to ∞, thus enforcing the chosen assignment $\mathbf{y}$ to satisfy the constraints. There issues are discussed in details later in the chapter.

- Efficiency

  Another difference between ρ and $\mathbf{w}$ is that ρ should always be positive. The reason is that $d_{\Psi_i}(\mathbf{x}, \mathbf{y}) \geq 0$ and the assignments that violate the constraints should be punished (See Eq. (2.6)). This allows us to design an admissible heuristic and speed up exact inference using A^* search. We cannot have this nice result when we treat constraints as features. This is of particular importance, since the constraints could be non-local, therefore efficient dynamic programming algorithms are not applicable.

There are several advantages of using constraints. First, constraints provide a platform for encoding prior knowledge, possibly expressed as high level predicates. As we will show later, this is especially important when the number of labeled instances is small. Second, constraints can be more expressive than features used by the *existing* model so adding constraints can sometimes prevent us to redesign the model. Instead of building a model from complex features, CCMs provide a way to combine "simple" learned models with a small set of "expressive" constraints. Importantly, combining

simple models with constraints often results in better performance. For example, the top-ranking system in the CoNLL 2005 shared task uses a CCM approach and outperforms many systems built using complex models [94].

2.6.2 Inference with Constraints

Adding expressive constraints comes with a price – the dynamic programming inference algorithms typically used in off-the-shelf statistical models can no longer be applied. In this section, we discuss three different types of inference algorithms that allow solving the inference problem in Eq. (2.7) with expressive constraints.

Integer Linear Programming

In the earlier related works that made use of constraints, the constraints were assumed to be binary functions; in most cases, a high level (first order logic) description of the constraints was compiled into a set of linear inequalities, and exact inference was done using an integer linear programming formulation (ILP) [92, 95, 96, 94, 97, 98]. Although ILP can be intractable for very large-scale problems, it has been shown to be quite successful in practice when applied to many practical NLP tasks [95].

A^* Search

Approximated Search

While the A^* algorithm is technically sound, in this chapter, we use beam search to approximate the solution for the inference problem in Eq. (2.7). The advantage of using this procedure is that the memory usage of beam search is fixed while the memory usage of the A^* algorithm can be potentially big. We found that the approximated inference procedure performs very well in our experiments. The comparison of the

three proposed inference algorithms on other domains is an interesting issue to address in future research.

2.7 Work in BioNLP Domain

BioNER Here, the goal is to identify, within a collection of text, all of the instances of a name for a specific type of thing: for example, all of the drug names within a collection of journal articles, or all of the gene names and symbols within a collection of MEDLINE abstracts. Recognizing biological entities in text allows for further extraction of relationships and other information by identifying the key concepts of interest and allowing those concepts to be represented in some consistent, normalized form.

This task has been challenging for several reasons. First, there does not exist a complete dictionary for most types of biological named entities, so simple text-matching algorithms do not suffice. In addition, the same word or phrase can refer to a different thing depending upon context (eg ferritin can be a biological substance or a laboratory test). Conversely, many biological entities have several names (eg PTEN and MMAC1 refer the same gene). Biological entities may also have multi-word names (eg carotid artery), so the problem is additionally complicated by the need to determine name boundaries and resolve overlap of candidate names. Because of the potential utility and complexity of the problem, NER has attracted the interest of many researchers, and there is a tremendous amount of published research in this topic.

The approaches generally fall into three categories: lexicon-based, rules-based and statistically based. One of the most successful rules-based approaches to gene and protein NER in biomedical texts has been the AbGene system of Tanabe and Wilbur [99]. It has been used as the NER component in extracting relationships by several other re-

searchers [100, 101]. AbGene works by extending the Brill POS tagger [102] to include gene and protein names as a tag type with the system trained on 7,000 hand-tagged sentences from biomedical text. AbGene then applies manually generated post-processing rules based on lexical-statistical characteristics that help further identify the context in which gene names are used and eliminate false positives and negatives. The system achieved a precision of 85.7 per cent at a recall of 66.7 per cent.

In contrast to the tagging approach used by Tanabe and Wilbur, Chang et al. created the GAPSCORE system [103], which assigns a numerical score to each word within a sentence by examining the appearance, morphology and context of the word and then applying a classifier trained on these features. Words with higher scores are more likely to be gene and protein names or symbols.

A number of other groups have worked in this area. Hanisch et al. [104] used a large dictionary of gene and protein names and semantically classified words that tend to appear in context with protein names, reporting a specificity of 95 per cent and sensitivity of 90 per cent. Zhou et al. [105] trained a hidden Markov model (HMM) on a set of features based on word formation (ie capitalization), morphology (ie prefix and suffix), POS, semantic triggers (head nouns and verbs) and intra-document name aliases. They reported an overall precision of 66.5 per cent at a recall of 66.6 per cent on the GENIA corpus [106]. Other gene and protein NER systems include those by Narayanaswamy et al. [107], and Mika and Rost [108].

Event Recognition BioNLP 2009 shared task [109] concerns the detailed behavior of bio-molecules, characterized as bio-molecular events (bio-events). The difference in focus is motivated in part by different applications envisioned as being supported by the IE methods. For example, BioCreative aims to support curation of PPI databases, for a

long time one of the primary tasks of bioinformatics. The BioNLP task aims to support the development of more detailed and structured databases which are gaining increasing interest in bioinformatics research in response to recent advances in molecular biology.

The event types addressed in the BioNLP task were selected from the GENIA ontology, with consideration given to their importance and the number of annotated instances in the GENIA corpus. The selected event types all concern protein biology, implying that they take proteins as their theme. The first three types concern protein metabolism, i.e. protein production and breakdown. Phosphorylation is a representative protein modification event, and Localization and Binding are representative fundamental molecular events. Regulation (including its sub-types, Positive and Negative regulation) represents regulatory events and causal relations. The last five are universal but frequently occur on proteins.

Bjorne et al.'s [110] system achieved the best results in this task. It is characterized by heavy reliance on efficient, state-of-the-art machine learning techniques and a wide array of features derived from a full dependency analysis of each sentence. The system is a pipeline of three major processing steps: trigger recognition, argument detection and semantic post-processing. By separating trigger recognition from argument detection, authors use methods familiar from named entity recognition to tag words as event triggers. Event argument detection then becomes the task of predicting for each triggerâĂŞtrigger or triggerâĂŞnamed entity pair whether it corresponds to an actual instantiation of an event argument. Both steps can thus be approached as classification tasks. In contrast, semantic post-processing is rule-based, directly implementing argument type constraints following from the definition of the task.

On the other hand, Riedel et al. [111] do not build a pipelined system that first pre-

dicts event clues and cellular locations, and then relations between these; instead, they design and learn a joint discriminative model of the complete event structure for a given sentence. This allows them to incorporate global correlations between decisions in a principled fashion. Moreover, instead of designing and implementing specific inference and training methods for structured model, authors use Markov Logic and define our global model declaratively.

Chapter 3

Learning from Negative Examples in Set-Expansion

3.1 Introduction

In this chapter, we address the task of *set-expansion*. Set-expansion has been viewed as a problem where a few examples of the desired concept are given as input and the goal is to output an extensive list of instances of the desired concept. For example, if the seed-set is {*Steffi Graf*, *Martina Hingis*, *Serena Williams*}, the system should output an extensive list of female tennis players.

Set-expansion is an important application of its own. It can also be used to facilitate several other Information Extraction tasks such as *Fine-Grained NER* [112, 113], *Coreference* [114, 37] etc.

The task of *set-expansion* has been addressed in several works which would be discussed in more detail in Section 3.2. Existing systems for set-expansion either work on structured or semi-structured or free text or a combination of them. *In this chapter, we focus on set-expansion from free text.* For set-expansion on free text, distributional similarity has been widely used. The state-of-the-art systems [115, 116] use a centroid-based approach wherein they first find the centroid of the entities in the seed-set and then find the entities similar to the centroid.

Most of the work on set-expansion has focussed on taking only positive examples. For example, as discussed above, to produce a list of female tennis players, a few female tennis players are given as input to the system. However, just specifying a few female tennis players doesn't define the concept precisely. The set-expansion systems tend to

output some male tennis players along with female tennis players. Specifying a few male tennis players as negative examples to the system defines the concept more precisely. Table 3.1 compares the state-of-the-art approach for set-expansion on free text with the approach presented in this chapter. The table shows only a small portion of the lists output by the system. We used 7 positive examples for both the approaches. Only 1 negative example was used for the second approach. The errors have been underlined and italicized. We see that the output in 1^{st} column is corrupted by male tennis players. Adding only 1 negative example to the seed-set improves the list-quality significantly. Second column contains no errors. *In this chapter, we propose ways to learn from negative examples in set-expansion and show significant improvement.*

We present an inference-based approach to set-expansion in which we don't compute the centroid. Rather, we work directly with the entities in the seed-set. The new approach developed by us naturally allows for both positive and negative examples in the seed-set.

The centroid-based approach to set-expansion doesn't directly admit the negative examples. We developed a way of incorporating negative examples into the centroid-based approach by learning a weight-vector over the corpus vocabulary using linear programming. We use this extension of centroid-based approach as a baseline system and show in the experiments that the inference-based approach developed by us performs much better than the baseline.

There has been some work regarding the choice of seeds for the set-expansion system [117, 118]. Certain seed-sets give better performance than others. In this chapter, we present an easy solution to this problem where the seeds are dynamically chosen by the user.

To summarize, this chapter makes the following major contributions:

1. Showing the use of negative examples for set-expansion

FEMALE TENNIS PLAYERS	
State-of-the-art	**This Chapter**
Monica Seles	Mary Pierce
Steffi Graf	Monica Seles
Martina Hingis	Martina Hingis
Mary Pierce	Lindsay Davenport
Lindsay Davenport	Steffi Graf
Jennifer Capriati	Jennifer Capriati
Kim Clijsters	Kim Clijsters
Mary Joe Fernandez	Karina Habsudova
Nathalie Tauziat	Sandrine Testud
Kimiko Date	Kimiko Date
Conchita Martinez	Chanda Rubin
Anke Huber	Anke Huber
Judith Wiesner	Nathalie Tauziat
Andre Agassi	Jana Novotna
Pete Sampras	Conchita Martinez
Jana Novotna	Nathalie Dechy
Karina Habsudova	Amanda Coetzer
Jim Courier	Barbara Paulus
Justine Henin	Arantxa Sanchez-Vicario
Julie Halard	Amy Frazier
Meredith McGrath	Iva Majoli
Goran Ivanisevic	Magdalena Maleeva
Jelena Dokic	Jelena Dokic
Michael Chang	Julie Halard

Table 3.1: This table compares the state-of-the-art approach for set-expansion on free text with the approach presented in this chapter. The bold and italicized entries correspond to male tennis players and are erroneous. Addition of only 1 negative example to the seed-set improves the list-quality significantly. Second column contains no errors.

2. Extending the state-of-the-art approach to learn from negative examples

3. Presenting an inference-based approach to set-expansion

4. Developing an active learning based strategy to find good seed sets

The rest of the chapter is organized as follows: Section 3.2 describes the related work. Preliminaries to set-expansion algorithms is presented in Section 3.3. The centroid-based approach to set-expansion is discussed in Section 3.4. Section 3.5 presents a novel way of

incorporating negative examples in the centroid-based approach. Section 3.6 describes the inference-based approach to set-expansion. The issue of selection of good seeds is discussed in Section 3.7. Section 7.10 and Section 3.9 describe the dataset used for the experiments and the results of the experiments respectively. Finally, we conclude in Section 3.10.

3.2 Related Work

The task of *set-expansion* has been addressed in several works. We report here the most significant efforts towards this task.

3.2.1 Web-based Set-Expansion systems

Google™ has a proprietory system, Google Sets[1], for set-expansion. Another system for *set-expansion* is Boowa[2] [119, 118, 120]. Boowa works by finding semi-structured web pages that contain "lists" of items, and then aggregating these "lists" so that the most promising items are ranked higher. The KnowItAll system of Etzioni et al. [121] depends on the output of existing search engines to extract collections of facts from the Web. Etzioni et al. [121] use Pattern Learning, Subclass Extraction and List Extraction to improve KnowItAll's recall.

3.2.2 Set-Expansion systems for free text

For set-expansion on free-text, pattern recognition and distributional similarity have primarily been used.

Riloff and Jones [122] used a two-level bootstrapping mechanism based on pattern recognition for set-expansion. In each iteration, they add 5 new members to the list.

[1] `http://labs.google.com/sets`
[2] `http://www.boowa.com/`

Since they need to make one pass over the entire corpus for every iteration, their method is quite inefficient. Moreover, their algorithm is very sensitive to the erroneous members which may get added to the list during the expansion.

Talukdar et al. [123] present a context pattern induction method for named-entity extraction. Their method automatically selects trigger words to mark the beginning of a pattern, which is then used for bootstrapping from free text. However, they focussed on very broad entity types like Location, Person and Organization whereas we are interested in finer concepts like Athletes, Actors etc. Moreover, they used hundreds of seeds for constructing the semantic lexicons. On the other hand, we give a much smaller number of seeds.

Sarmento et al. [115] present a corpus-based approach to set-expansion. For a given set of seed entities they use co-occurrence statistics taken from a text collection to define a membership function that is used to rank candidate entities for inclusion in the set. They represent entities as vectors and essentially construct a centroid of the seed-set.

Pantel et al. [116] developed a parallel implementation for computing the pairwise semantic similarity between the entities. They applied the learned similarity matrix to the task of set-expansion using the centroid-based algorithm developed by Sarmento et al. [115]. They present a large empirical study to quantify the effect of corpus size, corpus quality, seed composition and seed size on set-expansion performance.

3.2.3 Set-Expansion systems using Integrated approaches

Talukdar et al. [124] present a graph-based semi-supervised label propagation algorithm for acquiring open domain labeled classes and their instances from a combination of unstructured and structured text sources. Pennacchiotti and Pantel [125] present a framework called Ensemble Semantics for modeling information extraction algorithms that combine multiple sources of information and multiple extractors. Pasca and Van

Durme [126] present an approach to information extraction that exploits both Web documents and query logs to acquire open-domain classes of instances, along with relevant sets of open-domain class attributes.

Ghahramani and Heller [127] illustrates a Bayesian Sets algorithm that solves a particular sub-problem of set-expansion, in which candidate sets are given, rather than a corpus of documents.

3.2.4 Use of Negative Examples in Set-Expansion

Thelen and Riloff [128] and Lin et al. [129] present a framework to learn several semantic classes simultaneously. In this framework, the instances which have been accepted by any one semantic class serve as negative examples for all other semantic classes. This approach is limited because it necessitates the learning of several semantic classes simultaneously. Moreover, negative examples are NOT useful if the different semantic classes are not related to one another. Winston et al. note that it is not easy to acquire good negative examples. The approach presented by us allows the use of negative examples even when there is only one semantic class. Also, we present a strategy to easily acquire good negative examples.

In this chapter, we focus on set-expansion from free text. So, we don't compare our system with the systems which use textual sources other than free text (e.g. semi-structured web pages or query logs). The works of Sarmento et al. [115] and Pantel et al. [116] are the state-of-the-art works that come closest to our approach. In our experiments, we compare the centroid-based approach employed by them with the approach developed by us.

3.3 Preliminaries

In Sections 3.4, 3.5 and 3.6, we would present three algorithms for set-expansion:

- Centroid-based approach
- Adding negative examples to centroid-based approach
- Inference-based approach

All these algorithms use a vector representation for the entities. In this section, we would describe how to generate feature vectors for the entities and the similarity metric that we used.

3.3.1 Feature Vector Generation

Preprocessing: The input to our set-expansion system consists of free text. To extract the relevant entities from free text, we preprocess the corpus with a state-of-the-art Named Entity Recognition tool developed by Ratinov and Roth[3] [130].

Next, we explain the process of feature generation for the entities. The features of an entity are based on the words surrounding the entity. For our vocabulary of the corpus, we take all the distinct tokens appearing in the corpus except the punctuation symbols, stopwords and some other very high frequency words. The resulting vocabulary is denoted by V. $v_i = V[i]$ represents the i^{th} word in the vocabulary. $VFreq[i]$ denotes the frequency of occurrence of word v_i in the corpus.

We denote the set of all the distinct entities appearing in the corpus by E. $e_i = E[i]$ represents the i^{th} entity. $EFreq[i]$ denotes the frequency of occurrence of entity e_i in the corpus. The vocabulary words appearing in a window of size W centered on each mention of entity e_i contribute towards the features of the entity e_i. We maintain a feature vector FV_i for every entity such that $FV_i[j]$ gives the frequency with which the vocabulary word v_j occurs as a feature of entity e_i.

[3] `http://cogcomp.cs.illinois.edu/page/software`

Entity	Sample of Feature Vector
Bill Clinton	[President, 24912], [administration, 790], [House, 766], [visit, 761], [talks, 742], [announced, 737], [summit, 703], [White, 684], [Republican, 541], [WASHINGTON, 508], [Congress, 490], [Democratic, 318], [budget, 243], [veto, 230], [government, 219], [election, 192], [political, 182], [Hillary, 149]
Pete Sampras	[USA, 323], [World, 254], [champion, 226], [number-one, 191], [defending, 124], [final, 115], [American, 112], [pts, 99], [beat, 86], [round, 81], [tennis, 73], [singles, 65], [Wimbledon, 62], [seeded, 40], [lost, 39], [semi-final, 38], [Grand, 36], [Slam, 36], [tournament, 34], [top-seed, 32], [Tennis, 5]
Tom Cruise	[actor, 21], [film, 21], [starring, 20], [Impossible, 18], [John, 17], [Travolta, 16], [Nicole, 15], [Kidman, 15], [Mission, 14], [Hollywood, 10], [co-producer, 7], [million, 6], [celebrities, 6], [leading, 6], [Scientology, 6], [superstar, 5], [screen, 4], [role, 4], [cinemas, 4], [thriller, 4], [actor-producer, 3], [matinee, 1]
Zinedine Zidane	[French, 99], [midfielder, 66], [Real, 52], [Madrid, 44], [player, 29], [injury, 28], [international, 23], [Cup, 22], [World, 21], [goal, 13], [thigh, 11], [match, 11], [ball, 8], [coach, 8], [striker, 8], [scored, 8], [win, 6], [record, 6], [time, 6], [footballer, 6], [Ronaldo, 2], [footballing, 2]

Table 3.2: *Examples of Features*: This table shows some of the features for four different entities. We see that features are quite good in representing the entities. The numbers along with the features tell the absolute frequency of the corresponding feature appearing with the entity under consideration.

We use *pointwise mutual information* (PMI) [131] to measure the association of a feature with the entity. We denote the PMI between entity e_i and word v_j by pmi_{ij}. pmi_{ij} is computed as follows:

$$pmi_{ij} = \frac{\frac{FV_i[j]}{N}}{\frac{EFreq[i]}{N}\frac{VFreq[j]}{N}} \tag{3.1}$$

where N is the total number of words in the corpus. PMI has been shown to be biased towards infrequent entities/features. Following Pantel and Lin [132], we multiplied the PMI value with the following discouting factor DF_{ij}:

$$DF_{ij} = \frac{FV_i[j]}{FV_i[j]+1}\frac{\min(EFreq[i], VFreq[j])}{min(EFreq[i], VFreq[j])+1} \tag{3.2}$$

Multiplication with the discounting factor gives us the discounted PMI which we denote by $dpmi_{ij}$.

$$dpmi_{ij} = pmi_{ij} DF_{ij} \tag{3.3}$$

Our feature vectors are composed of *dpmi* values of the features as given by Equation (3.3).

Examples of Features: Table 3.2 gives some of the features for four different entities as generated from the corpus. The numbers along with the features tell the absolute frequency of the feature. We see that the feature vectors represent the entities quite well. For example, Pete Sampras is a tennis player and the features indicate that he is from "USA", he has been a "top-seed", he has participated in "Wimbledon" etc.

It is to be noted that we did not convert the features to lower-case as a normalization step. We retained the capitalization of the features because it provides useful information. For example, the feature "House" in the case of Bill Clinton is referring to *White House* and is different from an ordinary "house". Also, we see that a lot of features are coming from proper-nouns and give useful information about the entity under consideration.

3.3.2 List Generation

We compute the similarity, sim_{ij}, between the entities e_i and e_j using the cosine coefficient [133]:

$$sim_{ij} = \frac{\sum_k dpmi_{ik} dpmi_{jk}}{\sqrt{\sum_k dpmi_{ik}^2} \sqrt{\sum_k dpmi_{jk}^2}} \tag{3.4}$$

In the above equation, *dpmi* values are obtained from Equation (3.3).

Given an entity e_i, we can find the similarity between e_i and all the entities in the entity

set E using Equation (3.4). Then we can sort all the entities in E based on this similarity score in the decreasing order. The resulting ranked list has the property that the entities with lower rank are more similar to e_i than the entities with higher rank. In the rest of the chapter, we would call such a list as *NBRLIST* of e_i. We take the letters *NBR* from the word *NeighBouR*.

3.4 Centroid-Based Approach to Set-Expansion

For doing set-expansion from free text, existing state-of-the-art systems [115, 116] primarily employ a centroid-based approach. We would denote the *Centroid* by $\mathcal{C}$ and the seed-set by $\mathcal{S}$. In a centroid-based approach, first of all the centroid of the seed-set is computed. The first step in computing the centroid is to find an average of the frequency vectors of the entities in the seed-set. The following equation gives the frequency of the vocabulary word v_j associated with the centroid:

$$FV_{\mathcal{C}}[j] = \frac{\sum_{e_i \in \mathcal{S}} FV_i[j]}{|\mathcal{S}|} \tag{3.5}$$

where $FV_i[j]$ gives the frequency of vocabulary word v_j associated with entity e_i. Then the discounted PMI of the resulting frequency vector is computed as was described in Section 3.3.1. After finding the centroid, *NBRLIST* of centroid is computed as described in Section 3.3.2. Finally, the first M members of the *NBRLIST* are output to the user where M is the cut-off.

3.5 Learning from Negative Examples in Centroid-Based Approach

Centroid-based approach to set-expansion doesn't easily allow learning from negative examples. In this section, we present a novel framework which allows the incorporation

of negative examples in a centroid-based approach.

In Section 3.4, we described how to compute the centroid of the seed-set. The active features of any entity are those features which have non-zero value. The active features of the centroid are the union of the active features of the entities in the seed-set. All the active features of the centroid are not equally important. To incorporate this knowledge into set-expansion, we associate a weight term with each entry in the vocabulary. Higher weight would mean that a particular word is more relevant to the underlying concept. By incorporating these weights into the similarity formula given by Equation (3.4), the new similarity formula becomes:

$$wsim_{ij} = \frac{\sum_k w_k dpmi_{ik} dpmi_{jk}}{\sqrt{\sum_k dpmi_{ik}^2}\sqrt{\sum_k dpmi_{jk}^2}} \tag{3.6}$$

where w_k is the weight associated with the word v_k. We wish to learn the weight vector w such that the similarity between the positive examples and the centroid becomes more than a prespecified threshold $\asymp$. Also, the similarity between negative examples and the centroid should become less than a prespecified threshold $\downarrow$.

We accomplish this objective using the following linear program:

$$max \sum_{e_i \in PositiveExamples} wsim_{\mathcal{C}\,i} - \sum_{e_j \in NegativeExamples} wsim_{\mathcal{C}\,j} \tag{3.7}$$

$$\text{s.t. } \sum_k w_k \leq \text{num of non-zero entries in centroid} \tag{3.8}$$

$$wsim_{\mathcal{C}\,i} \geq \asymp \quad \forall e_i \in PositiveExamples \tag{3.9}$$

$$wsim_{\mathcal{C}\,i} \leq \downarrow \quad \forall e_i \in NegativeExamples \tag{3.10}$$

$$w_k \geq 0 \qquad \forall k \tag{3.11}$$

$$w_k \leq \geq \qquad \forall k \tag{3.12}$$

In the above linear program, Equation (4.1) is the objective of the linear program which aims at

1. maximizing the similarity between positive examples and the centroid and
2. minimizing the similarity between negative examples and the centroid.

Note that $\mathcal{C}$ refers to centroid in Equations (4.1), (3.9) and (3.10). Equation (3.8) specifies that the sum of all the weights should not be larger than the number of non-zero entries in the centroid. Equations (3.9) and (3.10) specify the thresholds $\asymp$ and $\downarrow$ for the positive and negative examples respectively. Equation (3.11) specifies the non-negativity constraints on the weight vector. Equation (3.12) specifies the upper bound on the individual elements of the weight vector.

In our experiments, we set $\asymp$, $\downarrow$ and $\geq$ to 0.2, 0.0001 and 10 respectively. We used the Gurobi Optimization Toolkit[4] v4.5 for solving the above linear program. For the concepts that we experimented with, the linear program involved less than 5000 variables. Using

[4] `http://www.gurobi.com/`

the Gurobi Toolkit, we were able to find the optimal solution to the above linear program within three seconds.

3.6 Inference-Based Approach to Set-Expansion

Centroid-based approach to set-expansion has some limitations. In the centroid-based approach, centroid is supposed to fully represent the underlying concept. All the similarity scores are computed with respect to the centroid. There is no way to confirm the decisions made with respect to the centroid. There is a lot of information in the individual positive and negative examples which is not exploited in the centroid-based approach. Moreover, as more and more positive examples are added to the seed-set, the number of active features of the centroid keep on increasing. It is quite possible that it may lead to over-generalization.

In this section, we present an inference-based method for set-expansion. Unlike Section 3.4, we do not compute the centroid of the positive examples. The new approach is based on the intuition that the positive and negative examples can complement each others' decision to better represent the underlying concept. Each example can be thought of as an expert which provides positive or negative evidence regarding the membership of any entity to the underlying concept. We develop a mechanism to combine the decisions of such experts.

