Groundtruth Budgeting: A Novel Approach to Semi-Supervised Relation Extraction in Medical Language

by
Russell J. Ryan
S.B., Massachusetts Institute of Technology (2009)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

Jan 2011 [February 2011]

Author ....
Department of Electrical Engineering and Computer Science
Jan 31, 2011

Certified by ....
Özlem Uzuner
Assistant Professor of Information Studies – SUNY, Albany
Thesis Co-Supervisor

Certified by ....
Peter Szolovits
Professor of Computer Science and Engineering – MIT CSAIL
Thesis Co-Supervisor

Accepted by ....
Dr. Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

Groundtruth Budgeting: A Novel Approach to Semi-Supervised Relation Extraction in Medical Language

by

Russell J. Ryan

Submitted to the
Department of Electrical Engineering and Computer Science

Jan 31, 2011

In partial fulfillment of the requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

We address the problem of weakly-supervised relation extraction in hospital discharge summaries. Sentences with pre-identified concept types (for example: medication, test, problem, symptom) are labeled with the relationship between the concepts. We present a novel technique for weakly-supervised bootstrapping of a classifier for this task: Groundtruth Budgeting. In the case of highly-overlapping, self-similar datasets as is the case with the 2010 i2b2/VA challenge corpus, the performance of classifiers on the minority classes is often poor. To address this we set aside a random portion of the groundtruth at the beginning of bootstrapping which will be gradually added as the classifier is bootstrapped. The classifier chooses groundtruth samples to be added by measuring the confidence of its predictions on them and choosing samples for which it has the least confident predictions. By adding samples in this fashion, the classifier is able to increase its coverage of the decision space while not adding too many majority-class examples. We evaluate this approach on the 2010 i2b2/VA challenge corpus containing of 477 patient discharge summaries and show that with a training corpus of 349 discharge summaries, budgeting 10% of the corpus achieves equivalent results to a bootstrapping classifier starting with the entire corpus. We compare our results to those of other papers published in the proceedings of the 2010 Fourth i2b2/VA Shared-Task and Workshop.

Thesis Co-Supervisor: Özlem Uzuner
Title: Assistant Professor of Information Studies – SUNY, Albany

Thesis Co-Supervisor: Peter Szolovits
Title: Professor of Computer Science and Engineering – MIT CSAIL

Acknowledgments

This research would not have been possible without the advice, ideas, prodding, and patience of my adviser, Prof. Özlem Uzuner. First and foremost, thank you Özlem. The sage wisdom and direction of Prof. Peter Szolovits were also invaluable and for that I'm very thankful.

My sincere thanks to go Jason Rennie and Mark Tobenkin for letting me tap into their expert knowledge of machine learning and convex optimization and for putting up with my inane questions.

I thank Prof. Szolovits and Prof. Uzuner for generously funding my research for the full three terms I have been here, and the MIT EECS department for accepting me to the MEng program. It has been a valuable learning experience.

Finally, I'd like to extend a special thanks to my family and friends who supported and encouraged me throughout my time at the Institvte.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures **11**

List of Tables **15**

1 Introduction **17**

1.1 Motivation 18

1.1.1 Annotation Cost 18

1.1.2 Real World Applications 18

1.2 Self-Similarity of Medical Language Corpora 20

1.3 Contributions 20

1.4 Thesis Structure 21

2 Background and Related Work **23**

2.1 Problem Overview 23

2.1.1 The 2010 i2b2/VA Challenge Corpus 24

2.2 Related Work 25

2.2.1 Bootstrapping 27

3 System Overview **29**

3.1 PyCARE 29

3.2 Pre-processing 30

3.2.1 Sentence-Breaking / Tokenization 30

3.2.2 Stemming 31

3.2.3 Part-of-Speech Tagging 31

3.2.4 Link Grammar Parsing 31

3.2.5 UMLS CUI and Relation Identification 31

4 Methods **33**

4.1 Support Vector Machines 33

4.2 Lexical and Syntactic Features 34

4.3 Bootstrapping 35

4.4 System Parameters 36

4.4.1 Choice of Kernel 36

4.4.2 Seed Size 37

4.4.3 Confidence Metrics 37

4.4.4 Example Filter Metrics 38

4.4.5 Stopping Criterion 40

4.5 Seed Selection 40

4.5.1 Maximal Cosine Diversity 41

4.5.2 Maximal Geometric Diversity 41

4.5.3 Random Seed 42

4.6 Parameter Tuning 42

4.6.1 Confidence Metric 42

4.6.2 Filter Metric 43

4.6.3 Stopping Criterion 44

4.6.4 Seed Selection 45

4.7 Groundtruth Budgeting 45

4.8 Evaluation Metrics 46

4.8.1 Evaluation Strategy 47

5 Results **49**

6 Discussion **55**

6.1 UMLS Metathesaurus Performance 55

6.2 Corpus Similarity 55

6.3 Model Selection 56
6.4 Seed Size 56
6.5 Groundtruth Budgeting 57
6.5.1 Analysis of our Results 57
6.5.2 Error Analysis 58
6.6 Comparison to other i2b2 Challenge Entrants 59