Algorithm 3 gives the procedure for inference-based set expansion. In steps 1 and 2, we find out the *NBRLIST* of positive and negative examples respectively. *NBRLIST* of an entity is defined in Section 3.3.2. The entities which have high similarity to the positive examples are more likely to belong to the underlying concept. Similarly, the entities which have high similarity to the negative examples are likely to NOT belong to the underlying concept.

Steps 1 and 2 of Algorithm 3 give us one list corresponding to every positive and neg-

Algorithm 3: InferenceBasedSetExpansion

Input : E (Entity Set), W (List of positive examples), B (List of negative examples)
Output: L (Ranked List)
begin

1 *Compute NBRLISTs of Positive Examples*
for $j \leftarrow 1$ **to** $|W|$ **do**
  for $i \leftarrow 1$ **to** $|E|$ **do**
    $WSV_j[i] \longleftarrow sim_{iw_j}$
  Sort the entities in E based on WSV_j and store the result in WL_j

2 *Compute NBRLISTs of Negative Examples*
for $j \leftarrow 1$ **to** $|B|$ **do**
  for $i \leftarrow 1$ **to** $|E|$ **do**
    $BSV_j[i] \longleftarrow sim_{ib_j}$
  Sort the entities in E based on BSV_j and store the result in BL_j

3 *Initialize the score for each entity to* 0
for $i \leftarrow 1$ **to** $|E|$ **do**
  $score_i \longleftarrow 0$

4 *Compute the contribution from positive examples*
for $j \leftarrow 1$ **to** $|WL|$ **do**
  for $i \leftarrow 1$ **to** $|E|$ **do**
    $e \longleftarrow WL_j[i]$
    $score_e \longleftarrow score_e + reward(i, 0) + WSV_j[e]$

5 *Compute the contribution from negative examples*
for $j \leftarrow 1$ **to** $|BL|$ **do**
  for $i \leftarrow 1$ **to** $|E|$ **do**
    $e \longleftarrow BL_j[i]$
    $score_e \longleftarrow score_e + reward(i, 1)$

6 $L \longleftarrow$ List of entities in E sorted by *score*

ative example. We associate a reward (or penalty) with each entity in these lists based on the rank of the entity. Our reward (or penalty) function is based on the effective length, $\mathcal{L}$, of a list. The entities which have higher rank than the effective length of the list are given a reward (or penalty) of zero. Effective length, $\mathcal{L}$, of a list is computed by multiplying the required list length (or cut-off) by a list factor, $\mathcal{F}$. If M is the specified cut-off, then $\mathcal{L} = M \times \mathcal{F}$. The reward is calculated according to the following equation:

$$reward(r,n) = \begin{cases} (-1)^n \times \mathcal{L} \times a & \text{if } r = 1 \\ (-1)^n \times (\mathcal{L} - r) \times b & \text{if } 1 < r \leq \mathcal{L} \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

In the above equation, r refers to the rank of the entity. n is set to 0 for lists corresponding to positive examples and n is set to 1 for lists corresponding to negative examples. Thus, for lists corresponding to negative examples, the reward is negative and hence, acts like a penalty. The values a and b were determined empirically and set to 100 and 10 respectively. Equation (3.13) gives higher reward or penalty to the entities with lower rank.

Figure 3.1 shows the effect of $\mathcal{F}$ on the Mean Average Precision (MAP) (please see Section 3.9 for a discussion on MAP) for the concept of female tennis players as the number of seeds is varied. Only positive examples were used for generating Figure 3.1. To find the best value of $\mathcal{F}$, we take the average of MAP across different number of seeds. We find that $\mathcal{F} = 2$ has the highest average of 67.7. Although $\mathcal{F} = 1$ gives good performance for higher number of seeds, its performance is quite low when the number of seeds is small. As we increase the value of $\mathcal{F}$ beyond 2, the performance goes on decreasing. We used $\mathcal{F} = 2$ for all our experiments.

Steps 3, 4 and 5 in Algorithm 3 compute the *score* for each entity in the entity-set E. Step 3 initializes the *score* for each entity to 0. Steps 4 and 5 compute the contributions from the lists corresponding to the positive and negative examples respectively towards the score of entities. If $rank(i,j)$ denotes the rank of entity e_i in the list corresponding to example e_j, then the final score of entity e_i can be written as:

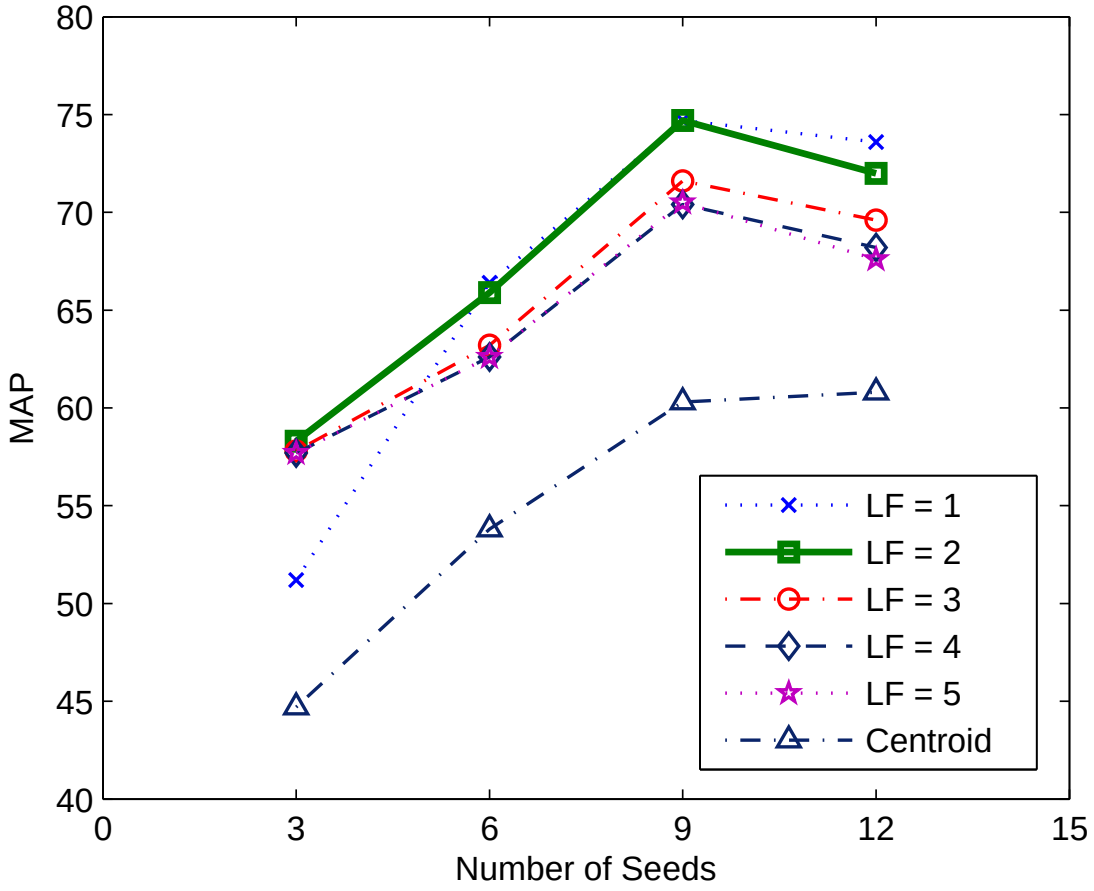


Figure 3.1: This figure shows the effect of list factor ($\mathcal{F}$) on the performance of set-expansion. When averaged across different number of seeds, $\mathcal{F} = 2$ gave the best results.

$$score_i = \sum_{w \in W} [reward(rank(i,w),0) + sim_{iw}] + \sum_{b \in B} [reward(rank(i,b),1)] \tag{3.14}$$

In the above equation, W and B refer to the list of positive and negative examples respectively. The *sim* values are computed using Equation (3.4). Finally, step 6 of Algorithm 3 sorts the entities in descending order based on the final score as computed in Equation (3.14). The first M members of the resulting list are output by the system.

Here, M is the required list length.

3.7 Acquisition of Positive and Negative Examples

In this section, we describe how to get good positive and negative examples for any concept. Generally, in set-expansion, the user specifies some examples belonging to the desired concept and the set-expansion system returns the list of entities belonging to the desired concept. Although the user can easily specify some positive examples based on experience, it would not be feasible for the user to come up with good negative examples for many concepts. Even the positive examples specified by the user may not be able to cover many entities belonging to the desired concept.

Algorithm 4 describes an interactive algorithm to get good positive and negative examples for any concept. The algorithm takes a seed-set as one of its inputs. To begin with, the seed-set can be as small as 1 instance of the desired concept. The algorithm maintains a *WhiteList* and a *BlackList* corresponding to the positive and negative examples respectively. It performs set-expansion using Algorithm 3. The list obtained from Algorithm 3 is presented to the user. The user can specify further positive and negative examples which are added to the *WhiteList* and *BlackList* respectively as long as he/she is not satisfied.

For the positive feedback, the user should select those correct entities which have high rank in the list returned by the system. This is because such entities have low similarity to the entities in the list of positive examples. Therefore, addition of such entities would improve the recall of the system. For the negative feedback, the best results are obtained when the user selects those wrong entities (i.e. errors) which have low rank in the list returned by the system. In Section 3.9, we would present experimental results to show the impact of proper choice of positive and negative examples on the performance of set-expansion.

Algorithm 4: SetExpansionWithUserFeedback

Input : E (Entity Set), S (Seed Set)
Output: L (Ranked List)
begin

```
WhiteList ⟵ ∅
BlackList ⟵ ∅
for s in S do
    WhiteList ⟵ Whitelist ∪ s
L ⟵ InferenceBasedSetExpansion (E, WhiteList, BlackList)
while User is not satisfied with L do
    wl ⟵ Positive Feedback from end of list
    bl ⟵ Negative Feedback from beginning of list
    WhiteList ⟵ WhiteList ∪ wl
    BlackList ⟵ BlackList ∪ bl
    L ⟵ InferenceBasedSetExpansion (E, WhiteList, BlackList)
```

3.8 Datasets

We used English Gigaword Corpus (henceforth referred to as *GCOR*) for our experiments. GCOR was produced by Linguistic Data Consortium (LDC) catalog number LDC2003T05 and ISBN 1-58563-260-0 [134]. This is a comprehensive archive of newswire text data in English that has been acquired over several years by the LDC. Four distinct international sources of English newswire are represented in GCOR: *Agence France Press English Service*, *Associated Press Worldstream English Service*, *The New York Times Newswire Service*, *The Xinhua News Agency English Service*.

In our experiments below, we worked with *Agence France Press English Service* (henceforth referred to as *AFE*) section of *GCOR*. The characteristics of *AFE* are shown in Table 3.3.

We see from Table 3.3 that the total size of the corpus is more than 1 GB. We also see that *AFE* contains a very large number of entities. However, the frequency distribution of the entities is highly non-uniform. A lot of entities occurred only once in AFE and the most frequent entity, *Clinton*, occurred 46039 times.

Attribute	Value
Number of Files	44
Total Size	1,216 MB
Number of Docs	656,269
Total Tokens	170,969,000
Vocabulary Size	548,862
Distinct Entities (PER)	386,209

Table 3.3: Characteristics of AFE section of GCOR

To prepare the gold-sets for the concepts that we experimented with, we used "List of ..." pages and the general content pages of Wikipedia.

3.9 Experiments

3.9.1 Inference vs Centroid Based Approaches

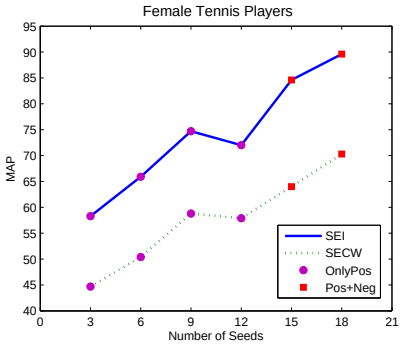
We would use the following notation to refer to the three set-expansion systems that we have presented:

1. *SEC* - Set Expansion system using Centroid. This is the current state-of-the-art [115, 116] and was presented in Section 3.4. This system can't learn from the negative examples.

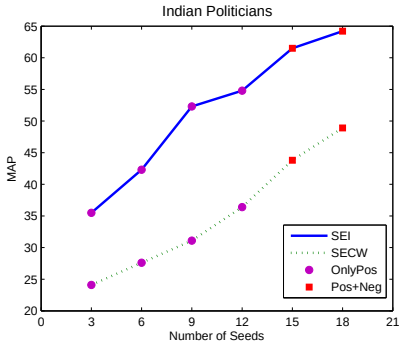
2. *SECW* - Set Expansion system using Centroid where Weights are associated with the vocabulary terms. This system was explained in Section 3.5. This system can learn from negative examples.

3. *SEI* - Set Expansion system using Inference. This is the new approach to set-expansion and it was explained in Section 3.6.

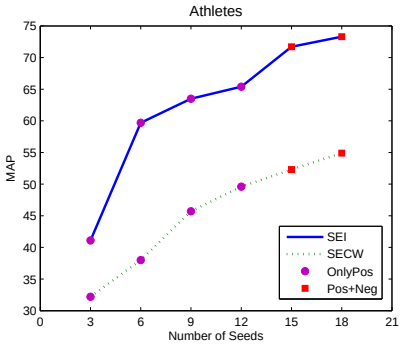
SEC and SECW serve as the baseline systems. Table 3.4 compares the performance of SEI with the two baselines on 5 different concepts as mentioned below:



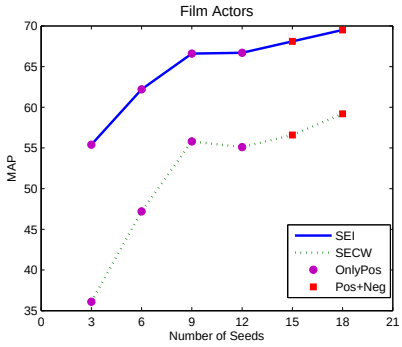
(a) Set-Expansion Results for *FTP*



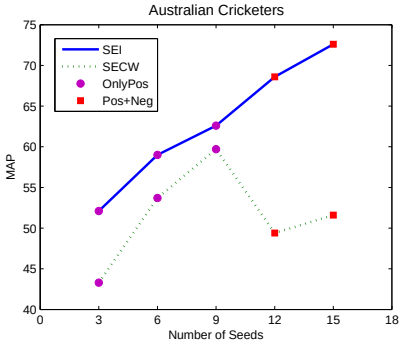
(b) Set-Expansion Results for *IP*



(c) Set-Expansion Results for *ATH*



(d) Set-Expansion Results for *FA*



(e) Set-Expansion Results for *AC*

Figure 3.2: This figure shows the MAP values for 5 different concepts for both SEI and SECW (Baseline). Two things can immediately be noted from the graphs: (1) Negative examples significantly improve the MAP values for both SEI and SECW. (2) SEI performs much better than SECW for all the five concepts.

1. Female Tennis Players (FTP)

2. Indian Politicians (IP)

3. Athletes (ATH)

4. Film Actors (FA)

Concept	SEC	SECW	SEI
FTP	57.9	70.3	**89.6**
IP	36.4	48.9	**64.2**
ATH	49.6	54.9	**73.3**
FA	55.1	59.2	**69.5**
AC	59.7	51.6	**72.6**

Table 3.4: This table compares the MAP of SEI with the 2 baselines on 5 different concepts. Our algorithm, SEI, performs significantly better than both the baselines on all the concepts. SECW is our improvement to the centroid method and is the second best. It performs better than SEC (current state-of-the-art) on all concepts except AC. For details, please refer to Section 3.9.1.

5. Australian Cricketers (AC)

The evaluation metric used in Table 3.4 and in later experiments is Mean Average Precision (MAP). MAP is the mean value of average precisions computed for each ranked list separately. MAP has been commonly used for evaluating ranked lists in the field of Information Retrieval. It contains both recall and precision-oriented aspects, and it is sensitive to the entire ranking. For a list of length $\mathcal{M}$, the Average Precision (henceforth referred to as AP) is given by the following equation:

$$AP = \frac{\sum_{r=1}^{\mathcal{M}}[P(r) \times rel(r)]}{\text{\#TrueEntities}} \tag{3.15}$$

In the above equation, r is the rank, rel is a binary function on the relevance of a given rank and $P(r)$ is the precision at given cutoff rank. To calculate the percentage, we multiply the above value by 100.

For the results presented in Table 3.4, we used 12 positive and 6 negative examples for each concept other than AC. For AC, we used 9 positive and 6 negative examples.

Table 3.4 clearly shows that SEI does much better than both the baselines on all the concepts. We observed that the lists produced by SEI hardly contained any errors in the first half of the lists as we also saw in Table 3.1. This is because the entities which come in the beginning of the list in SEI get high scores from several positive examples and are

NOT penalized by any of the negative examples. On the other hand, SEC and SECW are unable to make such inferences.

We also see from Table 3.4 that except AC, SECW performs better than SEC on all the concepts. The better performance of SECW is because of the use of negative examples. Thus, the strategy developed by us in Section 3.5 for incorporating negative examples is quite effective.

For further analysis, in Figure 3.2, we compare the performance of SEI with SECW on all the concepts as the seed-set size is increased. First we supplied only positive examples as indicated by the *circle* markers in Figure 3.2. After supplying sufficient number of positive examples, we provided negative feedback on 6 examples as indicated by *square* markers in Figure 3.2. For the sake of clarity, we do not show the performance of SEC in Figure 3.2. SEC performs similar to SECW on positive examples but is unable to learn from negative examples.

Two conclusions can readily be drawn from Figure 3.2:

1. Negative examples significantly improve the performance of set-expansion for both SEI and SECW. Only for the concept of *Australian Cricketers*, SECW failed to benefit from negative examples.

2. SEI performs much better than SECW on all the concepts irrespective of the seed-set size.

Table 3.5 categorizes the negative examples that were used for different concepts. We see that the good negative examples are closely related to the true instances of the desired concept. For example, the negative examples for the concept *Australian Cricketers* consist of the cricket players from other countries. We can see from Figure 3.2 that for SEI, the negative examples improve the MAP for FTP, IP, ATH, FA and AC by 17.6%, 9.4%, 7.9%, 2.8% and 10.0% respectively.

Desired Concept	Negative Examples
Female Tennis Players	Male Tennis Players
Film Actors	Musicians, Film Directors
Athletes	Football Platers, Skiers
Indian Politicians	Other Politicians
Australian Cricketers	Cricketers from other countries

Table 3.5: This table shows the negative examples that were used for different concepts. We see that the negative examples are closely related to the instances of the desired concept.

3.9.2 Positive vs. Negative Examples

In last subsection, we saw that adding negative examples substantially increases the performance of set-expansion for both SEI and SECW. In this subsection, we would compare the performance enhancement obtained by the addition of negative examples to that of the positive examples. Figure 3.3 shows such a comparison for the concept of female tennis players. Space restriction does not permit us to discuss other concepts. The 'circle' markers show the performance of set-expansion as we give more and more positive examples. We see that after 9 seeds, the MAP curves for positive examples for both SEI and SECW start becoming flat. For SEI, MAP actually decreases slightly from 74.7 to 74.3 as the number of seeds was increased from 9 to 30. Thus, we see that positive examples alone are not sufficient for fully specifying the concept.

On the other hand, only a few negative examples are able to significantly improve the performance of set-expansion as is evident by sudden jump in MAP values after we start adding negative examples. In Figure 3.3, the 'square' markers show the MAP values for both SEI and SECW after we start adding negative examples. For SECW, the MAP jumped from 57.9 (at 12 seeds) to 75.7 after specifying negative examples - an improvement of 17.8%. Similarly, for SEI, we find an improvement of 17.6% in MAP after specifying negative examples. MAP is 72.0 at 12 seeds and 89.6 at 21 seeds for SEI.

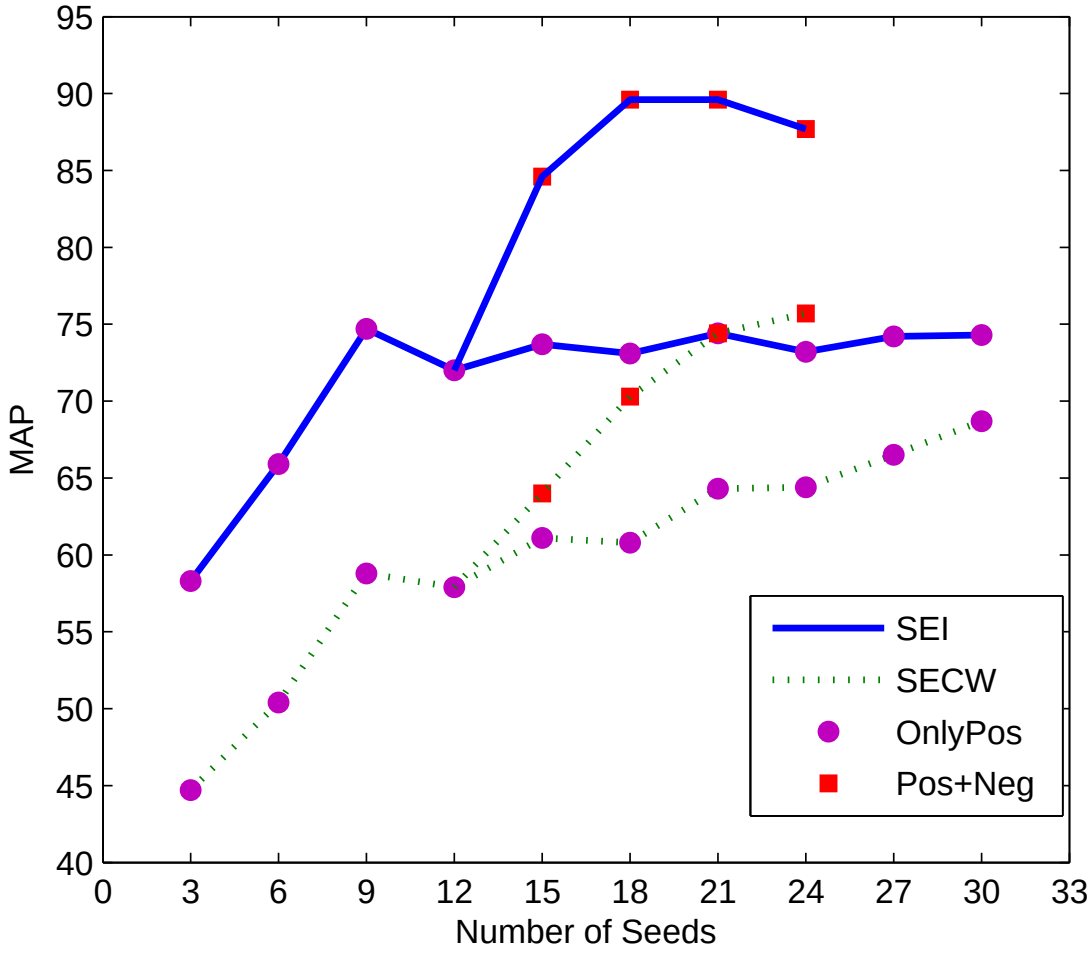


Figure 3.3: This figure compares the effect of positive and negative examples on the performance of set-expansion. After a certain stage, positive examples don't improve the performance of set-expansion significantly. Addition of negative examples along with the positive examples significantly boosts the MAP values for both SEI and SECW.

3.9.3 Active Learning

In Section 3.7, we presented a strategy for choosing good positive and negative examples based on active learning. In this section, we report the experimental results of the effect of choosing positive and negative examples on the performance of set-expansion. In Section 3.7, we noted that a good way of choosing positive examples is to select the examples near the end of the list and that the negative examples should be selected from the beginning of the list. Figure 3.4 compares this 'good' way of choosing positive

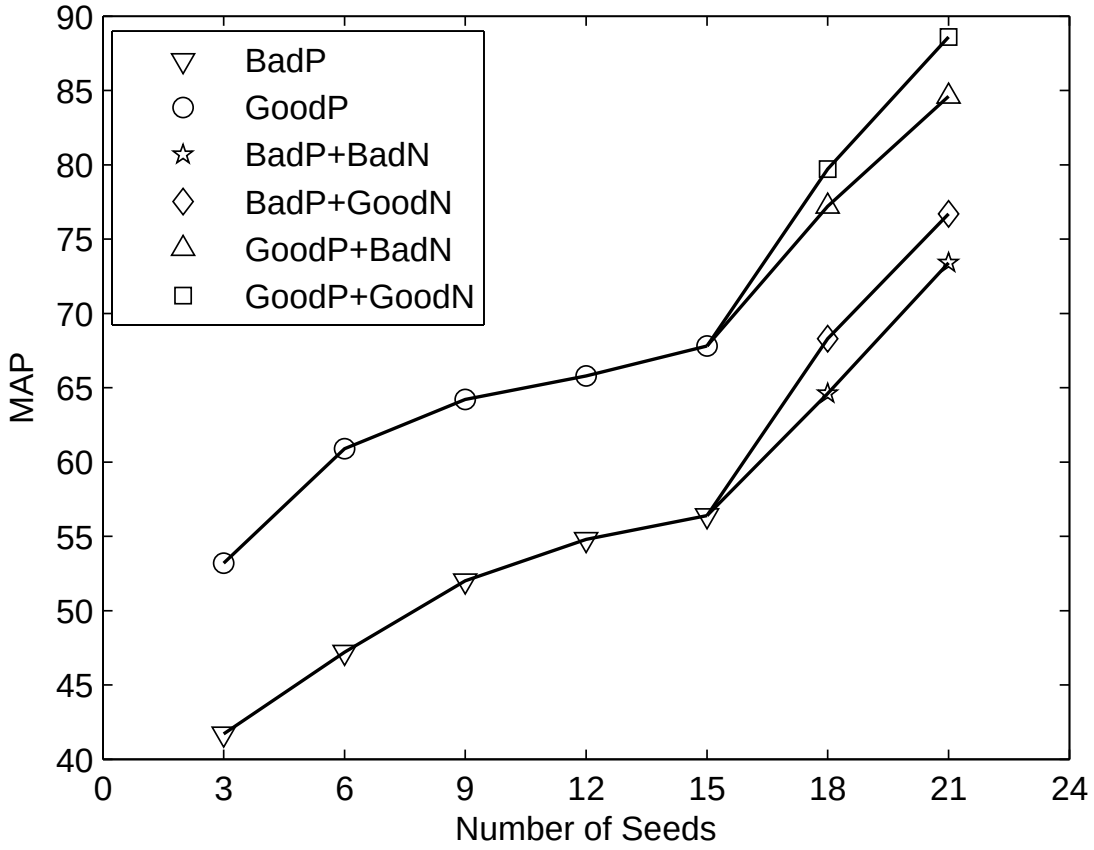


Figure 3.4: This figure shows the importance of proper choice of positive and negative examples. The good way of choosing positive and negative examples (*GoodP* and *GoodN*) was discussed in Section 3.7. At 21 seeds, the difference between the extreme combinations is 15.2%.

and negative examples (*GoodP* and *GoodN*) with a 'bad' way of choosing the examples (*BadP* and *BadN*) where the positive examples were taken from the beginning of the list and the negative examples were taken from the end of the list for SEI.

In Figure 3.4, we gave only positive examples till 15 seeds. We see that 'good' positive examples perform substantially better than the 'bad' positive examples. For example, at 12 seeds, MAP is 65.8 and 54.8 respectively for 'good' and 'bad' way of choosing positive examples.

For both good and bad ways of choosing positive examples, we had good and bad ways of choosing negative examples, thus giving 4 lines in Figure 3.4 after 15 seeds. We

see that when we choose the positive and negative examples according to Algorithm 4, we get better results than when we don't follow the algorithm. For example, at 21 seeds, 'GoodP+GoodN' has the MAP value of 88.6 whereas 'BadP+BadN' has a much lower MAP value of 73.4. The other 2 combinations of choosing the examples have the values in between 73.4 and 88.6.

3.10 Conclusions

In this chapter, we showed that the negative examples can significantly improve the performance of set-expansion by helping to better define the underlying concept. We incorporated weights in the commonly used centroid-based approach so that it can benefit from negative examples. We also developed an inference-based approach to set-expansion which naturally allows for negative examples and performs significantly better than the centroid-based approach. Finally, we presented an active-learning based strategy for choosing good positive and negative examples. The experimental results substantiate our claims.

Chapter 4

Joint Approach for Mention Detection

4.1 Introduction

In this chapter, we study the problem of concept recognition in the clinical domain. Most of state-of-the-art approaches for concept recognition in clinical domain can be categorized into two main categories [14]. In the first approach [19, 16, 20, 135, 15, 136, 137], concept boundaries and concept types are predicted in a single step using some sequence-prediction model like CRF [84], MEMM [138] etc. In the second approach, concept boundaries are first predicted using some sequence-prediction model and then a multi-class classifier is used to predict the concept types [21]. Both these approaches are limited by the fact that they can model only local dependencies (most often, first-order models like linear chain CRFs are used to allow tractable inference).

Clinical narratives, unlike newswire data, provide a domain with significant knowledge that can be exploited systematically. Knowledge in this domain can be thought of as belonging to two categories: (1) *Background Knowledge* captured in medical ontologies like UMLS[1], MeSH[2] and SNOMED CT[3] and (2) *Discourse Knowledge* expressed in the fact that the narratives adhere to specific writing style. While the former can be used by generating more expressive knowledge-rich features, the latter is more interesting from our current perspective, since it provides global constraints on what *output* structures are likely and what are not. We exploit this structural knowledge in our global inference

[1]http://www.nlm.nih.gov/research/umls/
[2]http://www.nlm.nih.gov/mesh/meshhome.html
[3]http://www.ihtsdo.org/snomed-ct/

formulation.

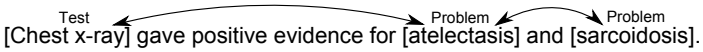
For global inference in NLP, [4] suggested an Integer Linear Programming (ILP) based approach. Since then, it has been used in a range of NLP tasks including semantic role labeling [139], the generation of route directions [140], temporal link analysis [141], set partitioning [142], syntactic parsing [143], sentence compression [144] and coreference resolution [145, 146]. However, in most of these works, researchers have focussed only on hard constraints while formulating the inference problem.

Formulating all the constraints as hard constraints is not always desirable because in many cases, constraints are not perfect. In this chapter, we propose Integer Quadratic Programs (IQPs) as a way of formulating the inference problem. IQPs is a richer family of models than ILPs and it enables us to easily incorporate soft constraints into the inference procedure. Our experimental results show that soft constraints indeed give much better performance than hard constraints.

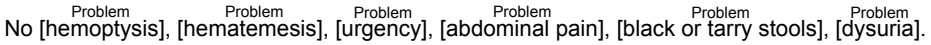
It should be noted that it is possible to reduce IQPs to ILPs using variable substitution. However, resulting ILPs can be exponentially larger than original IQPs. Thus, IQPs provide a strict modeling advantage compared to ILPs.

Finally, our results demonstrate that our joint inference procedure was successfully able to exploit the structural knowledge contained in clinical narratives. This, in turn, helped us to obtain statistically significant performance improvements over existing state-of-the-art method for concept recognition in clinical domain. We report detailed results on publicly available datasets so that future works can compare their approach with ours.[4]

[4]We would make our software and evaluation script publicly available for research use.



(a) Example 1



(b) Example 2

Figure 4.1: This figure motivates the global inference procedure we used. For discussion, please refer to §4.2.3.

4.2 Methodology

Task Description Input consists of clinical reports in free-text (unstructured) format. The task is: (1) to identify the boundaries of medical concepts and (2) to assign types to such concepts. Each concept can have 3 possible types, namely (1) Test (laboratory tests etc.), (2) Treatment (treatments, medications, surgeries, etc.) and (3) Problem (symptoms, diseases, complaints, etc.). We would refer to these three types by `TEST`, `TRE` and `PROB` in the following discussion.

Our Approach First of all, we find the concept boundaries using multiple boundary detectors. Then we use a multi-class classifier which tell us the probabilities with which a particular concept takes different types (`TEST`, `TRE`, `PROB` or `NULL`). These probability values are then used in an inference procedure which computes the final assignment of types to concepts. The different stages of our approach are described in more detail in the following subsections.

4.2.1 Step 1: Finding Concept Boundaries

In the first step, we identify the concept boundaries using multiple boundary detectors. For boundary detection, we used CRF in one of our modules. In two other modules, we used deterministic finite state automata (DFSAs). DFSAs were designed to detect some regular expressions (like *medication-dosage patterns* and *test-value patterns*) in the

	Feature
	SurfaceForm(w_i), SurfaceForm(w_{i-1}), SurfaceForm(w_{i+1}), POS(w_i), POS(w_{i-1}), POS(w_{i+1}), SP(w_i), SP(w_{i-1}), SP(w_{i+1}), conj[POS(w_{i-1}), POS(w_i)], conj[POS(w_i), POS(w_{i+1})], conj[POS(w_{i-1}), POS(w_i), POS(w_{i+1})], conj[SP(w_{i-1}), SP(w_i)], conj[SP(w_i), SP(w_{i+1})], conj[SP(w_{i-1}), SP(w_i), SP(w_{i+1})]
✓	MT(w_i), MT(w_{i-1}), MT(w_{i+1}), conj[MT(w_{i-1}), MT(w_i)], conj[MT(w_i), MT(w_{i+1})], conj[MT(w_{i-1}), MT(w_i), MT(w_{i+1})]

(a) Features for finding Concept Boundaries

	Feature
	Tokens of the concept, Full text of the concept (after normalization), concept bi-grams, concept headword, suffixes of concept headword, capitalization pattern of concept, shallow parse label of constituent containing headword, whether concept contains only digits
✓	Metamap type of concept, MetaMap type of headword, Occurrence of concept in MeSH, Occurrence of concept in SNOMED CT, MeSH Descriptor, SNOMED CT Descriptor

(b) Features for finding concept types

Table 4.1: This table shows the features used for finding (a) concept boundaries and (b) concept types. ✓symbols in this table denote features derived from Domain-Specific Knowledge sources.

clinical text. CRF module used BIO encoding for representing chunks and it was implemented using MALLET toolkit [18]. Features used by CRF module are described in Table 4.1a. In this table, features have been divided into 2 rows. Features in first row are the baseline features which are typically used for boundary detection. Features in second row have ✓symbol in front of them. ✓symbol denotes that these features are derived from domain-specific knowledge sources. Following abbreviations have been used in this table: SP = Shallow Parse and MT = MetaMap Type. In Table 4.1, "*conj*" denotes conjunction of features. Knowledge-derived features will be explained in more detail in next section (§4.3).

4.2.2 Step 2: Finding Concept Types

After determining concept boundaries, the next step is to determine the probabilities for concept types. For finding these probabilities, we train a multi-class SVM classifier [147]. Table 4.1b gives the features used for training this classifier. Just like in previous subsec-

tion, features which have a ✓symbol in front of them are derived from domain-specific knowledge sources.

4.2.3 Step 3: Inference Procedure

The final assignment of types to concepts is determined by an inference procedure. The basic principle behind our inference procedure is: *"Types of concepts which appear close to one another are often closely related. For some concepts, type can be determined with more confidence. And relations between concepts' types guide the inference procedure to determine the types of other concepts."* We will now explain it in more detail with the help of examples. Figure 4.1 shows two sentences in which the concepts are shown in brackets and correct (gold) types of concepts are shown above them.

Now, consider first and second concepts in Figure 4.1a. These concepts follow the pattern: *[Concept1] gave positive evidence for [Concept2]*. Let us call this pattern as *P1*. In clinical narratives, such a pattern very strongly suggests that *Concept1* is of type `TEST` and *Concept2* is of type `PROB`. So, we can impose a constraint on the output of type classifier that whenever it sees that two concepts follow pattern *P1*, then first concept should be assigned the type `TEST` and second concept should be assigned the type `PROB`. Another pattern that we see in Figure 4.1a is: *[Concept1] and [Concept2]* between second and third concepts. Such a pattern suggests that the two concepts should have the same type. Thus, we see that same concept (*atelectasis* in our example) can appear in multiple constraints. In other words, different constraints can interact with each other.

Next, consider different concepts in Figure 4.1b. All these concepts are separated by commas and hence, form a list. It is highly likely that all the concepts which appear in a list should have the same type. The advantage of such a constraint can be explained as follows: Suppose that type classifier is not sure about the type of third concept *urgency* (which means that it doesn't give a high probability to any of the concept types). But

	Pattern
1	using [TRE] for [PROB]
2	[TEST] showed [PROB]
3	Patient presents with [PROB] status post [TRE]
4	use [TRE] to correct [PROB]
5	[TEST] to rule out [PROB]
6	Unfortunately, [TRE] has caused [PROB]
7	[Concept1], [Concept2], [Concept3], ...
8	[Concept1] versus [Concept2]

Table 4.2: This table shows some of the patterns that were used in constraints.

the type classifier is confident that the second and the fourth concepts (*hematemesis* and *abdominal pain*) are both problems. Based on the constraint that all the elements in a list should have the same type, type classifier can correctly infer that *urgency* should also be of type problem.

It is also to be noted that each constraint can either be coded as a hard-constraint or a soft-constraint. Implementation of the inference procedure will be discussed in a later section (§4.4).

Acquisition of Constraints A small portion of training data (15 documents) was used to obtain constraints. From this data, a list of frequent patterns was generated. This list was then manually filtered to get the final constraints. Some of the patterns that were used in constraints are shown in Table 4.2. A total of 18 patterns were used in the final system.

4.3 Domain-Specific Knowledge Features

MetaMap Types MetaMap [6] is a configurable program which takes free-text as input. It identifies the UMLS concepts appearing in the text. Thus, it can be thought of as a shallow parser for the medical text. Associated with each UMLS concept is a semantic

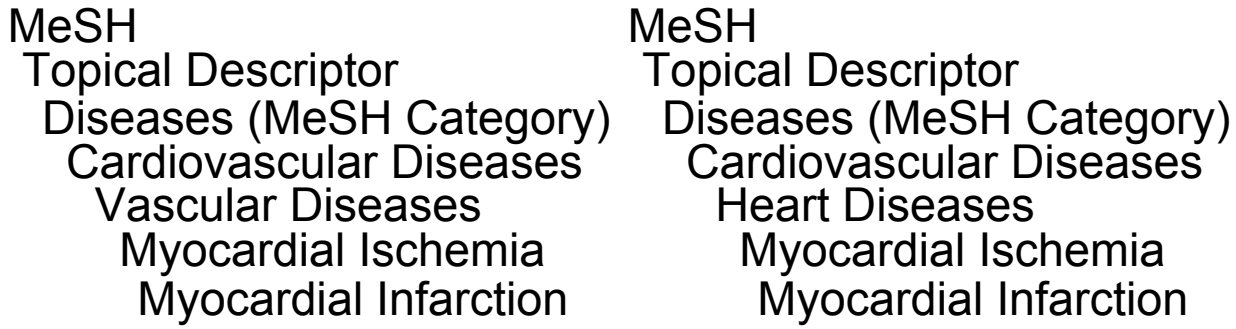


Figure 4.2: Two different paths from root to concept in MeSH Parent Graph for *Myocardial Infarction*

type (like *Acquired Abnormality*, *Clinical Drug*, *Neoplastic Process* etc.). We used UMLS concepts (given by MetaMap) and their associated semantic types as features in our system. Our usage of MetaMap is similar to that of [20].

Clinical Descriptors Medical ontologies like MeSH, SNOMED CT etc. represent medical concepts in a hierarchical fashion. In such hierarchies, there is often more than 1 path from the root of the hierarchy to any given concept. Collection of all the paths from root of the hierarchy to any given concept is a directed acyclic graph (DAG). We would refer to such a DAG as a *parent graph* of a concept. As an example, Figure 4.2 shows two different paths in MeSH *parent graph* for the concept *Myocardial Infarction*. We designed clinical descriptors to exploit the important information contained in parent graphs of concepts.

Briefly speaking, clinical descriptor for any concept is formed by taking the most frequent members at the top few levels (along all the paths) in the parent graph of the concept. For example, clinical descriptor of the concept *Myocardial Infarction* (or *Heart Attack*) consists of the following terms: *Disease*, *Traumatic Injury*, *Disease affecting entire cardiovascular system*, *Myocardial Ischemia*, *Heart Disease* etc.

4.4 Modeling Global Inference

In this section, we discuss in detail our approach to inference. Inference is done at the level of sentences. Suppose there are m concepts in a sentence. Each of the m concepts has to be assigned one of the following types: `TEST` (1), `TRE` (2), `PROB` (3) or `NULL` (4) where the numbers in parenthesis denote the corresponding values that will be used in the inference problem. To represent this as an inference problem, we define the indicator variables $x_{i,j}$ where i takes values from 1 to m (corresponding to concepts) and j takes values from 1 to 4 (corresponding to 4 possible types). $p_{i,j}$ refers to the probability that i^{th} concept is of j^{th} type. $p_{i,j}$'s are given by the SVM classifier for concept types which was described in §4.2.2.

So, we can write the following optimization problem to find the optimal concept types:

$$max \sum_{i=1}^{m} \sum_{j=1}^{4} x_{i,j} \cdot p_{i,j} \tag{4.1}$$

$$\text{subject to } \sum_{j=1}^{4} x_{i,j} = 1 \qquad \forall i \tag{4.2}$$

$$x_{i,j} \in \{0, 1\} \qquad \forall i, j \tag{4.3}$$

The Objective function in Equation (4.1) expresses the fact that we want to maximize the probability of assignment of concept types. Equation (4.2) enforces the constraint that each concept has a unique type which means that for every i, only one of the variables $x_{i,j}$ can take the value 1 while the remaining being 0. We would refer to these as **Type-1** constraints. Equation (4.3) simply expresses that the variables $x_{i,j}$ are indicator variables.

4.4.1 Constraints Used

In this subsection, we will describe two additional types of constraints (**Type-2** and **Type-3**) that were added to the optimization procedure described above. Whereas **Type-1** constraints described above were formulated as *hard constraints*, **Type-2** and **Type-3** constraints are formulated as *soft constraints*.

Type-2 Constraints

These constraints are further divided into 2 types as follows:

Type-2a Constraints Certain constructs like comma, conjunction, etc. suggest that the 2 concepts appearing in them should have the same type. Figure 4.1b shows an example of such type of constraints. Now, we will discuss how to enforce this requirement in the optimization problem. Suppose, there are n_2 such constraints. Also, assume that l^{th} constraint says that the concepts $\mathcal{R}_l$ and $\mathcal{S}_l$ should have the same type. Now, we define a variable w_l as follows:

$$w_l = \sum_{m=1}^{4} (x_{\mathcal{R}_l,m} - x_{\mathcal{S}_l,m})^2 \tag{4.4}$$

Now, if the concepts $\mathcal{R}_l$ and $\mathcal{S}_l$ have the same type, then w_l would be equal to 0. Also, if the concepts $\mathcal{R}_l$ and $\mathcal{S}_l$ don't have the same type, then w_l would be equal to 2. So, l^{th} constraint can be enforced by subtracting $(\rho_2 \cdot \frac{w_l}{2})$ from the objective function given by Equation (4.1). Thus, a penalty of ρ_2 would be enforced iff l^{th} constraint is violated.

Type-2b Constraints Certain patterns (e.g. *"using [concept1] for [concept2]"*) suggest that the 2 concepts appearing in them should *not* have the same type. These constraints are very similar to Type-2a constraints. Suppose, there are n_4 such constraints. Also, assume that l^{th} constraint says that the concepts $\mathcal{E}_l$ and $\mathcal{F}_l$ should *not* have the same

type. Now, we define a variable y_l (similar to w_l in Equation (4.4)) as follows:

$$y_l = \sum_{m=1}^{4} (x_{\mathcal{E}_l,m} - x_{\mathcal{F}_l,m})^2 \tag{4.5}$$

Now, if the concepts $\mathcal{E}_l$ and $\mathcal{F}_l$ have the same type, then y_l would be equal to 0. Otherwise, y_l would be equal to 2. So, l^{th} constraint can be enforced by subtracting $(\rho_2 \cdot (1 - \frac{y_l}{2}))$ from the objective function given by Equation (4.1)[5]. Thus, a penalty of ρ_2 would be enforced iff l^{th} constraint is violated.

Type-3 Constraints

Some short patterns suggest possible types for the concepts which appear in them. Each such pattern, thus, enforces constraint on the types of concepts which appear in them. Figure 4.1a shows an example of such type of constraints. Suppose there are n_3 such constraints. Also, assume that the k^{th} constraint says that the concept $\mathcal{A}_{1,k}$ should have the type $\mathcal{B}_{1,k}$ and that the concept $\mathcal{A}_{2,k}$ should have the type $\mathcal{B}_{2,k}$. Equivalently, k^{th} constraint says the following in boolean algebra notation: $(x_{\mathcal{A}_{1,k},\mathcal{B}_{1,k}} = 1) \wedge (x_{\mathcal{A}_{2,k},\mathcal{B}_{2,k}} = 1)$. For k^{th} constraint, we introduce one more variable $z_k \in \{0, 1\}$ which satisfies the following condition:

$$z_k = 1 \Leftrightarrow k^{th}\text{constraint is satisfied} \tag{4.6}$$

This can be re-written as:

$$z_k = 1 \Leftrightarrow (x_{\mathcal{A}_{1,k},\mathcal{B}_{1,k}} = 1) \wedge (x_{\mathcal{A}_{2,k},\mathcal{B}_{2,k}} = 1) \tag{4.7}$$

Now, using boolean algebra, it can be shown that Equation (4.7) is equivalent to the following linear inequalities:

[5]We use the same penalty for Type-2a and Type-2b constraints because of their similarity.

$$
\begin{aligned}
& max \sum_{i=1}^{m} \sum_{j=1}^{4} x_{i,j} \cdot p_{i,j} - \sum_{k=1}^{n_3} \rho_3 (1 - z_k) \\
& - \sum_{l=1}^{n_2} \left(\rho_2 \cdot \frac{\sum_{m=1}^{4} (x_{\mathcal{R}_l,m} - x_{\mathcal{S}_l,m})^2}{2} \right) \\
& - \sum_{l=1}^{n_4} \left(\rho_2 \cdot (1 - \frac{\sum_{m=1}^{4} (x_{\mathcal{E}_l,m} - x_{\mathcal{F}_l,m})^2}{2}) \right)
\end{aligned} \tag{4.9}
$$

$$
\text{subject to } \sum_{j=1}^{4} x_{i,j} = 1 \qquad \forall i \tag{4.10}
$$

$$
x_{i,j} \in \{0,1\} \qquad \forall i,j \tag{4.11}
$$

$$
x_{\mathcal{A}_{1,k},\mathcal{B}_{1,k}} \geq z_k \quad \forall k \in \{1...n_3\} \tag{4.12}
$$

$$
x_{\mathcal{A}_{2,k},\mathcal{B}_{2,k}} \geq z_k \quad \forall k \in \{1...n_3\} \tag{4.13}
$$

$$
z_k \geq x_{\mathcal{A}_{1,k},\mathcal{B}_{1,k}} + x_{\mathcal{A}_{2,k},\mathcal{B}_{2,k}} - 1 \forall k \in \{1...n_3\} \tag{4.14}
$$

Figure 4.3: Final Optimization Problem which has been formulated as an Integer Quadratic Program (IQP)

$$
\left.
\begin{aligned}
x_{\mathcal{A}_{1,k},\mathcal{B}_{1,k}} &\geq z_k \\
x_{\mathcal{A}_{2,k},\mathcal{B}_{2,k}} &\geq z_k \\
z_k &\geq x_{\mathcal{A}_{1,k},\mathcal{B}_{1,k}} + x_{\mathcal{A}_{2,k},\mathcal{B}_{2,k}} - 1
\end{aligned}
\right\} \tag{4.8}
$$

Thus, we can incorporate the k^{th} constraint in the optimization problem by adding to it the above constraints (given by Equation (4.8)) and by subtracting $(\rho_3(1 - z_k))$ from the objective function given by Equation (4.1). Thus, if k^{th} constraint is satisfied ($z_k = 1$), then no penalty is imposed but if the constraint is not satisfied ($z_k = 0$), then a penalty of ρ_3 is imposed.

4.4.2 Final Optimization Problem - An IQP

After incorporating all the constraints mentioned above, the final optimization problem is shown in Figure 4.3. Please note that in the above problem, the only unknown vari-

ables are $x_{i,j}$'s. Other variables like $p_{i,j}$, $\mathcal{A}_{i,k}$, $\mathcal{R}_l$, $\mathcal{S}_l$ etc. are given by outside sources like SVM classifier and pattern finding algorithms.

Optimization problem shown in Figure 4.3 is an Integer Quadratic Program (IQP) which means that it has quadratic objective function and linear constraints. We used Gurobi Optimization toolkit to solve such IQPs. Gurobi is very efficient in solving such IQPs. In our case, it solves 76 IQPs per second on a quad-core server with Intel Xeon X5650 @ 2.67 GHz processors and 50 GB RAM.

4.5 Experiments and Results

4.5.1 Datasets

For our experiments, we used the datasets provided by i2b2/VA team as part of 2010 i2b2/VA shared task[6] [14]. The datasets used for shared task contained de-identiiňĄed clinical reports from three medical institutions: Partners Healthcare (`PH`), Beth-Israel Deaconess Medical Center (`BIDMC`) and the University of Pittsburgh Medical Center (UPMC). UPMC data was divided into 2 sections, namely discharge (`UPMCD`) and progress notes (`UPMCP`). A total of 349 training reports and 477 test reports were made available to the participants. However, data which came from UPMC (more than 50% data) was not made available for public use. As a result, we had only 170 clinical reports for training and 256 clinical reports for testing. Table 4.3 shows the number of clinical reports made available by different institutions. The strikethrough text in this table indicates that the data was not made available for public use and hence, we couldn't use it. We used about 25% of the training data as a development set.

[6]https://www.i2b2.org/NLP/Relations/

	PH	BIDMC	~~UPMCD~~	~~UPMCP~~
Train	97	73	~~98~~	~~81~~
Test	133	123	~~102~~	~~119~~

Table 4.3: Dataset Characteristics

4.5.2 Results

Evaluation We report precision, recall and F1 scores for concept recognition[7]. We wrote our own script for calculating these scores. i2b2 also provided a script for calculating these scores. The scores reported by i2b2 script are slightly higher than the ones given by our script. However, we could not verify the reason for this because we didn't have the source code of i2b2 evaluation script. Along with the scores reported by our script, we also report the overall score given by i2b2 script to facilitate comparison with other future works.

In this section, we would refer to following five systems:

1. *Baseline* **(B)**: This system doesn't perform any global inference. Also, it doesn't use features derived from domain-specific knowledge sources (marked by ✓in Table 4.1) for training the classifiers.

2. *Baseline* + *Knowledge* **(BK)**: Like *Baseline* system, this system doesn't perform global inference. However, it uses all the features for training the classifiers.