7 Next Steps **63**

8 Contributions **65**
8.1 Summary of our Contributions 65

Bibliography **67**

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

1-1 An example sentence taken from a discharge summary for a patient. 19

2-1 An example of a sentence from a patient discharge summary with semantic concepts labeled. 23

3-1 An example of the output of Daniel Sleator's Link Grammar parser on the sentence "Chest x-ray on Monday revealed pneumonia." The links indicate relationships between the connected words in the parse tree. 31

4-1 Plots of the micro F_1 and macro F_1 scores in the Treatment-Problem (TrP) relation category versus the number of examples bootstrapped for each confidence metric listed in Table 4.2. 43

4-2 Plots of the micro F_1 and macro F_1 scores in the Treatment-Problem (TrP) relation category for the `Top-N`, `Ratio-N`, and `Random-N` versus the number of examples bootstrapped. 44

5-1 Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 10% training data and budgets 14%, while the bootstrapper starts with all 24%. Each use an unlabeled corpus of 76% 50

5-2 Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 20% training data and budgets 12%, while the bootstrapper starts with all 32%. Each use an unlabeled corpus of 68% 51

5-3 Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 30% training data and budgets 11%, while the bootstrapper starts with all 41%. Each use an unlabeled corpus of 59% 51

5-4 Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 40% training data and budgets 9%, while the bootstrapper starts with all 49%. Each use an unlabeled corpus of 51% 52

5-5 Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 50% training data and budgets 8%, while the bootstrapper starts with all 58%. Each use an unlabeled corpus of 42% 52

5-6 Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 60% training data and budgets 6%, while the bootstrapper starts with all 66%. Each use an unlabeled corpus of 34% 53

5-7 Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 70% training data and budgets 5%, while the bootstrapper starts with all 75%. Each use an unlabeled corpus of 25% 53

5-8 Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 80% training data and budgets 3%, while the bootstrapper starts with all 83%. Each use an unlabeled corpus of 17% 54

6-1 An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 79 and 80 is TrNAP, while the bootstrapper claims it is TrAP. 58

6-2 An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 5388 and 5389 is TrIP, while the bootstrapper claims it is TrAP. 59

6-3 An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 5730 and 5728 is TrCP, while the bootstrapper claims it is TrAP. 59

6-4 An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 6798 and 6799 is TrNAP, while the bootstrapper claims it is TrAP. 60

6-5 An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 11271 and 11274 is TrIP, while the bootstrapper claims it is TrAP. 60

6-6 An example of a case where the bootstrapper is able to distinguish the relationship where the budgeter cannot. The budgeter incorrectly claims the relation for this sentence between concepts 18050 and 18051 is TrCP, while the bootstrapper correctly claims it is TrAP. 61

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

2.1 Relations between concepts appearing in Figure 2-1 24

2.2 Occurrences of concept types in the 2010 i2b2/VA challenge corpus 24

2.3 A list of each relation group, their relations and occurrences in the 2010 i2b2/VA challenge corpus. 25

4.1 A summary of our kernel and its parameters used with `libsvm` 37

4.2 A breakdown of the different confidence metrics we have evaluated. 39

4.3 A list of the example filter metrics we have evaluated. 39

4.4 A summary from [21] of the various types of semantic drift in bootstrapping classifiers. Potential drift indicates the possibility of drift, drift indicates that there was drift, and abrupt drift means that the drift was significant. . 40

4.5 A table describing the cosine similarity in feature-space between each pair of relation clusters in the Treatment-Problem (TrP) relation group. 43

6.1 A table describing the cosine similarity in feature-space between each pair of relation clusters in the Treatment-Problem (TrP) relation group limited to only one sentence per line. 56

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

We address the problem of relation extraction in hospital discharge summaries. Sentences with pre-identified concept types (for example: medication, test, problem, symptom) are labeled with the relationship between the concepts. For example, there are five relations between a treatment and problem:

1. Treatment administered for problem. (TrAP)
2. Treatment causes problem. (TrCP)
3. Treatment improves problem. (TrIP)
4. Treatment not administered because of problem. (TrNAP)
5. Treatment worsens the problem. (TrWP)
6. No relation. (NONE)

Using an NLP-based feature-set described in [23], we build on the work of a fully-supervised classifier for relation extraction using the 2010 i2b2/VA challenge corpus in an attempt to find effective weakly-supervised methods for accomplishing the task.