3. *Baseline* + *Constraints* **(BC)**: Like *Baseline* system, this system doesn't use the knowledge based features. However, it performs global inference as explained in Section 4.4.

4. *Baseline* + *Knowledge* + *Constraints* **(BKC)**: This is our final system. It performs global inference and also uses all the features for training the classifiers.

[7]Overlapping concepts are considered valid matches.

5. **BKC-HARD**: This is similar to **BKC** system. However, it sets $\rho_2 = \rho_3 = 1$ which effectively turns **Type-2** and **Type-3** constraints into hard constraints by imposing very high penalty.

Importance of Soft Constraints

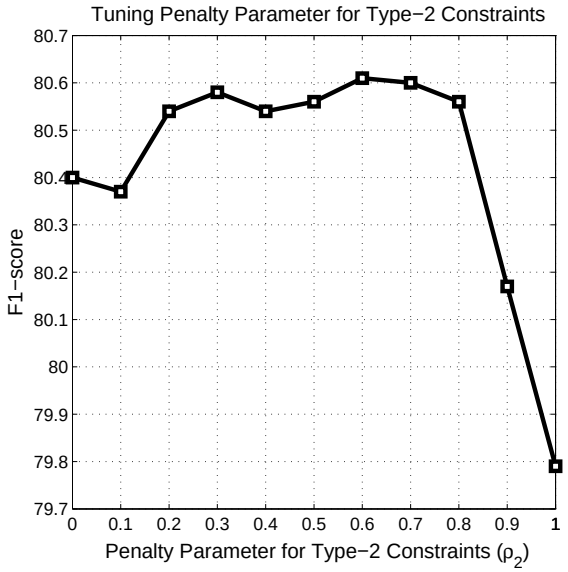
Figures 4.4a and 4.4b show the effect of varying the penalties (ρ_2 and ρ_3) for **Type-2** and **Type-3** constraints respectively. These figures show the F1-score of **BKC** system on the development set. Penalty of 0 means that the constraint is not active. As we increase the penalty, the constraint becomes stronger. As the penalty becomes 1, the constraint becomes hard in the sense that final assignments must respect the constraint.

We observe from Figure 4.4a that for **Type-2** constraints, F1-score attains 2 local maxima - one at $\rho_2 = 0.3$ and another at $\rho_2 = 0.6$. Global maxima is attained at $\rho_2 = 0.6$. As we increase ρ_2 after 0.8, the F1 score decreases rapidly.

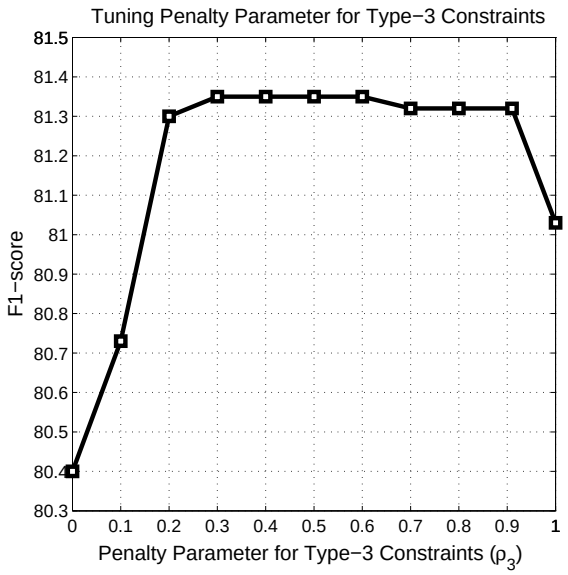
Similar trend is observed for **Type-3** constraints in Figure 4.4b. F1-score improves as we start increasing the penalty ρ_3 from the value of 0. F1-score reaches the maximum when ρ_3 is 0.3. It remains the same till $\rho_3 = 0.6$. As we increase ρ_3 further, the performance degrades.

Thus, we see the importance of tuning the penalty parameters. Although the constraints are useful, improperly tuned penalties can worsen the performance. Based on our findings as shown in Figure 4.4, we chose $\rho_2 = 0.6$ and $\rho_3 = 0.3$ in our experiments.

Hard vs Soft Constraints Table 4.4 compares the performance of **BKC-HARD** system with that of **BKC** system. First 3 rows in this table show the performance of both systems for the individual categories (`TEST`, `TRE` and `PROB`). Fourth row shows the overall score of both systems. All these scores were calculated using our own script. Fifth (or last) row reports the overall score of the systems as given by i2b2 script. **BKC** system outperformed **BKC-HARD** system on all the categories by statistically significant differences at



(a) Type-2 Constraints



(b) Type-3 Constraints

Figure 4.4: These figures show the result of tuning the penalty parameters (ρ_2 and ρ_3) for soft constraints.

$p = 0.05$ according to Bootstrap Resampling Test [148]. For the `OVERALL` category, **BKC** system improved over **BKC-HARD** system by $(86.1 - 85.1 =)1.0$ F1 points.

	BKC-HARD	BKC
TEST	84.7	**85.8**
TRE	84.7	**85.7**
PROB	85.6	**86.7**
OVERALL	85.1	**86.1**
i2b2	86.3	**86.9**

Table 4.4: This table shows that the system using soft constraints consistently performs much better than the one using hard constraints.

	B			BK			BC			BKC		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
TEST	92.4	79.4	85.4	91.9	80.2	85.7	92.7	79.6	85.7	92.1	80.4	**85.8**
TRE	92.1	73.6	81.8	92.0	79.5	85.3	92.3	76.8	83.8	92.0	80.2	**85.7**
PROB	83.6	83.6	83.6	88.9	83.7	86.3	85.9	83.8	84.8	89.6	83.9	**86.7**
OVERALL	88.4	79.4	83.6	90.7	81.4	85.8	89.6	80.5	84.8	91.0	81.7	**86.1**
i2b2	87.9	83.1	85.4	90.3	83.5	86.7	89.1	83.3	86.1	90.6	83.5	**86.9**

Table 4.5: This table compares the performance of four systems: (1) Baseline **(B)**, (2) Baseline + Knowledge **(BK)**, (3) Baseline + Constraints **(BC)** and (4) Baseline + Knowledge + Constraints **(BKC)**. Our final system, **BKC**, consistently performed the best. This result is statistically significant at $p = 0.05$ according to bootstrap resampling test. For detailed discussion, please refer to §7.11.

Comparing with state-of-the-art baseline

In 2010 i2b2/VA shared task, majority of top systems were CRF-based models. So, we decided to use CRF as our baseline. Table 7.1 compares the performance of 4 systems: **B**, **BK**, **BC** and **BKC**. As pointed out before, our **BK** system uses all the knowledge-based features and is very similar to the top-performing systems in i2b2 challenge. We see from Table 7.1 that **BKC** system consistently performed the best for individual as well as overall categories. This result is statistically significant at $p = 0.05$ according to Bootstrap Resampling Test [148]. Thus, the constraints that we used helped us to obtain statistically significant performance improvements over state-of-the-art **BK** system[8]. We also see from Table 7.1 that **BC** system performed significantly better than *Baseline* **(B)** system.

[8]Please note that the results reported in Table 7.1 can not be directly compared with those reported in the challenge because we had only a fraction of the original training and testing data.

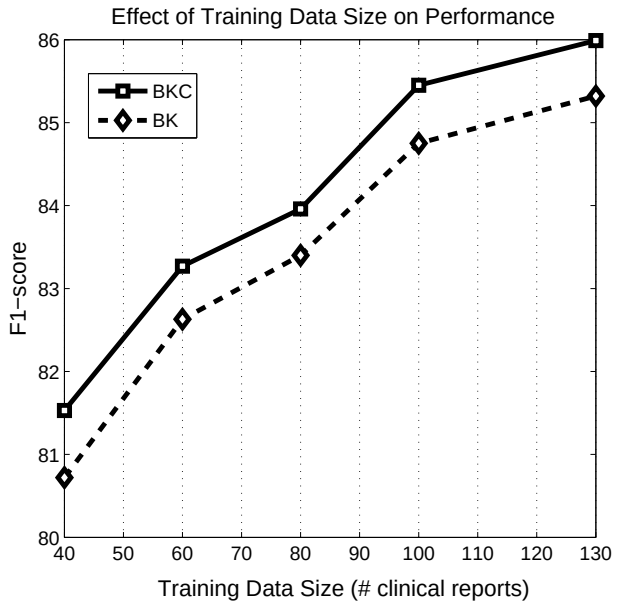


Figure 4.5: This figure shows the effect of training data size on performance of concept recognition.

Thus, the constraints are helpful even in the absence of knowledge-based features.

Since we report results on publicly available datasets, the future works would be able to compare their results with ours. We would also make our system and evaluation script publicly available for use by other researchers.

Effect of training data size

We next examine the effect of training data size on the performance of concept recognition. In Figure 4.5, we report the overall F1-score on a part of the development set as we vary the size of training data from 40 documents to 130 documents. We notice that the performance increases steadily as more and more training data is provided. This suggests that if we could train on full training data as was made available during challenge, the final scores would be much higher. We also notice from the figure that **BKC** system consistently performs better than state-of-the-art **BK** system as we vary the size of training data. This shows that the joint inference procedure designed by us is very

robust.

4.6 Discussion and Related Work

Joint inference approaches which incorporate declarative knowledge in statistical models have been widely used in last few years to solve Information Extraction (IE) tasks. Some of the representative models for joint inference include posterior regularization [91], generalized expectations [149, 150], constraint-driven learning [87], methods based on integer programs [2], gibbs sampling [86] and recently the methods that are based on dual-decomposition [151]. Among these approaches, posterior regularization, generalized expectations and constraint-driven learning were proposed for semi-supervised setting. However, in this chapter, we are considering a fully supervised scenario.

The optimization problem that we proposed in this chapter can be efficiently solved by modern optimizers like Gurobi. Since we can perform exact inference using such optimizers, we don't need to resort to approximate techniques like gibbs sampling, dual decomposition etc.

[2] suggested the use of integer programs to model joint inference in a fully supervised setting. Their approach is most closely related to ours. However, they used only hard constraints in their inference formulation. Another approach that is related to ours is that of joint learning. In this approach, the constraints are encoded as features over output space in a structured prediction model like CRF. As Figure 4.1 shows, our constraints are quite expressive and involve long range dependencies. We found that incorporating such constraints in CRF makes the training prohibitively expensive. Similar observation has been reported previously by [5] where the authors found the results with joint learning models to be quite unsatisfactory. In particular, the authors showed that a simple perceptron-based model (which used constraints only during inference) significantly outperformed a joint model based on CRF. It is also known that joint learning models,

	CRF			PTB-Chunker			i2b2-Chunker		
	P	R	F1	P	R	F1	P	R	F1
TEST	88.8	81.9	85.2	88.7	79.6	83.9	88.8	82.1	85.3
TRE	91.6	81.1	86.0	90.9	73.3	81.2	91.7	81.0	86.0
PROB	89.1	85.3	87.1	89.9	78.5	83.8	90.6	85.1	87.8
OVERALL	89.8	83.2	86.3	89.9	77.1	83.0	90.5	83.1	86.6

Table 4.6: This table shows the comparison between chunker and CRF on test portion of partners corpus.

even when successful, require much more training data than local models.

Chang et al. [152] recently used soft constraints in Constrained Conditional Models. However, unlike us, they performed approximate inference using beam search. In this chapter, we showed that it is possible to do exact inference efficiently even while using soft constraints.

4.7 Comparing with Chunker

In this section, we compare CRF with a shallow parser for the task of mention detection. Datasets used for experiments in this section are same as the coreference datasets which were distributed in 2011 i2b2 coreference challenge. We used Partners and Beth sections of the corpora. Training portion of Partners and Beth contains 136 and 115 documents respectively. Test portion of Partners and Beth contains 94 and 79 documents respectively.

Tables 4.6 and 4.7 show the comparison between shallow parser (or chunker) and CRF for doing mention detection. We used MALLET implementation of CRF and for chunker, we used the Illinois chunker. For chunker, we have 2 sets of results. In the first case, the chunker was trained on PTB data and in the second case, chunker was trained on i2b2 data.

The results shown in Tables 4.6 and 4.7 are quite interesting. We see that PTB chun-

	CRF			PTB-Chunker			i2b2-Chunker		
	P	R	F1	P	R	F1	P	R	F1
TEST	90.1	74.2	81.4	89.8	72.9	80.5	90.8	83.5	87.0
TRE	90.0	79.1	84.2	91.0	72.7	80.8	88.9	79.4	83.8
PROB	87.4	84.1	85.7	90.8	74.8	82.1	91.2	82.7	86.8
OVERALL	89.0	79.3	83.9	90.5	73.6	81.2	90.4	82.0	86.0

Table 4.7: This table shows the comparison between chunker and CRF on test portion of beth corpus.

ker performs poorly than CRF. For Partners dataset, PTB chunker 3.3 F1 points behind the CRF and for Beth dataset, PTB chunker is 2.7 F1 points behind the CRF. However, after retraining on i2b2 dataset, chunker's performance improves considerably and i2b2-chunker actually gives better performance than the CRF. For Partners dataset, i2b2-chunker performs better than the CRF by 0.3 F1 points and for Beth dataset, i2b2-chunker performs better than the CRF by 2.1 F1 points.

Chapter 5

Timex Extraction

5.1 Timex Extraction

Task Description: In Timex extraction task, the system is supposed to identify the spans and attributes of the temporal expressions in the text. There are 3 attributes associated with each event, namely TYPE, value (VAL) and modifier (MOD). Type attribute can have 4 possible values: DATE, TIME, Duration (DUR) and Frequency (FREQ). VAL attribute gives the time (value) associated with the temporal expression. Finally, a temporal expression can have one of the following 7 modes: NA, APPROX, MORE, LESS, START, MIDDLE, END.

Approach Used: Our overall approach for timex extraction is rule-based as rule-based methods have been shown to give the best results to date for this task. For timex extraction task, first of all we determine the "Admission Date" and "Discharge Date" as given in the clinical narrative. Then we use HeidelTime as a baseline temporal extraction system. And finally, we use our own rules which are specifically designed for clinical narratives to complement the output of HeidelTime. We explain each of these components of our temporal extraction system in the following subsections.

5.1.1 Finding Section Times

In this subsection, we describe the method used by us to determine "Admission Date" and "Discharge Date" in the clinical narrative.

1. Clinical narratives in i2b2 datasets typically have 4 sections: Admission Date, Discharge Date, History of Present Illness and Hospital Course. A new section is determined by the fact that the line ends with a semicolon.

2. After we determine the first line where “Admission Date” and “Discharge Date” sections begin, we use a regular expression to find out whether the following line has a date in it.

3. Now, a date can be written in several different formats. For example, we can write the same date Sep 14, 1999 in the following ways: 1999-09-14, 99/09/14, 09/14/99, 09/14/1999, 09-14 etc. Clinical reports list the dates in all such formats. Correctly determining the date requires consideration of the constraints on the date fields, namely day, month and year. For example,

   (a) Since the narratives were all taken from US hospitals, they put month before date.

   (b) Month takes value between 1-12 and day takes value between 1-31

   (c) Year can appear either as a first field or a last field.

   (d) Year can either be in 2 digit format or in 4 digit format.

   (e) It is possible that year may not be there at all (as it can be determined from context). But since we are dealing with clinical reports, the day field will generally be there.

   (f) Both "-" and "/" can act as separators between various fields of date expression.

4. Considering the above things, we designed the following 10 regular expressions using JodaTime Library:

   (a) yy-MM-dd

(b) MM-dd-yy

(c) yyyy-MM-dd

(d) MM-dd-yyyy

(e) yy/MM/dd

(f) MM/dd/yy

(g) yyyy/MM/dd

(h) MM/dd/yyyy

(i) MM-dd

(j) MM/dd

The given date expression was made to match with each of these regular expressions. JodaTime itself takes care of consistency checks on the date fields. First regular expression which matched the date expression was used to determine the date fields. This algorithm gave us almost 100% accuracy on i2b2 datasets.

The above procedure was also used to determine dates in other sections of the clinical narrative.

5.1.2 Using HeidelTime

1. We used HeidelTime as a baseline to obtain the temporal expressions. For the temporal expressions that HeidelTime identifies, it also gives the TYPE, VAL and MOD attributes. HeidelTime also expects the "Document (Section) Creation Time (DCT)" as one of the inputs which serves to resolve the ambiguity while determining the value of some temporal expressions. For "History of Present Illness" section, DCT is given to be the Admission date and for "Hospital course" section, DCT is given to be the Discharge Date.

2. HeidelTime assigns the type SET to timexes of type FREQ. So, we replace the SET type in HeidelTime with FREQ while outputting the result. Also, in the VAL attribute for timexes of type FREQ, HeidelTime doesn't prefix the value with R. So, we ourselves prefix the value of FREQ timexes with "R". In some cases, the VAL attribute given by HeidelTime is not formatted according to the i2b2 guidelines. So, we do a post-processing step to properly format the HeidelTime results. For example, for the phrase "several months", HeidelTime gives the value *PXM*. We change it to *P3M* and set the modifier attribute to *APPROX* as specified in i2b2 guidelines.

3. Next, the MOD attribute produced by HeidelTime is mapped to the i2b2 MOD attribute. For example, "more_than" of HeidelTime is same as "more" of i2b2.

5.1.3 Rules for Clinical Narratives

Although HeidelTime is very good in identifying timexes written in general English, it is not able to identify the clinical timexes. So, we added the following rules in our system to identify the clinical timexes.

1. Some timexes are of the form "POD#*n*". For such timexes, first we identify the temporal expression corresponding to the "operation date". If we don't find such an expression, then we set the operation date to be same as admission date. Next, we set the value of "POD#*n*" to *n* days after the operation date. We also capture other variations of "POD#*n*" like "postoperative day *n*" etc.

2. A similar procedure as described above is also followed for the expressions like "hospital day *n*" or "HD *n*". Reference date for computing the value of such timexes is the admission date.

3. We identify clinical expressions of the type "x *n*" or "x*n*" or "times *n*" etc. Such

expressions are of type frequency and identify only the number of times for which an event is repeated but don't specify the interval for such repetition. A value of "R*n*" is given to these expressions.

4. If year value is not specified for some of the dates mentioned within the document, then we set the year based on admission date or discharge date.

5. Several timexes contain the word "day" in them. Expressions like "per day" are assigned the type FREQ with a period of 1 day. For expressions like "2 days after admission (discharge)", we calculate the value based on admission (discharge) date and assign the type DATE to such expressions.

6. Expressions like "tid" or "t.i.d." etc. are of type FREQ with period of 8 hours. Similar rules are also developed for expressions of type "bid". These expressions have the period of 12 hours.

7. Several timexes start with the letter "q". We developed rules for all such expressions. For example, our rules cover the following expressions: qid (Period: 6 hours), qad (Period: 48 hours), qd (Period: 24 hours), qds (Period: 6 hours), qAM or qPM (Period: 24 hours), qn or qnoc ("every night", Period: 24 hours), qmt ("every month", Period: 1 month), qw ("every week", Period: 1 week), etc. Please note that our rules also cover variations of such expressions as well. Expressions of type "q*n*h" have the period of *n* hours where *n* is a number.

5.2 Experiments and Results

5.2.1 Datasets

For our experiments, we used the data provided by i2b2 team as part of i2b2 2012 shared task. The input consists of plain text files and the output consists of event and timex

	ST	+HT	+Rules
P	1.00	0.84	0.83
R	0.13	0.54	0.76
F1	0.22	0.66	0.79
TYPE	0.13	0.50	0.71
VAL	0.12	0.41	0.56
MOD	0.13	0.49	0.70

Table 5.1: Two tables in part (a) and part (b) show the results for event extraction and timex extraction tasks respectively. P and R in these tables refer to Precision and Recall respectively. In part (b), ST stands for Section Times and HT stands for HeidelTime.

annotations along with their respective attributes. Training data has a total of 190 records and the test data has 120 records.

5.2.2 Timex Extraction

Table 5.1 gives the Precision (P), Recall (R) and F1 scores for the timex extraction task. For evaluating these scores, a predicted temporal expression is considered to be correct if its extent overlaps with that of some gold temporal expression (i.e. attributes of timexes are not taken into account). Fraction of correctly predicted timexes whose attribute (TYPE, VAL or MOD) match the attribute of gold timex is reported separately in last 3 rows headed by the name of attribute (TYPE, VAL or MOD).

In the first column in Table 5.1, we report scores for the case where we only find the section times (ST) i.e. admission and discharge date. Second column reports scores for the case where we also use HeidelTime (HT) in addition to finding section times. And the last column reports the scores for the case when the full system is used. We see that F1 score increases by 0.44 as a result of using HeidelTime. Addition of rules developed by us leads to a further increase of 0.13 in F1 score. Similar improvements can also be seen for TYPE, VAL and MOD attributes.

Chapter 6

A Case Study on Security-related Concepts

6.1 Introduction

While dealing with clinical narratives, there are several privacy concerns. Clinical narratives often contain sensitive information about the patients. In a hospital system, clinical narratives need to be visible to many people so that they can perform their respective functions. Sometimes, it is also necessary to share the clinical narratives among hospital systems. It is important that the privacy of patients should be respected while sharing such information across hospital systems.

There are several types of sensitive data that are found in the clinical narratives. We categorize the sensitive data into 5 major types below:

1. Mental health and abuse in the family
2. Drug Abuse
3. HIV data
4. Genomic data; indication of genetic information in EHRs
5. Sexually transmitted diseases

However in the data that we have, we only found significant number of drug abuse cases. We didn't find sufficient number of cases for other 4 types. So, in this study, we restrict ourselves to the cases of drug abuse.

6.2 Drug Abuse

Wikipedia gives the following definition of drug abuse which is consistent with the definitions of drug abuse found in medical sources like MedlinePlus etc.

> *Substance abuse, also known as drug abuse, is a patterned use of a substance (drug) in which the user consumes the substance in amounts or with methods neither approved nor supervised by medical professionals. Substance abuse/drug abuse is not limited to mood-altering or psycho-active drugs. If an activity is performed using the objects against the rules and policies of the matter (as in steroids for performance enhancement in sports), it is also called substance abuse.*

6.3 Task Description

In this chapter, following 3 things will be addressed:

1. To identify the concepts related to drug abuse.
2. To identify the assertion status (positive or negative) of concepts.
3. To identify whether the concept belonged to the patient.

6.4 Datasets for Experiments

For our experiments, we used the clinical narratives made available by i2b2 team as part of 2011 i2b2/VA coreference challenge. These clinical narratives came from 2 institutions: (a) Partners HealthCare, Boston and (b) Beth Israel Deaconess Medical Center.

Data was annotated by 2 annotators where one of them was a medical expert. Now, we report the results on Inter-Annotator agreement (IAA) on 10 documents. There were a total of 57 concepts related to drug abuse in the data that we selected.

For concept extraction, there was disagreement over 4 cases. So, IAA for concept extraction = 92.9%.For determining assertions, there was disagreement over 3 cases. All the cases of disagreement were related to mild alcohol usage. So, IAA for assertion detection = 94.7%. Finally, we decided that all cases of drug abuse (whether mild or strong) should be annotated to be positive. For determining experiencer of the drug abuse event, there was total agreement. So, IAA = 100.0%.

Since we had very limited data, we decided to use semi-supervised methods for finding drug-abuse events. We reserved all the annotated data for testing.

6.5 Method Description

In the next few subsections, we describe the methodology that we used.

6.5.1 Concept Identification

Concept identification was done using dictionary lookup. We compiled a list of commonly used substances used for drug abuse from web sources. Next, we obtained all the phrases appearing in the clinical narratives using a shallow parser. All those phrases which contained any term located in the drug-abuse dictionary were considered to be drug-abuse events.

6.5.2 Assertion Status

We adapted 3 state-of-the-art expert systems to find the assertion status of the concepts. Below, we describe these three systems in more detail:

Callkit

This is an implementation of the ConText algorithm [153, 154] by Imre Solti. It first of all identifies the trigger words for the negation. Consider the following sentence as an example:

The patient denies any IV drug use but did describe cocaine use for last 2 months.

In the above sentence, *'denies'* is the trigger word for negation. It is important to note that the algorithm differentiates between pseudo-triggers (like 'no increase', 'not cause' etc.) and the actual trigger words.

After determining the triggers, the algorithm determines the scope of the trigger words. The scope of a trigger word generally starts from the word to the right of the trigger and extends till the end of the sentence. But certain termination words (like 'but' in the above example) can cause the scope of a trigger to end early. Also, for certain triggers, the scope lies to the left of the trigger instead of the right. For example, consider the following sentence:

Lung injury was ruled out by the MRI exam.

In the above sentence, the scope of *'was ruled out'* is *'Lung injury'*.

Then if a concept falls within the scope of some trigger word for negation, its scope is changed to negative.

UtahConText

This has similar implementation as that of Callkit. However, it uses slightly different lists of trigger words.

MSRA

Just like ConText algorithm, it also keeps a list of trigger words and identifies the scope of trigger words. However, it addresses the issue that there may be multiple trigger words whose scope may span the concept. To resolve such a thing, it maintains a score for all possible categories. Whenever the concept falls under the scope of some trigger word, it updates the score of the corresponding category. Finally, the category with the maximum score wins. The following scoring formula was used in our implementation. It should be noted that the scoring formula depends on the distance because it is intuitive that when a concept is close to the trigger word, then it is more likely that the trigger word is associated with the concept.

$$x_{category} = \begin{cases} 1 & \text{if } d - w \leq 0 \\ 0.8 & \text{if } d - w = 1 \\ 0.6 & \text{if } d - w = 2 \\ 0.4 & \text{if } d - w = 3 \\ \frac{1}{d-w} & \text{if } d - w \geq 4 \end{cases} \tag{6.1}$$

where window size was chosen to be 3.

6.5.3 Patient or not

All the 3 systems described above give information about the experiencer of the event as well. The mechanism used to identify the experiencer is exactly the same as described for determining the assertion.

P	97.9
R	82.5
F1	89.5

Table 6.1: This table shows the performance of concept extraction for drug-abuse concepts.

	Negation	Experiencer
Callkit	97.9	93.6
Utah	100.0	95.7
MSRA	100.0	95.7

Table 6.2: This table compares the performance of three systems for negation and experiencer detection for drug-abuse concepts.

6.6 Results

Table 6.1 shows the results for concept identification in terms of Precision, Recall and F1 scores. We see from this table that although we achieved very high precision, recall is somewhat low.

Table 6.2 gives the results for assertion and experiencer determination for correctly identified concepts. We find that all systems perform quite well in detecting negation and experiencer. Utah and MSRA performed the best.

6.7 Error Analysis

We can note from the above section on results that our system has a somewhat lower recall for concept identification. This is because of the reason that the list of substances used for drug-abuse that we generated was not comprehensive enough. In our list, we included the commonly used drug-abuse substances. However, the error analysis showed that several other substances are also used for drug-abuse. Some of the concepts that we missed include the following: *codeine*, *morphine sulphate*, *etoh*, *IVDU*, *drug use*,

drunk heavily, *illicit substances* and *pack-year history*.

For negation and experiencer detection, we made mistakes on cases which are particularly difficult. For example, consider the following sentence:

Patient's primary care provider was called to discuss outpatient plans to help the patient stop smoking .

In the above sentence, the phrase '*patient stop smoking*' can mislead the system to predict a negated event. However, when we see the overall context, we can see that the patient is still continuing with his/her smoking habit. Next, consider the following sentence:

He works as a counselor at an alcohol and drug treatment facility for teenagers .

In the above sentence, the word '*alcohol*' can mislead the system to predict a positive drug-abuse event. However, there is no drug-abuse (either positive or negative) being reported here at all.

6.8 Medical Set Expansion

In Section 6.7, we saw that our system has somewhat low recall for concept identification. For concept identification, we have very limited annotated data. This prevents us from developing a supervised learning approach for concept identification.

6.8.1 Semi-Supervised Methods for Concept Identification

In the literature, several semi-supervised methods have been proposed for concept identification. The essential underlying principle behind these semi-supervised methods is that of bootstrapping. In bootstrapping, the input consists of a few examples (also called seeds) of the concept type which we are interested in. Then the system tries to grow the seed set by finding concepts which are similar to the seeds. Distributional context of the concepts generally provides a good way to test the similarity of any two concepts. Bootstrapping approach terminates when the system is unable to grow the seed set further.

For bootstrapping approach to be successful, there should be a lot of instances of the concepts which we are interested in. If this is not the case, then the distributional context of the concepts would be very sparse and thus, insufficient for computing the similarity between two mentions. This is exactly the problem that we face in the datasets that we are experimenting with. These datasets have very few instances of "drug abuse" events, thus, limiting the usefulness of bootstrapping approach.

6.8.2 Active Learning Solution for Concept Identification

Since our datasets have only few instances of relevant concepts, we need to provide some extra level of supervision to our concept identification system. We rely on active learning methods to provide this extra level of supervision. In an active learning based solution, the system asks some questions to the user. The answers provided by the user are used by the system to learn a model for identifying relevant concepts. A good active learning system should ask minimal number of questions from the user.

Moreover, since we lack a good distributional context of the relevant concepts, we use the tree positions of the concepts in a medical encyclopedia named SNOMED CT to find the similarity between mentions.

Level	Concepts	
Level 0	Cocaine,	Cocaine measurement
Level 1	Drug measurement, **Psychostimulant,**	Tropane alkaloid, Ester type local anesthetic
Level 2	Azabicyclo compound, Alkaloid, Measurement of substance, Heterocyclic compound, Psychotherapeutic agent	Local anesthetic, Ester, Stimulant, Tropane alkaloid,
Level 3	CNS drug, Aza compound, Heterocyclic compound, Azabicyclo compound, Drug pseudoallergen by function, Tropane alkaloid,	**Psychoactive substance,** Anesthetic, Measurement, Organic compound, Alkaloid, Psychotherapeutic agent
Level 4	CNS drug, Aza compound, Techniques, Heterocyclic compound, Organic compound, Alkaloid, Psychotherapeutic agent, General drug type	**Psychoactive substance,** Evaluation procedure, Chemical categorized structurally, Azabicyclo compound, Drug pseudoallergen, Tropane alkaloid, Substance categorized functionally,

Table 6.3: This table shows the descriptor for concept "cocaine".

6.8.3 Using SNOMED CT for Medical Set Expansion

Using SNOMED CT, we build a detailed descriptor of every concept. Every concept can appear at multiple places in SNOMED CT. We define the descriptor of a concept to be simply the parents of the concept upto 5 higher levels. We explain it below with the help of an example. Let us consider the concept "cocaine". The descriptor of this concept is shown in Table 6.3. At level 0, two SNOMED CT concepts corresponding to "cocaine" are shown. Concepts at any level $i+1$ are basically the parents of concepts at level i. It is normal for some of the concepts to repeat at later levels. These descriptors were made by a simple breadth-first search on the SNOMED CT graph starting from the concept under consideration.

6.8.4 User Involvement

In this subsection, we describe how the user contributes to the learning of a model for concept identification. To begin with, input to the system consists of a few seeds. Let us represent this seed set by $\mathcal{S}$. Let s_i denote the i^{th} element of seed set. For finding the substances which are potentially used for drug abuse, the input can be the following: "cocaine", "marijuana", "alcohol". Then the system computes the descriptors of each of the concepts and then merges those descriptors into a single descriptor. Let us assume that for concept x, parents at level i are denoted by the set $\mathcal{L}_i(x)$. Then the levels of the overall descriptor are defined by the following equation:

$$\mathcal{L}_i(\mathcal{S}) = \bigcup_{j=1}^{|\mathcal{S}|} \mathcal{L}_i(s_j) \qquad \forall i \tag{6.2}$$

After some preprocessing (like removing overly general concepts), the descriptor is shown to the user. Then the user is supposed to identify one or more most appropriate SNOMED CT concepts from the descriptor. User response is recorded into a list. Let us call this list as $MedRep(\mathcal{S})$. No further input from user is now required.

6.8.5 Computing the Score of a Concept

In this subsection, we describe how to compute the similarity of any given SNOMED CT concept to the seed set, $\mathcal{S}$, provided by the user. Let us denote the given SNOMED CT concept by the variable x. Also, assume that for concept x, parents at level i are denoted by the set $\mathcal{L}_i(x)$. Then the similarity, $sim(x, \mathcal{S})$, of the concept x to the seed set $\mathcal{S}$ is defined by the following equation:

$$sim(x, \mathcal{S}) = \left| \left(\bigcup_{i=1}^{4} \mathcal{L}_i(x) \right) \bigcap MedRep(\mathcal{S}) \right| \tag{6.3}$$

In other words, similarity of a concept to the seed set is the number of *unique* SNOMED

Algorithm 5: MedicalSetExpansion

Input : $\mathcal{S}$ (Seed Set), $\mathcal{D}$ (Document Set)
Output: $\mathcal{R}_{\mathcal{D}}(\mathcal{S})$ (Ranked List of concepts)
begin
1 **for** *every seed* $s \in \mathcal{S}$ **do**
Compute the descriptor of s using Breadth First Search on SNOMED CT graph
2 Compute the overall descriptor of $\mathcal{S}$ by merging the individual descriptors according to Equation (6.2)
3 Display the overall descriptor to user after some pre-processing
4 Record user response in $MedRep(\mathcal{S})$
5 **for** *each noun phrase x in* $\mathcal{D}$ **do**
Compute $sim(x, \mathcal{S})$ according to Equation (6.3)
6 $\mathcal{R}_{\mathcal{D}}(\mathcal{S}) \longleftarrow$ List of NPs sorted by similarity (descending order)

CT concepts in the descriptor of the concept that also appear in the representative model of the seed set given to the system.

6.8.6 Performing Concept Identification

After receiving the user input, the system proceeds to find the relevant concepts from the provided dataset. Relevant concepts are found using the following steps:

1. First of all, we use a chunker to find all the NPs (noun phrases) in the given document.

2. Each of the noun phrases found in Step 1 is mapped to SNOMED CT concepts using a biomedical engine (MetaMap).

3. Then we compute the score of each NP as described in previous subsection (§6.8.5).

4. Finally, the noun phrases are displayed to the user in decreasing order of score.

Algorithm 5 explains the overall algorithm for medical set expansion.

6.9 Focussing on Drug Abuse Events

Using the concept recognition technique described in §6.8.6, it is possible to build a recognizer for any concept type that we may be interested in. For example, one may build a recognizer for finding out the mentions of heart problems. Other examples of recognizers include lung problems, kidney problems, pain-killers, closed surgeries, drug abuse events, sex-related matters, genomic data etc.

In this section, we will focus on the recognizer for drug abuse events. In §6.5.1, we described a recognizer for drug abuse events based on dictionary lookup. §C gives a list of popular drugs that are often used for abuse. This list was compiled from these websites: Wikipedia[1], SAMHSA[2], MedlinePlus[3] and WebMD[4].

In §6.8.6, we described a yet another technique of concept recognition using medical set expansion. In that technique, model for concept identification consists of a list (called as $MedRep(\mathcal{S})$) which basically contains the representatives of the desired concept type in a medical encyclopedia (namely SNOMED CT). §D gives a list of elements contained in $MedRep(\mathcal{S})$ for the concept type "drugs used for substance abuse".

6.9.1 Results

Table 6.4 shows the results for concept identification for the dataset described in §7.10. We see from this table that the recall improved from 82.5 to 89.3 whereas the precision dropped a little. Overall, the F1 score increased from 89.5 to 92.1.

To further test the effectiveness of our system in identifying the substances used for drug abuse, we prepared a dataset using medical forums where people discuss issues related to addiction with drugs. The dataset contained a total of 135 distinct substances

[1]http://en.wikipedia.org/wiki/Substance_abuse
[2]http://www.samhsa.gov/
[3]http://www.nlm.nih.gov/medlineplus/drugabuse.html
[4]http://www.webmd.com/mental-health/substance-abuse

P	95.1
R	89.3
F1	92.1

Table 6.4: This table shows the performance of concept extraction for drug-abuse concepts.

that can be used for drug abuse. Out of these 135 substances, our system could correctly identify 55 substances. Thus, we achieved a recall of 40.7.

6.10 Error Analysis

The above results indicate that our system still misses many drugs that are used for abuse. §E gives a list of drugs that were missed by our system. Below we identify the main reasons for missing such drugs:

1. One primary reason for the low recall was that SNOMED CT does not always have the trademark names for the drugs. For example, *Lorazepam* is a drug that can potentially be abused. Its tradename is *Ativan*. Although, SNOMED CT has an entry for *Lorazepam*, it does not have an entry for *Ativan*. Similar thing happened with the concepts *Percocet*, *Vicodin*, *Darvocet*, *Ritalin* and *Lorcet* which were tradenames for *oxycodone*, *hydrocodone*, *propoxyphene*, *methylphenidate* and *hydrocodone bitartrate* respectively.

2. Another reason for the low recall is that sometimes the drugs are mentioned by their street names which are not present in SNOMED CT. For example, street names for the drug *marijuana* are *ganja*, *grass*, *green*, *Mary Jane* etc. Similarly, street names for the drug *cocaine* are *candy*, *Charlie*, *toot*, *crack* etc.

3. Third reason for the low recall is that SNOMED CT sometimes doesn't have the abbreviations for the drug names. For example, it does not have the abbreviations

LAAM (levacetylmethadol), PCP (phencyclidine) etc.

6.11 Future Work

Following are the good directions for the future work:

1. Wikipedia has a lot of medical knowledge. As discussed above in §6.10, a good amount of knowledge in Wikipedia is not even covered in medical encyclopedias like SNOMED CT. So, it will be a very good project to extract the medical knowledge in Wikipedia and put it in a structured database. For example, Wikipedia can tell the tradenames and common abbreviations for a lot of drugs. Following are the good sources of information in Wikipedia:

   (a) Hyperlinks in free text

   (b) Redirect Pages

   (c) Disambiguation Pages

   (d) Infoboxes

2. Like Wikipedia, there are several other sources of medical information on the web. One very good source for medical information is MedlinePlus. It will be good to extract medical information from it. There is another website, MediLexicon, which gives a lot of useful medical abbreviations.

3. Another good way to get useful medical knowledge is to send automated queries to web search engines. The top pages from the search results can then be used to glean useful medical information. It will be good to design the protocol such that the queries to the search engine are minimized because some search engines block the IP addresses which send too many queries.

6.12 Related Work

Recently, there has been a lot of work centered around Wikipedia. Ratinov et al. [155] analyze local and global approaches for disambiguation to Wikipedia. Yan et al. [156] present an unsupervised relation extraction method for discovering and enhancing relations in which a speciïňĄed concept in Wikipedia participates. Using respective characteristics of Wikipedia articles and Web corpus, they develop a clustering approach based on combinations of patterns: dependency patterns from dependency analysis of texts in Wikipedia, and surface patterns generated from highly redundant information related to the Web. Nguyen and Moschitti [157] extend distant supervision (DS) based on Wikipedia for Relation Extraction (RE) by considering (i) relations deïňĄned in external repositories, e.g. YAGO, and (ii) any subset of Wikipedia documents. They show that training data constituted by sentences containing pairs of named entities in target relations is enough to produce reliable supervision. Wu and Weld [158] present WOE, an open IE system which improves dramatically on TextRunnerâĂŹs [159, 160] precision and recall. The key to WOEâĂŹs performance is a novel form of self-supervised learning for open extractors âĂŤ using heuristic matches between Wikipedia infobox attribute values and corresponding sentences to construct training data.

Conclusion

In this chapter, we presented a study on the detection of drug abuse events in medical text. We explored different state-of-the-art techniques for determining the negation status and experiencer of drug abuse events. For finding the drug abuse concepts, we used an active learning based approach to set expansion. The medical knowledge needed in set-expansion process was obtained from SNOMED CT. We showed that our concept identification technique is able to successfully find even uncommon drugs which

are used for abuse. However, since SNOMED CT does not have tradenames and street names for many concepts, a good direction for future research is to augment the current system with the knowledge from web.

Chapter 7

Coreference Resolution: State-of-the-Art

7.1 Definitions

This chapter addresses the task of coreference resolution for EHRs. *Coreference resolution* is the task of finding referring expressions in a text that refer to the same entity, i.e., finding expressions that corefer. The set of coreferring expressions is called as a *coreference chain*.

Consider the following text sampled from one of the EHRs in the corpus used by us:

> *This 63-year-old man had* ***[malignant fibrous histiocytoma of duodenum]****, discovered in 02/95. Other than* ***[a mass in the duodenum]****, the patient was also diagnosed with anemia. A* ***[leiomyosarcoma]*** *was resected after embolization of the splenic artery. However,* ***[it]*** *could not be completely excised; moreover* ***[the tumor]*** *metastasized to the liver as was discovered on follow up scan in 06/95.*

In the above text, all the phrases which are shown in brackets refer to the same entity and hence form a coreference chain. It is clear that identifying such coreference chains requires a lot of medical knowledge. For example, we need to know that *mass* can refer to a *malignant histiocytoma*. To address this need, we used domain-specific knowledge sources like UMLS etc.

7.2 Previous Work Done

MSRA got the best results in i2b2. LIMSI got the best results in ODIE. However, they have used only very simple models. MSRA used many domain-specific features which give nice results.

There has been an increasing interest in knowledge-rich coreference resolution [161, 162, 163, 36, 164, 165]. Wikipedia is one of the most common knowledge resources that have been used by researchers. However, Wikipedia is not very good for clinical text because it doesn't have sufficient coverage of medical terms and also lacks precision. *In this chapter, we used domain-specific knowledge sources like UMLS, MeSH and SNOMED CT to improve coreference resolution in clinical domain.*

One of the earliest works in coreference resolution in clinical domain is that of Zheng et al. [29]. In this work, authors review recent advances in general purpose coreference resolution to lay the foundation for methodologies in the clinical domain. Later, Zheng et al. [30] describe a simple pairwise classification technique for coreference resolution in clinical domain and got an overall B-cubed score of 0.69 and MUC score of 0.35. Bodnari et al. [23] and Jindal et al. [25] also use a pairwise classification technique for clinical coreference resolution and use UMLS to get some of their semantic features. However, they don't use the concepts' parents information available in UMLS. Uzuner et al. [166] give a brief overview of several systems which participated in 2012 i2b2 coreference challenge. Most of the systems submitted in the challenge were rule-based. Rink et al. [26] used a multi-pass sieve architecture which is similar to the one developed by Raghunathan et al. [28]. Xu et al. [24] developed an effective strategy for pronoun resolution where they first determined the type of the pronoun and then chose the closest preceding concept of the same type as the antecedent. All these works assumed mentions' boundaries (along with their types) to be given just like ours.

7.3 Description of Corpora

Datasets: For our experiments, we used the coreference datasets made available by i2b2 team as part of 2011 i2b2 shared task. The datasets consist of EHRs from two different organizations: Partners HealthCare (`Part`) and Beth Israel Deaconess Medical Center (`Beth`). All records have been fully de-identified and manually annotated for coreference.

The total number of documents in the training set of `Part` and `Beth` are 136 and 115 respectively. Test set of `Part` and `Beth` contains 94 and 79 documents respectively. For more information about the datasets, please refer to Uzuner et al. [166] or Bodnari et al. [23].

7.4 Evaluation Metrics

We used B-cubed [167], MUC [168] and CEAF [169] as the evaluation metrics in our experiments. We also report the unweighted average of F1 scores of these 3 metrics because it was the official metric in i2b2 coreference challenge.

7.5 Task Description

Coreference resolution aims at clustering together textual mentions within a single document based on underlying referent entities. For our experiments, we used the datasets provided by i2b2 team as part of coreference challenge. We address the task of coreference resolution in two different settings as explained below.

In the first setting, we use the same problem definition as was specified in the Task 1C of i2b2 coreference challenge. In this setting, mentions have already been identified and classified into 4 types: test (`TEST`), treatment (`TRE`), problem (`PROB`) and pronoun (`PRON`). Coreference relation can exist only within the mentions of same type. However, `PRON`

mentions can corefer with any other mention. Given the entity mentions along with the types, the aim is to build coreference chains for the first 3 types: `TEST`, `TRE` and `PROB`. Since `PRON` mentions can corefer with the mentions of other types, there are no separate `PRON` chains. In the following, we will use the term *"medical mentions"* to collectively refer to mentions of type `TEST`, `TRE` and `PROB`.

In the second setting, we perform end-to-end coreference resolution for clinical notes. In this setting, the input consists of clinical notes in free-text format and the aim is to build coreference chains for the medical concepts. To perform end-to-end coreference resolution, we first identify mention boundaries and then classify the mentions into 4 types: `TEST`, `TRE`, `PROB` and `PRON`. Then coreference chains are found in a way similar to that of first setting.

In next few sections, we will describe our approach for coreference resolution when the mentions are already given (i.e. according to first setting). In §7.12, we will describe our approach for end-to-end coreference resolution.

7.6 Coreference Model

In this chapter, we view coreference resolution as a graph problem: Given a set of mentions and their context as nodes, generate a set of edges such that any two mentions that belong in the same equivalence class are connected by some path in the graph. We construct this entity-mention graph by finding out the best antecedent of each given mention (anaphor) such that the antecedent belongs to the same equivalence class as the anaphor. For finding the best antecedent for *medical mentions*, we use a variant of *Best-Link* strategy. The *Best-Link* strategy [170, 114, 146] for selecting the antecedent of a mention chooses that candidate as the antecedent which gets the maximum score according to a pairwise coreference function *pc*. We extend the *Best-Link* strategy by including a distance term and several constraints in its objective function as explained

in the next subsection. For finding the best antecedent for *pronominal mentions*, we use a different approach which will be explained in §7.9.

7.6.1 Decision Model: Constrained Best-Link

Given a document d and a pairwise coreference scoring function pc that maps an ordered pair of mentions to a value indicating the probability that they are coreferential, we generate a coreference graph G_d according to the following decision model:

For each mention m_i in document d, let B_{m_i} be the set of mentions appearing before m_i in d. Thus, $B_{m_i} = \{m_1, m_2, ..., m_{i-1}\}$. Let a be the highest scoring antecedent. Then, we have:

$$\begin{aligned} a &= \underset{m_j \in B_{m_i}}{\arg\max} \quad score_i(m_j) \\ &= \underset{m_j \in B_{m_i}}{\arg\max} \quad pc(m_j, m_i) - \frac{1}{k_1} \cdot d(m_j, m_i) + \sum_{l=1}^{L} C_l(m_j, m_i) \end{aligned} \tag{7.1}$$

In the above equation, $d(m_j, m_i)$ refers to the normalized distance between m_j and m_i which takes values between 0 and 1. In equation (7.1), C_l refers to l^{th} constraint and is defined as follows (for all values of l):

$$C_l(m_j, m_i) = \begin{cases} 0 & \text{if } l^{th} \text{ constraint is satisfied} \\ -p_l & \text{otherwise} \end{cases} \tag{7.2}$$

If $score_i(a)$ is greater than a threshold δ, then we add the edge (a, m_i) to the coreference graph G_d. Threshold parameter δ is chosen to be 0.5. Value of $pc(m_j, m_i)$ lies between 0 and 1. The value of k_1 is chosen to be sufficiently greater than 1 so that the pairwise classifier is given preference over the distance term in choosing the best antecedent. But if the pc values of any two candidates are almost similar, then the antecedent which is closer to the anaphor gets the higher score because of the distance term in Equation (7.1).

Thus, our decision model combines the advantages of both "best-link" and "closest-first" models which are generally used for coreference resolution. Setting $k_1 = \infty$ and $L = 0$ reduces our model to the standard "best-link" decision model.

p_l is the penalty associated with the l^{th} constraint. Thus, different constraints can have different penalties. Higher the penalty associated with the constraint, the stronger it is enforced. If $0 < p_l < 0.5$, then the constraint is soft because violation of such constraint by a mention pair doesn't necessarily rule it out. But if $p_l > 0.5$, then the constraint becomes hard.

The resulting graph produced by the decoding technique mentioned above contains connected components, each representing one equivalence class, with all the mentions in the component referring to the same entity. Equivalence classes are determined by taking the transitive closure of all the links.

7.6.2 Pairwise Coreference Function

We train 3 classifiers, one each for `TEST`, `TRE` and `PROB` classes. Each of these classifiers takes as input an ordered pair of mentions (a, m) such that a precedes m in the document, and produces as output a value that is interpreted as the conditional probability that a and m belong in the same equivalence class. Selection of positive and negative examples for training the classifiers is done in a way that is similar to that of Bengtson and Roth [114].

7.7 Description of Features

In this section, we describe the features used by pairwise classifiers. We divide the features into two main categories as described in the following two subsections.

7.7.1 Baseline Features

Baseline features refer to those features which are typically used for coreference resolution. These features are subdivided into following 3 categories.

Lexical Features Similar to Bengtson and Roth [114], we used the following lexical features: (a) Exact (or extent) match, (b) Substring relation and (c) Head match.

Syntactic Features For syntactic features, we used *Apposition* and *Predicate Nominative* as described by Raghunathan et al. [28].

Semantic Features Similar to Bengtson and Roth, we used WordNet to check whether given mentions are synonyms or hypernyms of one another.

7.7.2 Features Using Domain-Specific Knowledge

In medical terminology, same concept can be represented in several different ways. For example, *headache*, *cranial pain* and *cephalgia* all refer to the same concept. Similarly, *Atrial Fibrillation*, *AF* and *AFib* also refer to the same concept. The baseline features are not sufficient to address the ambiguity and variability that exists in medical terminology. To improve the performance of coreference resolution, we used several types of domain-specific knowledge which is explained below. Importance of using knowledge has been emphasized in other domains as well [162, 163, 171].

Expanding the abbreviations Clinical narratives use a lot of abbreviations. A few examples are: *MRI* (Magnetic Resonance Imaging), *COPD* (Chronic Obstructive Pulmonary Disease) etc. Abbreviations were expanded to their full forms as a normalization step. We collected abbreviations from several sources like training data, Wikipedia[1]

[1]http://en.wikipedia.org/wiki/List_of_medical_abbreviations

etc. For ambiguous abbreviations, we considered all possible expansions.

Converting Hyponyms to Hypernyms During preprocessing, we converted some of the common hyponyms to the corresponding hypernyms. Examples of such conversions are: chemotherapy → therapy, hemicolectomy → colectomy. Such conversions are quite helpful because it is a common practice in clinical documents to refer to some of the problems and treatments introduced earlier in the document with their more general names later on. These hyponym-hypernym pairs were collected from the training data.

Mapping to Biomedical Vocabularies We used MetaMap [6] and MetamorphoSys tools to map the mentions to concepts in biomedical vocabularies like UMLS[2], MeSH[3] and SNOMED CT[4]. Such mapping helps us to determine whether any two mentions are equivalent or not. For example, *cancer* and *malignancy* both map to same UMLS concept namely *Primary Malignant Neoplasm*. From such mapping, we can infer that *cancer* and *malignancy* can be coreferential to one another even though they are lexically quite different.

7.8 Description of Constraints

Although our model allows for both hard and soft constraints, we used only hard constraints in the current work. These constraints allow us to override the decision of pairwise classifier, where appropriate. Following is a list of constraints that we used.

- *Length Constraint*: Surface form of both the mentions must be at least 2 characters long.

[2]http://www.nlm.nih.gov/research/umls/
[3]http://www.nlm.nih.gov/mesh/meshhome.html
[4]http://www.ihtsdo.org/snomed-ct/

- *Body Parts Constraint*: If body parts (like chest, arm, head) are specified, they should not be incompatible.

- *Anatomical Terms Constraint*: If anatomical terms[5] (like proximal, anterior, dorsal) are specified, they should not be incompatible.

- *Temporal Constraint*: Certain words like *follow-up* or *repeat* convey the temporal information about the mentions. For example, the word *repeat* in the mention *repeat chest x-ray* indicates that *chest x-ray* is being done for the second time. If two mentions refer to tests or treatments which were done at different times, then they can't be coreferential.

- *Section Constraint*: Clinical reports often specify different sections like *History of Present Illness*, *Laboratory Data*, *Medications on Discharge* etc. We developed an algorithm for finding and normalizing the section headings. If a mention appears in either *Family History* section or *Social History* section in a clinical report, we don't consider it for coreference. This is because such mentions generally describe the problems associated with family members of the patient and not the patient himself/herself.

- *Value Constraint*: `TEST` mentions generally have a value associated with them. If any two `TEST` mentions don't have the same value, then they can't be coreferential.

- *Assertion Constraint*: We implemented an algorithm for finding the assertion status (like *present*, *absent* etc.) of `PROB` mentions as described by Xu et al. [137]. Two mentions can't be coreferential if they don't have the same assertion status.

[5]http://en.wikipedia.org/wiki/Anatomical_terms_of_location

7.9 Pronominal Coreference Resolution

In the datasets that we worked with, pronominal resolution is primarily limited to 4 types of pronouns: (1) which (2) that (3) this and (4) it. Other pronouns like these, those, whichever etc. hardly participate in coreference relation in our datasets. Also, personal pronouns like he, she, him, you, yourself etc. refer to persons and hence are not relevant to us because we are interested in forming coreference chains for only medical mentions (`TEST`, `TRE` and `PROB`).

Features commonly used for pronominal resolution [28, 172] include distance, number agreement, gender agreement, entity type, grammatical person (first, second and third) etc. However, many of these features are not very helpful in our case. For example, all the medical mentions have neuter gender. So, *gender agreement* is not helpful. Similarly, *grammatical person feature* is also not helpful because it is relevant only for personal pronouns. It should also to be noted that researchers [28, 172] commonly use the same technique for resolving different types of pronouns. However, in our experiments, we found that different pronouns behave very differently and therefore, we designed separate modules for finding the antecedent for different types of pronouns. Next two subsections describe our overall strategy for pronominal resolution.

7.9.1 Determining Anaphoricity

First of all, we determine whether the given pronoun is anaphoric or not. Ng and Cardie [173] have previously shown the benefits of predicting anaphoricity. To identify non-referential cases for pronoun *it*, we implemented the heuristics mentioned by Paice and Husk [174]. To determine the anaphoricity for the remaining pronouns (*this*, *that* and *which*), we learned a classifier with the following features: (a) Pronoun under consideration (*this*, *that* or *which*), (b) Part-of-Speech tag of pronoun and (c) Number of tokens in the immediate noun phrase encompassing the pronoun.

7.9.2 Finding the Antecedent

In the previous step, we filtered out the pronouns which were non-referential. For the remaining pronouns, we need to find the best antecedent. Depending on the pronoun under consideration, we used different techniques for finding the antecedent as described below.

which and that Referential cases of pronouns *which* and *that* behave quite similarly. So, we use the same strategy for determining their antecedents. Both these pronouns (*which* and *that*) are often used as a relative pronoun and they mark the beginning of a dependent clause. We select the closest medical mention in the associated independent clause as the antecedent for such pronouns. However, if there is any intervening noun phrase between the pronoun and the closest medical mention, then we leave such a pronoun as a singleton and mark its antecedent as `NULL`. It should be clear from the above description that we restrict the antecedent of pronouns *which* and *that* to come from the same sentence.

this and it For pronouns *which* and *that*, we could simply select the closest medical mention (subject to some constraints) as the antecedent. However, the antecedent of pronouns *this* and *it* can be separated from them by one or more medical mentions. Thus, antecedent of these pronouns (*this* and *it*) is not necessarily in the same sentence.

To determine the antecedent of pronouns *this* and *it*, we trained an SVM classifier to identify whether pronoun under consideration is being used as a test, treatment or problem. Thus, this classifier has 3 possible outputs: `TEST`, `TRE` or `PROB`. Following features were used for training this classifier: (a) Pronoun under consideration (*this* or *it*), (b) Verb in the associated clause, (c) Is pronoun acting as a subject or an object, (d) Is there a preposition in the path from pronoun to its associated verb and (e) Part-of-Speech of pronoun.

Finally, we selected the closest medical mention which satisfied the following criteria as the antecedent for pronouns *this* and *it*:

1. Antecedent should either be in the preceding sentence or if it is in the same sentence, it should be separated from pronoun by some conjunction (like and, but, although etc.).

2. Antecedent should have the same type (`TEST`, `TRE` or `PROB`) as the pronoun (as given by SVM classifier).

7.10 Experimental Setup

Datasets: For our experiments, we used the coreference datasets made available by i2b2 team as part of 2011 i2b2 shared task. The datasets consist of EHRs from two different organizations: Partners HealthCare (`Part`) and Beth Israel Deaconess Medical Center (`Beth`). All records have been fully de-identified and manually annotated for coreference.

The total number of documents in the training set of `Part` and `Beth` are 136 and 115 respectively. Test set of `Part` and `Beth` contains 94 and 79 documents respectively. For more information about the datasets, please refer to Uzuner et al. [166] or Bodnari et al. [23]. We used B-cubed [167], MUC [168] and CEAF [169] as the evaluation metrics in our experiments. We also report the unweighted average of F1 scores of these 3 metrics because it was the official metric in i2b2 coreference challenge.

Choice of Parameters: We use cross-validation on the training data to determine the system parameters. In Equation (7.1), we set $k_1 = 100$. With this choice of k_1, distance term becomes significant only if the scores given by pairwise classifier for different mention pairs differ by less than 0.01. As far as constraints are concerned, we decided to formulate all our constraints as hard constraints. As pointed out before, any value of

	B			BK			BKP			BKPC		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Test (TEST)												
MUC	29.7	52.8	38.0	-	-	-	39.0	82.7	53.0	57.8	66.0	**61.6**
B3	94.4	96.8	95.6	-	-	-	92.7	97.6	95.1	96.2	96.7	**96.4**
CEAF	81.7	93.8	87.3	-	-	-	82.4	94.6	88.1	93.1	94.9	**94.0**
Avg			73.6						78.7			**84.0**
Treatment (TRE)												
MUC	74.4	76.2	75.3	-	-	-	73.0	79.9	**76.3**	73.0	79.9	**76.3**
B3	95.9	95.9	**95.9**	-	-	-	94.7	96.2	95.4	94.7	96.2	95.4
CEAF	86.6	89.4	88.0	-	-	-	86.7	89.5	**88.1**	86.7	89.5	**88.1**
Avg			86.4						**86.6**			**86.6**
Problem (PROB)												
MUC	72.8	66.4	69.5	69.7	73.5	71.6	69.9	81.2	75.1	74.9	76.8	**75.8**
B3	96.6	94.8	**95.7**	95.4	95.8	95.6	93.8	96.3	95.0	95.1	95.7	95.4
CEAF	87.4	87.9	87.7	84.9	90.1	87.4	85.3	90.5	87.8	89.0	89.8	**89.4**
Avg			84.3			84.9			86.0			**86.9**

Table 7.1: This table compares the performance of four systems: *B*, *BK*, *BKP* and *BKPC* on Part dataset. Average F1 scores in this table show that the performance of coreference resolution is significantly improved by adding knowledge, pronominal resolution and constraints to the system. For detailed discussion, please refer to §7.11.

$p_l > 0.5$ makes l^{th} constraint hard. To formulate all our constraints as hard constraints, we chose $p_l = 1.0$ in Equation (7.2) for all values of l.

7.11 Results

Table 7.1 compares the performance of four systems as described below:

1. *Baseline (B)*: Baseline system uses only the baseline features described in §7.7.1. It doesn't perform pronominal resolution. Also, it doesn't use any constraints.

2. *Baseline + Knowledge (BK)*: This system uses all the features described in §7.7. In other aspects, it is similar to *Baseline* system.

3. *Baseline + Knowledge + Pronouns (BKP)*: This system adds pronominal resolution to *BK* system.

4. *Baseline + Knowledge + Pronouns + Constraints (BKPC)*: This is the final system. It adds the ability to deal with constraints to the *BKP* system.

In Table 7.1, we compare the performance of these 4 systems for `TEST`, `TRE` and `PROB` categories on `Part` corpus. We don't show the detailed results for `Beth` corpus because of space limitations. But it follows a very similar trend. Table 7.1 reports precision (P), recall (R) and F1 scores for MUC, B-cubed and CEAF evaluation metrics. It also shows the average F1 score of these three metrics. Please note that there are no separate scores for `PRON` category because there are no separate `PRON` chains. `PRON` mentions are included within the `TEST`, `TRE` and `PROB` chains.

It is interesting to note that adding knowledge to the system always leads to higher recall values. On the other hand, addition of constraints always leads to higher precision values. Next, we note that different metrics behave differently in evaluating the performance of the systems. It can be seen that B-cubed metric gives the highest scores. Even for *Baseline* system, B-cubed metric gives about 95% F1 score. This is because of the fact that the corpora used by us contain a very large number of singletons. B-cubed metric gives very high scores because it highly awards the correct prediction of singletons. MUC, on the other hand, is totally insensitive to singletons. CEAF is intermediate between B-cubed and MUC as far as singletons are concerned. From this discussion, we can see that B-cubed metric is not very discriminative for our corpora. But MUC and CEAF are quite good for comparing the performance of different systems. Average F1 score shown in Table 7.1 is the official metric used in i2b2 shared task and is a good indicator of the performance of the system. Next, we note the following major points about each category of mentions.

Test: We don't use the features derived from domain-specific knowledge sources for TEST mentions. This is because of the fact that coreferring mentions of TEST type tend to have similar surface forms. So, knowledge-based features are not helpful for TEST mentions. In Table 7.1, we see from average F1 score that constraints and pronominal resolution are very helpful for TEST mentions. Average F1 score jumps from 73.6 to 78.7 on adding pronouns to baseline system. On further addition of constraints, average F1 jumps from 78.7 to 84.0 (an increase of 5.3 F1 points).

Treatment: Just as in the case for TEST mentions, we don't use features derived from domain-specific knowledge sources for TRE mentions as well. Average F1 score shows that pronominal resolution gives small improvement of 0.2 F1 points for TRE mentions.

Problem: For PROB mentions, both knowledge and constraints were used. Average F1 scores in Table 7.1 show that PROB mentions benefit significantly from knowledge, pronouns and constraints. Average F1 score goes from 84.3 to 84.9 on adding knowledge to baseline system. It further increases to 86.0 and then to 86.9 on adding pronominal resolution and constraints respectively.

The improvements obtained by adding knowledge, pronominal resolution and constraints shown in Table 7.1 are statistically significant at $p = 0.05$ according to Bootstrap Resampling Test [148]. The only exception to this is the TRE category which didn't get significant improvement by the addition of constraints.

Finally, in Table 7.2, we compare our system with several other state-of-the-art systems for coreference resolution in medical domain. The numbers reported in Table 7.2 refer to the unweighted average of B-cubed, MUC and CEAF F1 scores. We chose unweighted average for comparison because it was the official metric of i2b2 2011 shared task on coreference. For both Part and Beth corpora, our system outperformed all other systems. Xu et al. [24] got the highest scores in i2b2 2011 shared task on coreference.

	Avg of B^3, MUC, CEAF F1		
	TEST	TRE	PROB
	Part Corpus		
Xu et al.	82.6	85.7	86.8
Jindal & Roth	76.1	84.4	84.0
Dai et al.	79.7	81.6	80.5
Gooch & Roudsari	80.5	84.3	83.5
This Chapter	**84.0**	**86.6**	**86.9**
	Beth Corpus		
Xu et al.	78.0	83.9	**86.8**
Jindal & Roth	65.5	83.0	84.0
Dai et al.	75.6	80.2	79.7
Gooch & Roudsari	78.4	81.7	81.5
This Chapter	**79.2**	**84.4**	**86.8**

Table 7.2: This table compares our final system with several other state-of-the-art systems on both Part and Beth corpora. For both these corpora, our system outperformed all other systems. *Thus, we report the best results on shared task corpora.*

We can see from Table 7.2 that for TEST and TRE categories, we got significantly higher scores than Xu et al. for both Part and Beth corpora. In particular, we improved over Xu et al.'s score by 1.3 and 0.7 F1 points respectively for TEST and TRE categories (when averaged over both Part and Beth corpora). This improvement is statistically significant at $p = 0.05$ according to Bootstrap Resampling Test. For PROB category, we got an improvement of 0.1 F1 points for Part corpus. But it is not statistically significant. For PROB category in Beth corpus, our score is similar to that of Xu et al. As far as other systems [25, 175, 22] are concerned, our scores are much higher than theirs for all mention categories and all the differences are statistically significant at $p = 0.05$. *Thus, we report the best results on the i2b2 shared task corpora according to the best of our knowledge.*

7.11.1 Detailed Pronominal Resolution Analysis

In Table 7.3, we show the performance improvement corresponding to each pronoun individually for Partners corpus. The first column in this table shows the performance

	BK	BK+which	BK+this	BK+that	BK+it
Test (TEST)					
MUC	38.0	49.6	38.9	41.3	38.6
B3	95.6	95.2	95.4	95.4	95.5
CEAF	87.3	87.9	87.3	87.5	87.4
Avg	73.7	77.5	73.8	74.7	73.8
Treatment (TRE)					
MUC	76.2	77.6	75.7	76.3	76.0
B3	95.8	95.5	95.6	95.8	95.8
CEAF	87.4	87.6	87.3	87.5	87.4
Avg	86.5	86.9	86.2	86.5	86.4
Problem (PROB)					
MUC	71.6	73.8	72.0	72.6	71.8
B3	95.6	95.3	95.4	95.5	95.5
CEAF	87.4	87.8	87.4	87.5	87.4
Avg	84.9	85.6	84.9	85.2	84.9
Overall (OVERALL)					
MUC	68.7	71.6	68.7	69.6	68.8
B3	96.3	96.5	96.2	96.3	96.2
CEAF	87.1	87.8	87.1	87.3	87.1
Avg	84.0	85.3	84.0	84.4	84.1

Table 7.3: This table shows the F1 scores in all the metrics for each of the pronouns individually.

of the BK system. Then next 4 columns show the performance of BK system as the capability to resolve one of the pronouns (which, this, that or it) was added to it. We see from this table that different pronouns give different performance improvements. Pronoun 'which' gives the maximum performance improvement of 1.3 F1 points. 'which' is followed by 'that' which gives a performance improvement of 0.4 F1 points. Pronoun 'it' gives only a small improvement of 0.1 F1 points and pronoun 'this' did not give any noticeable improvement. It is also interesting to note that none of the pronouns lead to a degradation in the performance.

In Table 7.4, we show the cumulative performance of the BK system as the ability to resolve different pronouns (which, this, that and it) is added to it. The results shown in this table are quite consistent with the results shown in Table 7.3. We see that addition

	BK	BK+which	BK+which+this	BK+which+this+that	BK+All
Test (TEST)					
MUC	38.0	49.6	50.1	52.7	53.2
B3	95.6	95.2	95.1	95.0	94.9
CEAF	87.3	87.9	87.8	87.9	87.9
Avg	73.7	77.5	77.6	78.5	78.7
Treatment (TRE)					
MUC	76.2	77.6	77.0	77.1	76.9
B3	95.8	95.5	95.3	95.3	95.2
CEAF	87.4	87.6	87.5	87.5	87.5
Avg	86.5	86.9	86.6	86.6	86.5
Problem (PROB)					
MUC	71.6	73.8	74.1	75.0	75.2
B3	95.6	95.3	95.1	95.1	95.0
CEAF	87.4	87.8	87.7	87.8	87.8
Avg	84.9	85.6	85.7	86.0	86.0
Overall (OVERALL)					
MUC	68.7	71.6	71.6	72.4	72.5
B3	96.3	96.5	96.4	96.5	96.5
CEAF	87.1	87.8	87.8	88.1	88.1
Avg	84.0	85.3	85.3	85.7	85.7

Table 7.4: This table shows the F1 scores in all the metrics for pronouns collectively.

of pronouns 'which' and 'that' gives the performance improvement of 1.3 and 0.4 F1 points respectively. Addition of pronouns 'this' and 'it' did not give any noticeable performance improvements.

7.12 End-to-End Coreference Resolution

In this section, we would describe our approach for end-to-end coreference resolution. To perform end-to-end coreference resolution, we first identify mention boundaries along with mention types. We used a CRF model [84] to perform mention detection. CRF model used BIO encoding for representing chunks and was implemented using MALLET toolkit [18]. Features used by CRF model include surface forms of words, part-of-speech labels, shallow parse labels and features derived from MetaMap. We also

	Part Corpus			Beth Corpus		
	P	**R**	**F1**	**P**	**R**	**F1**
Test (TEST)						
MUC	48.5	50.8	49.6	31.4	38.0	34.4
B3	95.8	96.2	96.0	96.2	97.0	96.6
CEAF	94.1	93.1	93.6	93.3	92.4	92.9
Avg			79.7			74.6
Treatment (TRE)						
MUC	59.2	63.3	61.2	58.1	58.9	58.5
B3	91.7	93.8	92.7	92.0	92.6	92.3
CEAF	87.4	81.8	84.5	83.8	78.5	81.1
Avg			79.5			77.3
Problem (PROB)						
MUC	62.8	56.8	59.7	61.4	57.4	59.4
B3	93.8	93.5	93.6	92.4	92.5	92.4
CEAF	90.5	82.2	86.2	88.8	78.8	83.5
Avg			79.8			78.4

Table 7.5: This table shows the performance of our final system for end-to-end coreference resolution. For detailed discussion, please refer to §7.12.

used conjunction of these features. Once we have the mentions along with their types, we perform coreference resolution in the same way as described in §7.6 to §7.9.

For evaluation of end-to-end coreference resolution, we used the script provided by i2b2 2011 challenge organizers. Table 9.6 shows the performance of our final system for end-to-end coreference resolution. It reports precision (P), recall (R) and F1 scores for MUC, B-cubed and CEAF evaluation metrics. It also shows the average F1 score of these 3 metrics. This table shows the results for TEST, TRE and PROB categories on both Part and Beth corpora. As far as we know, *end-to-end results have not been reported previously on both these corpora*. By comparing the average F1 scores in Tables 7.1 and 9.6, we notice that the scores of our final system are about 5-8% lower for end-to-end task than the task where gold mentions were given. The decrease in performance is because of errors made in mention detection. However, it is very encouraging to see that the performance on end-to-end task is still quite high. For example, on Part dataset, average F1 score is

higher than 79% for all the categories (TEST, TRE and PROB). This is much higher than the best result of 63.4% F1 in CoNLL 2012 shared task on coreference [176]. Zheng et al. [30] performed end-to-end coreference resolution on ODIE corpus. However, their average F1 score is quite low (50.9%).

Chapter 8

Coreference Resolution for Persons

8.1 Coreference Resolution

It can be for medical concepts and for persons. Here we will only consider persons since they are quite different. Next chapter will describe coreference resolution for medical concepts.

8.2 Discourse Model: Patient, Doctors and Family Members

We employ a domain-inspired discourse model for generating coreference chains for the class PER. Our discourse model can be specified as: One patient, several doctors and a few family members. The development of this model was based on the observation that clinical narratives only discuss a single patient. Other than the patient, multiple doctors are mentioned in the narrative, including the attending physician, doctors who are consulted or who have previously treated the patient or whom the patient will next be visiting, etc. Other than the patient and doctors, the clinical narratives sometimes mention a few family members like father, husband, wife, etc.

Employing an appropriate discourse model simplifies the process of coreference resolution significantly. The discourse model specified above readily yields a 2-layer algorithm for coreference resolution which is described below.

PatientsâĂŹ Contexts	DoctorsâĂŹ Contexts
[patient] is a 61 year old male with a history of ...	He was seen by [doctor].
[patient] was diagnosed recently with pancreatic cancer after he ...	She will follow up with her pcp , [doctor] , at IVMC , after her discharge .
[patient] was admitted to the Retelk County Medical Center at that time and was treated with ...	cc : [doctor1] [doctor2] ...
DISCHARGE SUMMARY NAME : [patient]	She has been under the care of [doctor]

Table 8.1: This table shows the common contexts in which the mentions corresponding to patients and doctors appear.

8.3 2-Layer Algorithm for Coreference Resolution

We employ a 2-layer algorithm for determining the PER coreference chains. In the first layer, we divide the PER mentions into 3 categories: (1) mentions corresponding to patient, (2) mentions corresponding to any of the doctors and (3) the rest of mentions. The coreference pairs are generated in the second layer from within the categories obtained in the first layer since we know that coreference pairs do not cross the categories. We describe the 2 layers in detail below.

8.3.1 Design of the First Layer

We divide all the PER mentions into three categories (namely patient, doctors and the rest) based on the following criteria:

1. Surface Form of the Mention: A mention is added to the list of doctors if it has the tokens like “dr.”, “m.d.”, “cardiologist”, etc. Similarly, the mentions like “the patient”, “this patient”, etc. were added to the patient list.

2. Context: Table 8.1 shows common contexts in which patients and doctors appear.

Context	Assigned List
This is to notify you that your patient , AGACH , arrived in the Emergency Department ...	Doctor
Please call your primary care doctor for follow up next week .	Patient
If you have further chest pain , call your doctor .	Patient

Table 8.2: Second person pronoun can either refer to doctor or patient depending on the context.

The mentions which appear in such common contexts were added to the appropriate list.

3. Similarity: We consider two mentions to be similar if the surface forms of the mentions have at least 1 token in common. Note that the common token can't be a person title like "mr.", "mrs.", etc. or a doctor title like "dr.", "m.d.", etc. If one of the mentions among a set of similar mentions has already been classified as belonging to the doctor list or the patient list, then all other mentions in the set are also assigned to the same list.

4. All other mentions, with the exception of pronouns, are put in a separate list. Such mentions generally refer to the patients' family members (e.g., "his father", "his wife").

The personal pronouns are categorized in the three lists (patient, doctors, rest) based on the following criteria:

1. The first person pronouns like I, me, my etc. are added to the doctor list. This is because a clinical narrative is generally dictated by a physician or physician's assistant.

2. The second person pronouns are added to the patient list or the doctor list based on the context in which they appear. See Table 8.2 for examples. If the context is

Sentence	Role	Doctor
[His primary care physician] is [Dr. **NAME[ZZZ]].	primary care physician	Dr. **NAME[ZZZ]
[PCP] Name : [WHITE , ELVNO R]	PCP	WHITE , ELVNO R
She was seen by [her cardiologist] , [Dr. Clements] and had a Holter monitor on 2015-05-01 .	cardiologist	Dr. Clements

Table 8.3: This table shows a few example sentences where the doctors participate in some role.

not very clear, then the pronoun is assigned to the same list as any other second person pronoun in the vicinity for which the context is clear.

3. The third person pronouns like he, his, etc. are added to the patient list. This heuristic was found to be quite precise. The third person pronouns very rarely refer to doctors.

4. Pronouns like âĂIJwhoâĂİ are added to the doctor list only if there is some doctor mention preceding the pronoun within a margin of 2 words. Otherwise, the pronoun is added to the patient list.

8.3.2 Design of the second layer

In this layer, we generate the actual coreference pairs as explained below:

1. Since our model assumes only one patient, all the mentions in the patient list are assigned to the same coreference chain.

2. From among the list of doctors, we generate a coreference pair between two mentions if any of the following two conditions are met:

Metric	Precision	Recall	F1
MUC	0.956	0.962	95.9
Bcubed	0.954	0.924	93.9
CEAF	0.842	0.834	83.8
Avg	91.7	90.7	91.2

Table 8.4: This table shows the performance on PARTNERS corpus.

Metric	Precision	Recall	F1
MUC	0.950	0.953	95.1
Bcubed	0.950	0.935	94.2
CEAF	0.843	0.820	83.1
Avg	91.4	90.2	90.8

Table 8.5: This table shows the performance on BETH corpus.

(a) Lexical Match: The two mentions share at least one similar token (with the exception of person and doctor titles).

(b) Role Participation: The two mentions are separated by not more than 2 words and the first mention specifies some role like physician, pcp, cardiologist etc. and the second mention doesn't specify any such role (See Table 8.3 for examples)

3. The second person and first person pronouns (if any) in the doctor list are assigned to separate coreference chains.

4. For the rest of mentions, the coreference pairs are generated according to the lexical match condition.

8.4 Results

Tables 8.4 and 8.5 show the performance of coreference resolution on PARTNERS and BETH corpora respectively.

Chapter 9

Joint Approach for Coreference Resolution

9.1 Introduction

This chapter presents the method for end-to-end coreference resolution for clinical narratives. End-to-end coreference resolution involves determining the mentions and also the coreference relations between them. Typically, a pipeline approach is used for end-to-end coreference resolution where the mentions are first determined and then the coreference relations are found among them. Named entity types and other attributes of mentions are generally used while determining the coreference relations among them. Such an approach has limitations because there may be some errors in the first phase where the attributes of the mentions are determined. These errors are propagated to the next stages and it is not possible to correct such errors later on. To overcome this problem, we present a flexible architecture in this chapter which doesnâĂŹt make hard decisions on mention types while performing mention detection. Instead a joint inference procedure makes the final decisions.

Another major contribution of this chapter is in pronominal resolution. Quite often, we find in coreference resolution literature that researchers use the same model for resolving all kinds of pronouns. We, however, found that different pronouns behave quite differently. So, we developed separate modules for finding the antecedents of different kinds of pronouns. The method used by us for pronominal resolution is quite general and will be useful for coreference resolution on other types of text as well.

We tested our approach on the data that was made available by i2b2/VA team in 2011

shared task on coreference resolution. Some of this data (say, $data^{ODIE}$) was annotated according to ODIE guidelines and the rest of the data (say, $data^{i2b2}$) was annotated according to i2b2 guidelines. The shared task involved end-to-end coreference resolution for $data^{ODIE}$. However, for $data^{i2b2}$, gold mentions were already given and the task was to find only the coreference chains.

Using our approach for end-to-end coreference resolution, we got the best results on $data^{ODIE}$. Also, for the first time, we report the results for end-to-end coreference resolution on $data^{i2b2}$. We also report the best results on both $data^{ODIE}$ and $data^{i2b2}$ for the case where gold mentions are already given.

9.2 Background and Significance

Coreference resolution is a very important task to understand the semantics of the text and to extract meaningful information from it. i2b2/VA organized a challenge on coreference resolution for clinical narratives in 2011 [166]. A lot of teams from around the world participated in the challenge. Most of the teams focused on the task where gold mentions (along with types) were already given and the aim was to simply recognize the coreference chains. Specification of the mentions along with their types simplifies the problem of coreference resolution considerably. However, for the real-world applications, what we really want is the capacity for end-to-end coreference resolution. Therefore, in this chapter, we focus on end-to-end coreference resolution.

In 2011 i2b2/VA challenge, three teams participated in end-to-end coreference resolution. Cai et al. [177] proposed a weakly supervised algorithm which performs classification and clustering steps together with the help of a global inference procedure. Their inference procedure uses mention types as one of the features. These mention types are still determined in a pipeline fashion. Both Lan et al. [178] and Grouin et al. [179] used rule-based systems to find the coreferential pairs where the mention types were used

in a pipeline fashion. Named-entity types have been shown to be important features for coreference resolution in the news domain also. But there also, researchers primarily take to pipeline approach. The well-known problem with pipeline based systems is that of error-propagation i.e., the errors made in earlier stages get passed on to the later stages.

First of all, Pascal and Baldridge [180] proposed to model coreference relations jointly with named entity types. However, they used the hard constraint that all the mentions in a coreference chain must have the same type. Considering the fact that named entity tagger may not give perfect distributions, this constraint is too restrictive. Therefore, in this chapter, we soften this constraint by introducing a penalty parameter which determines the degree to which this constraint is enforced.

Features commonly used for pronominal resolution [6,7] include distance, number agreement, gender agreement, entity type, grammatical person (first, second and third) etc. However, many of these features are not very helpful in our case. For example, all the medical mentions have neuter gender. So, gender agreement is not helpful. Similarly, grammatical person feature is also not helpful because it is relevant only for personal pronouns. It should also to be noted that researchers [6-9] commonly use the same technique for resolving different types of pronouns. However, in our experiments, we found that different pronouns behave very differently and therefore, we designed separate modules for finding the antecedent for different types of pronouns.

9.3 Materials Used

Coreference resolution aims at clustering together textual mentions within a single document based on underlying referent entities. For our experiments, we used the datasets provided by i2b2 team as part of coreference challenge. The data consists of three types of text files: (1) '*.txt' files contain the plain clinical narratives, (2) '*.con' files contain the

concepts found in the corresponding .txt files and (3) '*.chain' files contain the coreference chains.

The data provided in the challenge came from three different institutions: (1) Partners, (2) Beth and (3) Mayo. The data from Mayo institution was annotated according to ODIE guidelines [181] whereas the data from other two institutions was annotated according to i2b2 guidelines. We describe the characteristics of both ODIE and i2b2 data below in more detail. All records have been fully de-identified and manually annotated for coreference.

1. ODIE: ODIE annotation specifies the following types of mentions: "people", "procedure", "diseaseorsyndrome", "signorsymptom", "anatomicalsite", "laboratoryortestresult", "indicatorreagentdiagnosticaid", "organortissuefunction", "none" and "other". Mayo data has 2 types of reports: 'clinical' and 'pathology'. The training set contains 28 and 30 documents respectively of 'clinical' and 'pathology' reports. The test set contains 19 and 20 documents respectively of 'clinical' and 'pathology' reports.

2. I2b2: i2b2 annotation specifies the following types of mentions: "problem", "test", "treatment", "person" and "pronoun". The total number of documents in the training set of Part and Beth are 136 and 115 respectively. Test set of Part and Beth contains 94 and 79 documents respectively. For more information about the datasets, please refer to Uzuner et al. [1] or Bodnari et al. [11].

9.4 New Method

Finally, we solve a joint inference.

The resulting graph produced by the decoding technique mentioned above contains connected components, each representing one equivalence class, with all the mentions in the component referring to the same entity. Equivalence classes are determined by taking the transitive closure of all the links.

Assume that there are N mentions in a document. Also, assume that each mention has K possible types. We introduce indicator variable m_{ij} (for all values of i and j) which would be equal to 1 if and only if i^{th} mention is of j^{th} type. The probability with which i^{th} mention takes j^{th} type is denoted by p_{ij}. Let x_{ij} be the cost associated with assigning j^{th} type to i^{th} mention. It is given by the following equation:

$$x_{ij} = -\log_{10} p_{ij} \tag{9.1}$$

Now, for i^{th} mention, there are $(i-1)$ mentions which are preceding it. These $(i-1)$ mentions are possible candidates which can serve as the antecedent for i^{th} mention. We introduce an indicator variable c_{ji} to indicate that j^{th} mention is the antecedent for i^{th} mention. Assume that the probability that j^{th} mention is the antecedent for i^{th} mention is given by q_{ji}. Let y_{ji} be the cost associated with assigning j^{th} mention as the antecedent for i^{th} mention. It is given by the following equation:

$$y_{ji} = -\log_{10} q_{ji} \tag{9.2}$$

Let y_{ji}^{C} be the complementary cost of not assigning j^{th} mention as the antecedent of i^{th} mention. Then y_{ji}^{C} is given by the following equation:

$$y_{ji}^{C} = -\log_{10} q_{ji}^{C} = -\log_{10}(1 - q_{ji}) \tag{9.3}$$

Next, we want to impose the constraint that all the mentions (other than pronouns) which are in the same coreference chain should have the same type. We formulate this constraint as a soft constraint in our inference procedure. Let ρ be the cost associated with violating this constraint for any coreference pair. Let w_{jik} be the indicator variable which indicates that if j^{th} mention is chosen as the antecedent for i^{th} mention, then j^{th} mention agrees with i^{th} mention as far as k^{th} type is concerned. Mathematically, it can

be described as follows:

$$w_{jik} \Leftrightarrow 1 - c_{ji} \geq |m_{jk} - m_{ik}| \forall i \forall j \forall k \tag{9.4}$$

Now, consider the following equation:

$$v_{ji} = \frac{1}{2} \sum_{k=1}^{K} (1 - w_{jik}) \tag{9.5}$$

It can be easily verified that v_{ji} would be equal to 1 if and only if j^{th} mention has the same type as i^{th} mention. Otherwise, it would be equal to 0.

Now, the final optimization problem can be written as follows:

$$\begin{aligned} \min & \sum_{i=1}^{N} \sum_{j=1}^{K} \left((-\log_{10} p_{ij}) m_{ij} \right) \\ & + \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j<i}}^{N} \left[\left\{ (-\log_{10} q_{ji}) c_{ji} \right\} + \left\{ -\log_{10} (1 - q_{ji})(1 - c_{ji}) \right\} \right] \\ & + \frac{1}{2} \rho \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j<i}}^{N} \sum_{k=1}^{K} (1 - w_{jik}) \end{aligned} \tag{9.6}$$

subject to:

$$\sum_{j=1}^{K} m_{ij} = 1 \tag{9.7}$$

$$\sum_{\substack{j=1 \\ j<i}}^{N} c_{ji} \leq 1 \forall i \tag{9.8}$$

$$w_{jik} \Leftrightarrow 1 - c_{ji} \geq |m_{jk} - m_{ik}| \forall i \forall j \forall k \tag{9.9}$$

$$m_{ij}, c_{ji}, w_{jik} \in \{0, 1\} \tag{9.10}$$

Equation (9.6) represents the objective of optimization problem. It includes the costs described by Equations (9.1), (9.2), (9.3) and also the penalty associated with violating the constraint that coreferring mentions should have the same type. Equation (9.7) enforces the constraint that each mention can have only one unique type. Equation (9.8) enforces the constraint that any mention can have at most one antecedent. Equation (9.9) is same as Equation (9.4). Finally, Equation (9.10) expresses the fact that m_{ij}, c_{ji} and w_{jik} are all indicator variables.

9.5 Results

In this section, we will compare our system with previous state-of-the-art approaches. We used B-cubed [30], MUC [31] and CEAF [32] as the evaluation metrics in our experiments. The official metric of i2b2 coreference challenge was the unweighted average of F1 scores of these 3 metrics.

We report the scores for both the scenarios: (1) when gold mentions are given and (2) for end-to-end coreference resolution. For evaluation, we used the official evaluation script provided by challenge organizers. As noted before, we have two types of data: $data^{ODIE}$ and $data^{i2b2}$. $data^{ODIE}$ consists of a set of clinical narratives from Mayo Institution and is further subdivided into two categories, namely (1) Clinical reports and (2) Pathology reports. $data^{i2b2}$ consists of a set of clinical narratives from two different institutions namely, (1) Partners Healthcare and (2) Beth Israel. In the following, we will report the scores for different subdivisions of $data^{ODIE}$ and $data^{i2b2}$ separately.

9.5.1 When Gold Mentions Are Given

In this subsection, we will consider the case where the gold mentions are already given and the system has to only identify coreference chains. For this case, coreference relation can exist only within the mentions of same type. However, pronoun mentions can corefer

	CLINICAL	PATH
LIMSI	79.6	67.0
CITY	77.9	61.8
HITS	81.7	67.5
THIS CHAPTER	81.8	70.5

Table 9.1: This table shows that we get best results on both 'clinical' and 'pathology' sections of ODIE corpus for the case where gold mentions are already given.

with any other mention.

ODIE Data

Table 1 shows a comparison of our system with previous state-of-the-art approaches on both sections (clinical and pathology) of ODIE dataset. The numbers shown in Table 1 correspond to average F1 score across all the ODIE categories ("anatomicalsite", "procedure", etc.). From Table 1, we see that we get the best results on both 'clinical' and 'pathology' sections of ODIE dataset.

i2b2 Data

Table 2 shows a comparison of our system with previous state-of-the-art approaches [182, 183] [2,4,26,33-38] on i2b2 dataset. Just like for Table 1, the numbers shown in Table 2 correspond to average F1 score across all the i2b2 categories ("test", "treatment" etc.). From Table 2, we see that we get the best results on both 'partners' and 'beth' corpora.

9.5.2 End-to-End Coreference Resolution

In this subsection, we will present the results for end-to-end coreference resolution. For end-to-end coreference resolution, mentions and their types are not known in advance.

	Partners	Beth
MSRA	86.9	85.9
OPEN	85.2	84.7
CITY	84.2	82.6
BRAND	82.0	81.0
HITS	84.8	82.4
IIS	81.0	80.0
LIMSI	83.8	78.8
UIUC	83.0	78.7
THIS CHAPTER	87.4	86.0

Table 9.2: This table shows that we get best results on both 'partners' and 'beth' corpora for the case where gold mentions are already given.

	CLINICAL	PATH
LIMSI	62.9	58.0
HITS	49.9	50.1
THIS CHAPTER	64.4	63.3

Table 9.3: This table shows that we get best results on both 'clinical' and 'pathology' sections of ODIE corpus for end-to-end coreference resolution.

ODIE Data

Table 3 compares our results with previous best approaches for end-to-end coreference resolution on both 'clinical' and 'pathology' sections of ODIE dataset. The numbers in Table 3 correspond to average F1 score across all the ODIE categories. Table 3 shows that we get the best results on both sections of ODIE dataset.

i2b2 Data

As far as we know, end-to-end results have not been previously reported for i2b2 dataset. In Table 4, we give the results of our system for end-to-end coreference resolution on i2b2 dataset for the first time.

	Partners	**Beth**
THIS CHAPTER	80.5	78.9

Table 9.4: For the first time, we give the results on both 'partners' and 'beth' corpora for end-to-end coreference resolution.

	B-CUBED			MUC			CEAF			Average
	P	R	F1	P	R	F1	P	R	F1	F1
Disease	88.5	87.7	88.1	52.3	43.7	47.6	86.4	63.7	73.3	69.7
Sign	87.9	91.1	89.5	47.1	47.1	47.1	83.4	68.1	75.0	70.5
Anat	80.1	64.1	71.2	29.2	14.6	19.4	65.8	21.9	32.9	41.2
Proc	85.0	80.6	82.7	32.6	19.2	24.1	84.1	53.1	65.1	57.3
Overall	87.4	85.6	86.5	47.1	34.9	40.1	83.5	55.4	66.6	64.4

Table 9.5: This table shows the performance of our system for end-to-end coreference resolution on the test portion of âĂŸclinicalâĂŹ section of Mayo ODIE data.

9.6 Discussion

In Table 5, we show the performance of our system for individual categories for 'clinical' section of ODIE data. This table reports Precision, Recall and F1 score for B-cubed, MUC and CEAF evaluation metrics. It also reports the unweighted average of F1 scores of these 3 metrics. From this table, we can see that average F1 score is about 70% for 'disease' and 'sign' categories. For 'anat' and 'proc' categories, average F1 score is 41.2% and 57.3% respectively. Thus, we see that our system is not performing as well on 'anat' and 'proc' categories as on other 2 categories. The MUC score for 'anat' and 'proc' categories reveals that the recall for these categories is quite low. Thus, our system can perform even better if we manage to improve the recall for 'anat' and 'proc' categories. This will be the subject of future work.

Table 6 shows the performance of our system for individual categories for 'partners' corpora. This table reports the precision, recall and F1 score for B-cubed, MUC and CEAF evaluation metrics. It also reports the average F1 score of these 3 metrics. From this table, we see that the average F1 score for all three categories, namely, 'test', 'treat-

	B-CUBED			MUC			CEAF			Average
	P	R	F1	P	R	F1	P	R	F1	F1
Test	95.2	96.2	95.7	45.8	52.8	49.1	94.0	92.7	93.4	79.4
Treatment	91.6	93.3	92.4	57.0	60.5	58.7	87.5	80.5	83.8	78.3
Problem	94.0	93.2	93.6	62.3	54.6	58.2	90.5	81.7	85.9	79.2
Overall	95.1	95.4	95.2	58.6	57.3	57.9	90.6	85.4	87.9	80.4

Table 9.6: This table shows the performance of our system for end-to-end coreference resolution on the test portion of 'partners' corpus.

ment' and 'problem' is about 79%. Thus, we performed quite well on all the categories for 'partners' corpus. It can also be seen from Table 6 that in general, both precision and recall values are quite high. So, our system doesn't suffer from either poor recall or poor precision.

From Table 5 and Table 6, we see that our system gives better performance on i2b2 corpus than on ODIE corpus. This is partly because of the fact that we had much more training data for i2b2 corpus than for ODIE corpus. One interesting research direction for future can be to examine whether we can use training data with i2b2 annotations to improve the performance on ODIE data.

Chapter 10

Conclusion

This thesis addresses the area of Information Extraction in clinical narratives. It discusses several key IE tasks like mention detection, timex extraction, coreference resolution etc. IE tasks are often related to one another. This thesis presents the methods for solving IE tasks jointly. It also presents the use of several domain-specific knowledge sources to improve the performance of IE tasks. It reports the best performance for the task of coreference resolution on medical corpora.

10.1 Future Work

The work presented in this thesis can be extended in several ways. We discuss some of these possibilities below:

1. I presented the task of only the supervised coreference resolution. It would be interesting to see how well can unsupervised coreference resolution perform in medical domain.

2. For the task of mention detection also, we discussed only supervised methods. However, the constraints that we proposed may work quite well in unsupervised setup also. It would be interesting to see the performance of coreference resolution for unsupervised setup.

3. For timex extraction, we limited the scope to simple temporal expressions. However, it can be extended to complicated expressions as well.

4. Methods described here can also be applied to other types of medical text like blogs etc.

5. For set expansion, we experimented with only distributional methods. Pattern-based methods for set-expansion are likely to improve the performance.

6. It is also possible to do a study on active learning for the IE tasks presented in this thesis.

Appendix A

Hyponym-Hypernym Pairs

Some examples of hyponym-hypernym pairs generated by us are as follows:

1. Adenocarcinoma, carcinoma
2. Birthweight, weight
3. Brachytherapy, therapy
4. Chemotherapy, therapy
5. Cystoprostatectomy, prostatectomy
6. Cytopathology, pathology
7. Empiricvancomycin, vancomycin
8. Gastrojejunostomy, jejunostomy
9. Guidewire, wire
10. Hemicolectomy, colectomy
11. Hemilaminectomy, laminectomy
12. Hemodialysis, dialysis
13. Hepatosplenomegaly, splenomegaly
14. Ischemiccardiomyopathy, cardiomyopathy
15. Ketoacidosis, acidosis
16. Levalbuterol, albuterol
17. Lymphadenopathy, adenopathy
18. Methemoglobin, hemoglobin
19. Orhydronephrosis, hydronephrosis
20. Osteoarthritic, arthritic

21. Osteochondromatosis, chondromatosis
22. Periampullary, ampullary
23. Peripancreatic, pancreatic
24. Plasmapheresis, pheresis
25. Radiotherapy, therapy
26. Serratiaurosepsis, sepsis
27. Thromboembolus, embolus
28. Urosepsis, sepsis

Appendix B

Clinical Patterns Used

Following is a list of incompatible pairs of anatomical terms:

1. ipsilateral, contralateral
2. superficial, deep
3. visceral, parietal
4. axial, abaxial
5. rostral, caudal
6. anterior, posterior
7. dorsal, ventral
8. left, right
9. proximal, distal

Appendix C

Popular Drug Abuse Substances

Following is a list of most popular substances which are used for drug abuse:

1. alcohol
2. amphetamines
3. anabolic steroids
4. barbiturates
5. beer
6. benzodiazepines (particularly alprazolam, temazepam, diazepam and clonazepam)
7. buprenorphine
8. butane
9. cannabis
10. club drugs
11. cocaine
12. depressants (sedatives)
13. ecstasy
14. GHB
15. hallucinogens
16. heroin
17. inhalants
18. ketamine
19. LSD
20. marijuana

21. mephedrone
22. methamphetamine
23. methadone
24. methaqualone
25. narcotics
26. opioids
27. pain relievers
28. pcp
29. psychotherapeutics
30. qat/khat
31. rum
32. stimulants
33. tobacco
34. tranquilizers
35. whisky
36. wine

Appendix D

Representatives of Drug Abuse Concepts in SNOMED CT

Following list shows the representatives of the concept type "drugs used for substance abuse" in SNOMED CT encyclopedia:

1. Psychoactive substance
2. Alcoholic Beverage
3. Central Depressant
4. Alcohol agent
5. Alcohol products
6. Substance of abuse
7. Cannabis
8. Hallucinogen
9. Cannabinoid
10. Nicotiana
11. Tobacco
12. Tobacco smoking behavior
13. Tobacco use and exposure
14. Psychotherapeutic agent
15. Psychostimulant
16. Opiate
17. Morphine Derivative
18. Analgesic
19. Anesthetic

20. Drugs used to treat addiction
21. Carboxylic acid and/or salt
22. Barbiturate
23. Centrally acting muscle relaxant
24. Centrally acting hypotensive agent
25. Cardiovascular agent
26. Sympathomimetic agent
27. Aralkylamine
28. Inhaled Drug Administration
29. Hypnotics
30. Anxiolytic, sedative AND/OR hypnotic

Appendix E

Drug Abuse Concepts that We Missed

Following is a list of drug abuse substances that were not detected by our software:

1. actiq
2. adderall
3. ambien
4. amytal
5. anexsia
6. antabuse
7. ativan
8. avinza
9. biocodone
10. campral
11. concerta
12. damason-P
13. darvocet
14. darvon
15. demerol
16. depade
17. desoxyn
18. dexedrine
19. dextrostat
20. di-gesic

21. dicodid
22. dilaudid
23. duodin
24. duragesic
25. duramorph
26. fioricet
27. fiorinal
28. halcion
29. hycodan
30. hydrococet
31. kadian
32. kapanol
33. klonopin
34. LAAM
35. librium
36. lorcet
37. lortab
38. luminal
39. ms contin
40. msir
41. methadrine
42. mushrooms
43. nembutal
44. norco
45. oramorph
46. orlaam
47. PCP

48. palladone
49. panacet
50. percocet
51. percodan
52. quaalude
53. revia
54. ritalin
55. rohypnol
56. roxanol
57. roxicodone
58. ryzolt
59. seconal
60. soma
61. speed
62. steroids
63. stilnox
64. sublimaze
65. suboxone
66. subutex
67. symtan
68. temesta
69. tramal
70. tussionex
71. tylox
72. ultram
73. valium
74. vicodin

75. vicoprofen
76. vivitrol
77. xanax
78. xodol
79. zydone

References

[1] W. Chapman, P. Nadkarni, L. Hirschman, L. D'Avolio, G. Savova, and O. Uzuner, "Overcoming barriers to nlp for clinical text: the role of shared tasks and the need for additional creative solutions," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 540–543, 2011.

[2] D. Roth and W. Yih, "A linear programming formulation for global inference in natural language tasks," in *CoNLL*. Association for Computational Linguistics, 2004, pp. 1–8.

[3] D. Roth and W. Yih, "Integer linear programming inference for conditional random fields," in *ICML*, 2005. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/RothYi05.pdf pp. 737–744.

[4] D. Roth and W. Yih, "Global inference for entity and relation identification via a linear programming formulation," *Introduction to Statistical Relational Learning*, pp. 553–580, 2007.

[5] M.-W. Chang, L. Ratinov, N. Rizzolo, and D. Roth, "Learning and inference with constraints," in *Proceedings of the 23rd national conference on Artificial intelligence*, 2008, pp. 1513–1518.

[6] A. Aronson and F. Lang, "An overview of metamap: historical perspective and recent advances," *Journal of the American Medical Informatics Association*, vol. 17, no. 3, p. 229, 2010.

[7] L. F. Rau, "Extracting company names from text," in *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, vol. 1. IEEE, 1991, pp. 29–32.

[8] N. Chinchor and P. Robinson, "Muc-7 named entity task definition," in *Proceedings of the 7th Conference on Message Understanding*, 1997.

[9] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-performance learning name-finder," in *Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics, 1997, pp. 194–201.

[10] S. Sekine et al., "Nyu: Description of the japanese ne system used for met-2," in *Proc. of the Seventh Message Understanding Conference (MUC-7)*, vol. 17, 1998.

[11] 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*. Citeseer, 1998.

[12] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 188–191.

[13] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 142–147.

[14] O. Uzuner, B. South, S. Shen, and S. DuVall, "2010 i2b2/va challenge on concepts, assertions, and relations in clinical text," *Journal of American Medical Informatics Association*, 2011.

[15] M. Jiang, Y. Chen, M. Liu, S. Rosenbloom, S. Mani, J. Denny, and H. Xu, "A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 601–606, 2011.

[16] J. Patrick, D. Nguyen, Y. Wang, and M. Li, "A knowledge discovery and reuse pipeline for information extraction in clinical notes," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 574–579, 2011.

[17] S. Jonnalagadda, T. Cohen, S. Wu, and G. Gonzalez, "Enhancing clinical concept extraction with distributional semantics," *Journal of biomedical informatics*, vol. 45, no. 1, pp. 129–140, 2012.

[18] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.

[19] B. de Bruijn, C. Cherry, S. Kiritchenko, J. Martin, and X. Zhu, "Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 557–562, 2011.

[20] M. Torii, K. Wagholikar, and H. Liu, "Using machine learning for concept extraction on clinical documents from multiple data sources," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 580–587, 2011.

[21] K. Roberts and S. Harabagiu, "A flexible framework for deriving assertions from electronic medical records," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 568–573, 2011.

[22] P. Gooch and A. Roudsari, "Lexical patterns, features and knowledge resources for coreference resolution in clinical notes," *Journal of Biomedical Informatics*, 2012.

[23] A. Bodnari, P. Szolovits, and Ö. Uzuner, "Mcores: a system for noun phrase coreference resolution for clinical records," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 906–912, 2012.

[24] Y. Xu, J. Liu, J. Wu, Y. Wang, Z. Tu, J. Sun, J. Tsujii, I. Eric, and C. Chang, "A classification approach to coreference in discharge summaries: 2011 i2b2 challenge," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 897–905, 2012.

[25] P. Jindal and D. Roth, "Using domain knowledge and domain-inspired discourse model for coreference resolution for clinical narratives," *Journal of the American Medical Informatics Association*, 2012.

[26] B. Rink, K. Roberts, and S. Harabagiu, "A supervised framework for resolving coreference in clinical records," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 875–882, 2012.

[27] S. Jonnalagadda, D. Li, S. Sohn, S. Wu, K. Wagholikar, M. Torii, and H. Liu, "Coreference analysis in clinical notes: a multi-pass sieve with alternate anaphora resolution modules," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 867–874, 2012.

[28] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning, "A multi-pass sieve for coreference resolution," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 492–501.

[29] J. Zheng, W. Chapman, R. Crowley, and G. Savova, "Coreference resolution: A review of general methodologies and applications in the clinical domain," *Journal of biomedical informatics*, 2011.

[30] J. Zheng, W. Chapman, T. Miller, C. Lin, R. Crowley, and G. Savova, "A system for coreference resolution for the clinical narrative," *Journal of the American Medical Informatics Association*, 2012.

[31] J. R. Hobbs, "Resolving pronoun references," *Lingua*, vol. 44, no. 4, pp. 311–338, 1978.

[32] M. A. Walker, A. K. Joshi, and E. F. Prince, *Centering theory in discourse*. Oxford University Press on Demand, 1998.

[33] W. E. Winkler, "The state of record linkage and current research problems," in *Statistical Research Division, US Census Bureau*. Citeseer, 1999.

[34] A. E. Monge, C. Elkan et al., "The field matching problem: Algorithms and applications." in *KDD*, 1996, pp. 267–270.

[35] A. E. Monge and C. P. Elkan, "Efficient domain-independent detection of approximately duplicate database records," in *Proc. of the ACM-SIGMOD Workshop on Research Issues in on Knowledge Discovery and Data Mining*, 1997.

[36] V. Ng, "Supervised noun phrase coreference research: The first fifteen years," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 1396–1411.

[37] R. Mitkov, *Anaphora resolution*. Longman London, 2002, vol. 134.

[38] R. Mitkov, *Anaphora resolution: the state of the art*. Citeseer, 1999.

[39] S. P. Ponzetto and M. Poesio, "State-of-the-art nlp approaches to coreference resolution: theory and practical recipes," in *Tutorial Abstracts of ACL-IJCNLP 2009*. Association for Computational Linguistics, 2009, pp. 6–6.

[40] B. J. Grosz, "The representation and use of focus in dialogue understanding." 1977.

[41] C. L. Sidner, "Towards a computational theory of definite anaphora comprehension in english discourse." DTIC Document, Tech. Rep., 1979.

[42] B. J. Grosz, A. K. Joshi, and S. Weinstein, "Providing a unified account of definite noun phrases in discourse," in *Proceedings of the 21st annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1983, pp. 44–50.

[43] B. J. Grosz, S. Weinstein, and A. K. Joshi, "Centering: A framework for modeling the local coherence of discourse," *Computational linguistics*, vol. 21, no. 2, pp. 203–225, 1995.

[44] C. Aone and S. W. Bennett, "Evaluating automated and manual acquisition of anaphora resolution strategies," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1995, pp. 122–129.

[45] J. F. McCarthy and W. G. Lehnert, "Using decision trees for coreference resolution," *arXiv preprint cmp-lg/9505043*, 1995.

[46] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

[47] A. McCallum and B. Wellner, "Conditional models of identity uncertainty with application to noun coreference," *Advances in neural information processing systems*, vol. 17, pp. 905–912, 2004.

[48] D. Zelenko, C. Aone, and J. Tibbetts, "Coreference resolution for information extraction," in *Proceedings of the ACL Workshop on Reference Resolution and its Applications*, 2004, pp. 9–16.

[49] T. Finley and T. Joachims, "Supervised clustering with support vector machines," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 217–224.

[50] C. Nicolae and G. Nicolae, "Bestcut: A graph algorithm for coreference resolution," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 275–283.

[51] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos, "A mention-synchronous coreference resolution algorithm based on the bell tree," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 135.

[52] S. Lappin and H. J. Leass, "An algorithm for pronominal anaphora resolution," *Computational linguistics*, vol. 20, no. 4, pp. 535–561, 1994.

[53] C. Kennedy and B. Boguraev, "Anaphora for everyone: pronominal anaphora resoluation without a parser," in *Proceedings of the 16th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1996, pp. 113–118.

[54] R. Evans, "Applying machine learning toward an automatic classification of it," *Literary and linguistic computing*, vol. 16, no. 1, pp. 45–57, 2001.

[55] C. Müller, "Automatic detection of nonreferential it in spoken multi-party dialog," 2006.

[56] Y. Versley, A. Moschitti, M. Poesio, and X. Yang, "Coreference systems based on kernels methods," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 961–968.

[57] S. Bergsma, D. Lin, and R. Goebel, "Distributional identification of non-referential pronouns," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT-08)*, 2008, pp. 10–18.

[58] R. V. M. Poesio, "Processing definite descriptions in corpora," *Corpus-Based and Computational Approaches to Discourse Anaphora*, vol. 3, p. 189, 2000.

[59] A. McCallum and B. Wellner, "Toward conditional models of identity uncertainty with application to proper noun coreference," 2003.

[60] X. Yang, J. Su, G. Zhou, and C. L. Tan, "An np-cluster based approach to coreference resolution," in *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004, p. 226.

[61] X. Yang, J. Su, J. Lang, C. L. Tan, T. Liu, and S. Li, "An entity-mention model for coreference resolution with inductive logic programming," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2008, pp. 843–851.

[62] R. Iida, K. Inui, H. Takamura, and Y. Matsumoto, "Incorporating contextual cues in trainable models for coreference resolution," in *Proceedings of the EACL Workshop on The Computational Treatment of Anaphora*, 2003, pp. 23–30.

[63] X. Yang, G. Zhou, J. Su, and C. L. Tan, "Coreference resolution using competition learning approach," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, July 2003. [Online]. Available: http://www.aclweb.org/anthology/P03-1023 pp. 176–183.

[64] X. Yang, J. Su, and C. L. Tan, "A twin-candidate model for learning-based anaphora resolution," *Computational Linguistics*, vol. 34, no. 3, pp. 327–356, 2008.

[65] A. Rahman and V. Ng, "Supervised models for coreference resolution," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 968–977.

[66] L. Qiu, M.-Y. Kan, and T.-S. Chua, "A public reference implementation of the rap anaphora resolution algorithm," *arXiv preprint cs/0406031*, 2004.

[67] M. Poesio and M. A. Kabadjov, "A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation," in *Proceedings of LREC*, 2004.

[68] Y. Versley, S. P. Ponzetto, M. Poesio, V. Eidelman, A. Jern, J. Smith, X. Yang, and A. Moschitti, "Bart: A modular toolkit for coreference resolution," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*. Association for Computational Linguistics, 2008, pp. 9–12.

[69] P. Denis and J. Baldridge, "Specialized models and ranking for coreference resolution," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 660–669.

[70] V. Stoyanov, C. Cardie, N. Gilbert, E. Riloff, D. Buttler, and D. Hysom, "Coreference resolution with reconcile," in *Proceedings of the ACL 2010 Conference Short Papers*. Association for Computational Linguistics, 2010, pp. 156–161.

[71] E. Charniak and M. Elsner, "Em works for pronoun anaphora resolution," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 148–156.

[72] T. Miller, D. Dligach, and G. Savova, "Active learning for coreference resolution," in *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*. Montréal, Canada: Association for Computational Linguistics, June 2012. [Online]. Available: http://www.aclweb.org/anthology/W12-2409 pp. 73–81.

[73] E. Apostolova, N. Tomuro, P. Mongkolwat, and D. Demner-Fushman, "Domain adaptation of coreference resolution for radiology reports," in *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*. Montréal, Canada: Association for Computational Linguistics, June 2012. [Online]. Available: http://www.aclweb.org/anthology/W12-2414 pp. 118–121.

[74] R. T. Batista-Navarro and S. Ananiadou, "Building a coreference-annotated corpus from the domain of biochemistry," in *Proceedings of BioNLP 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011. [Online]. Available: http://www.aclweb.org/anthology/W11-0210 pp. 83–91.

[75] N. Nguyen, J.-D. Kim, and J. Tsujii, "Overview of bionlp 2011 protein coreference shared task," in *Proceedings of BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011. [Online]. Available: http://www.aclweb.org/anthology/W11-1811 pp. 74–82.

[76] Y. Kim, E. Riloff, and N. Gilbert, "The taming of reconcile as a biomedical coreference resolver," in *Proceedings of BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011. [Online]. Available: http://www.aclweb.org/anthology/W11-1813 pp. 89–93.

[77] N. T. H. Nguyen and Y. Tsuruoka, "Extracting bacteria biotopes with semi-supervised named entity recognition and coreference resolution," in *Proceedings of BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011. [Online]. Available: http://www.aclweb.org/anthology/W11-1814 pp. 94–101.

[78] D. Tuggener, M. Klenner, G. Schneider, S. Clematide, and F. Rinaldi, "An incremental model for the coreference resolution task of bionlp 2011," in *Proceedings of BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011. [Online]. Available: http://www.aclweb.org/anthology/W11-1823 pp. 151–152.

[79] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, Jan. 1986.

[80] J. Eisner, "Three new probabilistic models for dependency parsing: An exploration," in *COLING*, Copenhagen, August 1996. [Online]. Available: http://cs.jhu.edu/~jason/papers/#coling96 pp. 340–345.

[81] D. Roth, "Learning in natural language," in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/ijcai99r.pdf pp. 898–904.

[82] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," in *SIGDAT*, 2002.

[83] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Machine Learning*, vol. 37, no. 3, pp. 277–296, 1999.

[84] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=645530.655813 pp. 282–289.

[85] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–285, 1989.

[86] J. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 363–370.

[87] M.-W. Chang, L. Ratinov, and D. Roth, "Guiding semi-supervision with constraint-driven learning," *Urbana*, vol. 51, p. 61801, 2007.

[88] J. V. Graca, K. Ganchev, and B. Taskar, "Expectation maximization and posterior constraints," in *NIPS*, vol. 20, 2007.

[89] K. Bellare, G. Druck, and A. McCallum, "Alternating projections for learning with expectation constraints," in *UAI*, 2009.

[90] A. Carlson, J. Betteridge, R. C. Wang, E. R. H. Jr., and T. M. Mitchell, "Coupled semi-supervised learning for information extraction," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, 2010.

[91] K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar, "Posterior regularization for structured latent variable models," *The Journal of Machine Learning Research*, vol. 11, pp. 2001–2049, 2010.

[92] D. Roth and W. Yih, "A linear programming formulation for global inference in natural language tasks," in *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*, H. T. Ng and E. Riloff, Eds. Association for Computational Linguistics, 2004. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/RothYi04.pdf pp. 1–8.

[93] M. Chang, L. Ratinov, and D. Roth, "Guiding semi-supervision with constraint-driven learning," in *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Prague, Czech Republic: Association for Computational Linguistics, 6 2007. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/ChangRaRo07.pdf pp. 280–287.

[94] V. Punyakanok, D. Roth, and W. Yih, "The necessity of syntactic parsing for semantic role labeling," in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/PunyakanokRoYi05.pdf pp. 1117–1123.

[95] D. Roth and W. Yih, "Global inference for entity and relation identification via a linear programming formulation," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2007. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/RothYi07.pdf

[96] D. Roth and W. Yih, "Integer linear programming inference for conditional random fields," in *Proc. of the International Conference on Machine Learning (ICML)*, 2005. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/RothYi05.pdf pp. 737–744.

[97] R. Barzilay and M. Lapata, "Aggregation via Set Partitioning for Natural Language Generation," in *Proc. of HLT/NAACL*, June 2006.

[98] J. Clarke and M. Lapata, "Constraint-based sentence compression: An integer programming approach," in *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*. Sydney, Australia: ACL, July 2006, pp. 144–151.

[99] L. Tanabe and W. Wilbur, "Tagging gene and protein names in biomedical text," *Bioinformatics*, vol. 18, no. 8, pp. 1124–1132, 2002.

[100] H. Yu, V. Hatzivassiloglou, C. Friedman, A. Rzhetsky, and W. Wilbur, "Automatic extraction of gene and protein synonyms from medline and journal articles." in *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 2002, p. 919.

[101] L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia, "Overview of biocreative: critical assessment of information extraction for biology," *BMC bioinformatics*, vol. 6, no. Suppl 1, p. S1, 2005.

[102] E. Brill, "Processing natural language without natural language processing," *Computational Linguistics and Intelligent Text Processing*, pp. 179–185, 2003.

[103] J. Chang, H. Schütze, and R. Altman, "Gapscore: finding gene and protein names one word at a time," *Bioinformatics*, vol. 20, no. 2, pp. 216–225, 2004.

[104] R. ZIMMER, "Playing biology's name game: identifying protein names in scientific text," in *Pacific Symposium on Biocomputing 2003: Kauai, Hawaii, 3-7 January 2003*. World Scientific Publishing Company Incorporated, 2002, p. 403.

[105] G. Zhou, J. Zhang, J. Su, D. Shen, and C. Tan, "Recognizing names in biomedical texts: a machine learning approach," *Bioinformatics*, vol. 20, no. 7, pp. 1178–1190, 2004.

[106] J. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "Genia corpusŮa semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19, no. suppl 1, pp. i180–i182, 2003.

[107] M. Narayanaswamy, K. Ravikumar, K. Vijay-Shanker, and K. Ay-shanker, "A biological named entity recognizer," in *Pac Symp Biocomput*, 2003, p. 427.

[108] S. Mika and B. Rost, "Protein names precisely peeled off free text," *Bioinformatics*, vol. 20, no. suppl 1, pp. i241–i247, 2004.

[109] J. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, "Overview of bionlp'09 shared task on event extraction," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Association for Computational Linguistics, 2009, pp. 1–9.

[110] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski, "Extracting complex biological events with rich graph-based feature sets," in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*. Boulder, Colorado: Association for Computational Linguistics, June 2009. [Online]. Available: http://www.aclweb.org/anthology/W09-1402 pp. 10–18.

[111] S. Riedel, H.-W. Chun, T. Takagi, and J. Tsujii, "A markov logic approach to bio-molecular event extraction," in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*. Boulder, Colorado: Association for Computational Linguistics, June 2009. [Online]. Available: http://www.aclweb.org/anthology/W09-1406 pp. 41–49.

[112] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[113] A. Ekbal, E. Sourjikova, A. Frank, and S. Ponzetto, "Assessing the challenge of fine-grained named entity recognition and classification," in *Proceedings of the 2010 Named Entities Workshop*. Association for Computational Linguistics, 2010, pp. 93–101.

[114] E. Bengtson and D. Roth, "Understanding the value of features for coreference resolution," in *Proceedings of the Conference on EMNLP*. Association for Computational Linguistics, 2008, pp. 294–303.

[115] L. Sarmento, V. Jijkuon, M. de Rijke, and E. Oliveira, "More like these: growing entity classes from seeds," in *Proceedings of the sixteenth ACM conference on CIKM*. ACM, 2007, pp. 959–962.

[116] P. Pantel, E. Crestan, A. Borkovsky, A. Popescu, and V. Vyas, "Web-scale distributional similarity and entity set expansion," in *Proceedings of the 2009 Conference on EMNLP*. ACL, 2009, pp. 938–947.

[117] V. Vyas, P. Pantel, and E. Crestan, "Helping editors choose better seed sets for entity set expansion," in *Proceeding of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 225–234.

[118] R. Wang and W. Cohen, "Iterative set expansion of named entities using the web," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 1091–1096.

[119] R. Wang and W. Cohen, "Language-independent set expansion of named entities using the web," in *ICDM*. IEEE Computer Society, 2007, pp. 342–350.

[120] R. Wang and W. Cohen, "Character-level analysis of semi-structured documents for set expansion," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, 2009, pp. 1503–1512.

[121] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artificial Intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[122] E. Riloff and R. Jones, "Learning dictionaries for information extraction by multi-level bootstrapping," in *Proceedings of the National Conference on Artificial Intelligence*. JOHN WILEY & SONS LTD, 1999, pp. 474–479.

[123] P. Talukdar, T. Brants, M. Liberman, and F. Pereira, "A context pattern induction method for named entity extraction," in *Proceedings of the Tenth Conference on CoNLL*. ACL, 2006, pp. 141–148.

[124] P. Talukdar, J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira, "Weakly-supervised acquisition of labeled class instances using graph random walks," in *Proceedings of the Conference on EMNLP*. ACL, 2008, pp. 582–590.

[125] M. Pennacchiotti and P. Pantel, "Entity extraction via ensemble semantics," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 238–247.

[126] M. Pasca and B. Van Durme, "Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs," in *Proceedings of the 46th Annual Meeting of the ACL (ACL-08)*. Citeseer, 2008, pp. 19–27.

[127] Z. Ghahramani and K. Heller, "Bayesian sets," *Advances in Neural Information Processing Systems*, vol. 18, p. 435, 2006.

[128] M. Thelen and E. Riloff, "A bootstrapping method for learning semantic lexicons using extraction pattern contexts," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 214–221.

[129] W. Lin, R. Yangarber, and R. Grishman, "Bootstrapped learning of semantic classes from positive and negative examples," in *Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, vol. 1, no. 4, 2003, p. 21.

[130] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*, 6 2009. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/RatinovRo09.pdf

[131] C. Manning, H. Schutze, and MITCogNet, *Foundations of statistical natural language processing*. MIT Press, 1999, vol. 59.

[132] P. Pantel and D. Lin, "Discovering word senses from text," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 613–619.

[133] G. Salton and M. McGill, *Introduction to modern information retrieval*. McGraw-Hill New York, 1983, vol. 1.

[134] D. Graff, J. Kong, K. Chen, and K. Maeda, "English gigaword," *Linguistic Data Consortium, Philadelphia*, 2003.

[135] A. Minard, A. Ligozat, A. Abacha, D. Bernhard, B. Cartoni, L. Deléger, B. Grau, S. Rosset, P. Zweigenbaum, and C. Grouin, "Hybrid methods for improving information access in clinical documents: Concept, assertion, and relation identification," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 588–593, 2011.

[136] L. D'Avolio, T. Nguyen, S. Goryachev, and L. Fiore, "Automated concept-level information extraction to reduce the need for custom software and rules development," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 607–613, 2011.

[137] Y. Xu, K. Hong, J. Tsujii, I. Eric, and C. Chang, "Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 824–832, 2012.

[138] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 591–598.

[139] V. Punyakanok, D. Roth, W. Yih, and D. Zimak, "Semantic role labeling via integer linear programming inference," in *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004, p. 1346.

[140] T. Marciniak and M. Strube, "Beyond the pipeline: Discrete optimization in nlp," in *Proceedings of the Ninth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2005, pp. 136–143.

[141] P. Bramsen, P. Deshpande, Y. Lee, and R. Barzilay, "Inducing temporal graphs," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 189–198.

[142] R. Barzilay and M. Lapata, "Aggregation via set partitioning for natural language generation," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 359–366.

[143] S. Riedel and J. Clarke, "Incremental integer linear programming for non-projective dependency parsing," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 129–137.

[144] J. Clarke and M. Lapata, "Global inference for sentence compression: An integer linear programming approach," *Journal of Artificial Intelligence Research*, vol. 31, no. 1, pp. 399–429, 2008.

[145] P. Denis, J. Baldridge et al., "Joint determination of anaphoricity and coreference resolution using integer programming," in *Proceedings of NAACL HLT*, 2007, pp. 236–243.

[146] K.-W. Chang, R. Samdani, A. Rozovskaya, N. Rizzolo, M. Sammons, and D. Roth, "Inference protocols for coreference resolution," in *CoNLL Shared Task*. Portland, Oregon, USA: Association for Computational Linguistics, 2011. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/CSRRSR11.pdf pp. 40–44.

[147] 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.

[148] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of EMNLP*, vol. 4, 2004, pp. 388–395.

[149] G. S. Mann and A. McCallum, "Simple, robust, scalable semi-supervised learning via expectation regularization," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 593–600.

[150] G. Mann and A. McCallum, "Generalized expectation criteria for semi-supervised learning of conditional random fields," in *Proc. ACL*, 2008, pp. 870–878.

[151] R. Reichart and R. Barzilay, "Multi event extraction guided by global constraints," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2012, pp. 70–79.

[152] M.-W. Chang, L. Ratinov, and D. Roth, "Structured learning with constrained conditional models," *Machine learning*, pp. 1–33, 2012.

[153] H. Harkema, J. N. Dowling, T. Thornblade, and W. W. Chapman, "Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports," *Journal of biomedical informatics*, vol. 42, no. 5, pp. 839–851, 2009.

[154] W. Chapman, W. Bridewell, P. Hanbury, G. Cooper, and B. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries," *Journal of biomedical informatics*, vol. 34, no. 5, pp. 301–310, 2001.

[155] L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia." in *ACL*, vol. 11, 2011, pp. 1375–1384.

[156] Y. Yan, N. Okazaki, Y. Matsuo, Z. Yang, and M. Ishizuka, "Unsupervised relation extraction by mining wikipedia texts using information from the web," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1021–1029.

[157] T.-V. T. Nguyen and A. Moschitti, "End-to-end relation extraction using distant supervision from external semantic repositories." in *ACL (Short Papers)*, 2011, pp. 277–282.

[158] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 118–127.

[159] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, "Textrunner: open information extraction on the web," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 2007, pp. 25–26.

[160] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web." in *IJCAI*, vol. 7, 2007, pp. 2670–2676.

[161] O. Uryupina, M. Poesio, C. Giuliano, and K. Tymoshenko, "Disambiguation and filtering methods in using web knowledge for coreference resolution," in *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference*, 2011.

[162] A. Rahman and V. Ng, "Coreference resolution with world knowledge," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 814–824.

[163] V. Bryl, C. Giuliano, L. Serafini, and K. Tymoshenko, "Using background knowledge to support coreference resolution," in *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), August*, 2010.

[164] S. Ponzetto and M. Strube, "Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution," in *Proceedings of the NAACL*. Association for Computational Linguistics, 2006, pp. 192–199.

[165] D. Bean and E. Riloff, "Unsupervised learning of contextual role knowledge for coreference resolution," in *Proc. of HLT/NAACL*, 2004, pp. 297–304.

[166] O. Uzuner, A. Bodnari, S. Shen, T. Forbush, J. Pestian, and B. R. South, "Evaluating the state of the art in coreference resolution for electronic medical records," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 786–791, 2012.

[167] A. Bagga and B. Baldwin, "Algorithms for scoring coreference chains," in *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*. Citeseer, 1998, pp. 563–566.

[168] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, "A model-theoretic coreference scoring scheme," in *Proceedings of the 6th conference on Message understanding*. Association for Computational Linguistics, 1995, pp. 45–52.

[169] X. Luo, "On coreference resolution performance metrics," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2005, pp. 25–32.

[170] V. Ng and C. Cardie, "Improving machine learning approaches to coreference resolution," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 104–111.

[171] L. Ratinov and D. Roth, "Learning-based multi-sieve co-reference resolution with knowledge," in *EMNLP*, 2012.

[172] H. Poon and P. Domingos, "Joint unsupervised coreference resolution with markov logic," in *Proceedings of the Conference on EMNLP*. Association for Computational Linguistics, 2008, pp. 650–659.

[173] V. Ng and C. Cardie, "Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.

[174] C. Paice and G. Husk, "Towards the automatic recognition of anaphoric features in english text: the impersonal pronoun ŞitŤ," *Computer Speech & Language*, vol. 2, no. 2, pp. 109–132, 1987.

[175] H. Dai, C. Chen, C. Wu, P. Lai, R. Tsai, and W. Hsu, "Coreference resolution of medical concepts in discharge summaries by exploiting contextual information," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 888–896, 2012.

[176] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes," *EMNLP-CoNLL 2012*, p. 1, 2012.

[177] J. Cai, E. Mujdricza-Maydt, Y. Hou, and M. Strube, "Weakly supervised graph-based coreference resolution for clinical texts," in *Proceedings of the 2011 i2b2/VA/Cincinnati Workshop on Challenges in Natural Language Processing for Clinical Data.*, 2011.

[178] M. Lan, J. Zhao, K. Zhang, H. Shi, and J. Cai, "Comparative investigation on learning-based and rule-based approaches to coreference resolution in clinic domain: A case study in i2b2 challenge 2011 task 1," in *Proceedings of the 2011 i2b2/VA/Cincinnati Workshop on Challenges in Natural Language Processing for Clinical Data. i2b2. Boston, MA, USA*, 2011.

[179] C. Grouin, M. Dinarelli, S. Rosset, G. Wisniewski, and P. Zweigenbaum, "Coreference resolution in clinical reports - the limsi participation in the i2b2/va 2011 challenge," in *Proceedings of the 2011 i2b2/VA/Cincinnati Workshop on Challenges in Natural Language Processing for Clinical Data*, 2011.

[180] P. Denis, J. Baldridge et al., "Global joint models for coreference resolution and named entity classification," *Procesamiento del Lenguaje Natural*, vol. 42, pp. 87–96, 2009.

[181] G. K. Savova, W. W. Chapman, J. Zheng, and R. S. Crowley, "Anaphoric relations in the clinical narrative: corpus creation," *Journal of the American Medical Informatics Association*, vol. 18, no. 4, pp. 459–465, 2011.

[182] H. Yang, A. Willis, A. de Roeck, and B. Nuseibeh, "A system for coreference resolution in clinical documents," in *Proceedings of the 2011 i2b2/VA/Cincinnati Workshop on Challenges in Natural Language Processing for Clinical Data*, 2011.

[183] P. Anick, P. Hong, N. Xue, and Y. Yang, "Coreference resolution for electronic medical records," in *Proceedings of the 2011 i2b2/VA/Cincinnati Workshop on Challenges in Natural Language Processing for Clinical Data*, 2011.