We present a technique for improving performance of weakly-supervised bootstrapping using noisy and very self-similar datasets. We call our technique 'groundtruth budgeting'. Groundtruth budgeting offers improved performance over bootstrapping by simulating the circumstances of active learning. A small portion of the training seed is separated for use as a budget of new training data annotations. After each round of bootstrapping, the classifier can request samples from the budgeted group to be revealed based on those about which it is least confident. This simulates the circumstances of active learning on a reduced portion of the corpus. In some circumstances, groundtruth budgeting both offers better performance than bootstrapping and requires a smaller training set.

1.1 Motivation

1.1.1 Annotation Cost

Sibanda [27] show that fully-supervised systems are capable of good performance at relation extraction on medical text. However, the cost of acquiring the training data for such a system is quite high.

In the medical domain, training data comes at a serious premium because the only qualified annotators are nurses and doctors. Deploying fully-supervised systems will not scale, because the volume of annotations required is prohibitively expensive. We aim to drastically reduce the implementation cost of relation extraction systems in the medical domain by developing weakly-supervised or unsupervised approaches to the task.

Weakly-supervised systems make use of a small seed of human-annotated training data and a broader corpus of un-annotated data to accomplish the learning task. Fortunately an abundance of unlabeled patient discharge summaries – written records of patient interactions with doctors – are available, as most digital medical records are simply digitally transcribed versions of written discharge summaries.

1.1.2 Real World Applications

There are a number of real-world applications for a system that can extract relations from medical data. Medical discharge summaries provide a motivating example of the utility of such a system. A discharge summary is a written description of a patient's interaction with doctors in a hospital. Each patient's medical record contains many of these summaries collected over the patient's lifetime. An example of a sentence from such a discharge summary appears in Figure 1-1.

```
A culture taken from the lumbar drain showed Staphylococcus aureus
resistant to the Nafcillin which he was receiving and he was therefore
placed on Vancomycin.
```

Figure 1-1: An example sentence taken from a discharge summary for a patient.

This sentence indicates that the patient had a `Staphylococcus aureus` infection, and the doctor attempted to treat it unsuccessfully with `Nafcillin`. Finally, `Vancomycin` was administered to treat the infection. Humans can easily understand the meaning behind these summaries because they intuitively grasp the semantic role each word plays with respect to the other words in the sentence. Relationship extraction is the task of extracting these relationships that humans can understand intuitively. In this case, the relations that a human can easily infer are that `Nafcillin` and `Vancomycin` are both drugs used to treat `Staphylococcus aureus` infections. In this case, the infection had become resistant to `Nafcillin`, so the treatment did not work. Furthermore, a `lumbar drain` is a test that can be used to check for the presence of `Staphylococcus aureus` infections.

If a system were able to extract the meaning behind these interactions automatically, then it would be possible to build a summarizer that could produce a succinct summary of the various symptoms, problems, medications, and tests that a patient has exhibited, received, or undergone. Doctors are busy, and are often unable to read a patient's entire medical history because the salient facts are buried in the text of these dense medical summaries. A summarizer would allow a doctor to quickly absorb an overview of the information first, and then get specific details later.

Such a system for relationship extraction could also be used for medical research purposes. Given patient discharge information from hospitals across the country, the system could be used to research commonalities between patients afflicted by certain conditions, to discover contraindications of drugs that were not previously known, or to provide objective measures of a drug's performance nationwide.

To make such a system practical to implement, it is important to minimize the amount of human annotation it requires. Groundtruth labels come at a premium in this domain because the only qualified annotators are doctors and nurses. To this end, it is necessary that we pursue weakly-supervised approaches to building these systems in order to make them cost-effective.

1.2 Self-Similarity of Medical Language Corpora

The 2010 i2b2/VA challenge corpus is unique in that it contains mostly medical jargon and is composed of very similar sentences. In fact, using our set of NLP features and considering the centroid of every relation cluster in feature space shows the relation clusters are highly overlapped, with cosine similarities of over 0.9.

Given these similarities, it is difficult to build a model that can achieve good performance on all of the classes. Models are often biased towards the majority class. Using weakly-supervised approaches such as bootstrapping have not yielded significant improvements, and often cause the classifier to quickly diverge in the accuracy of its predictions.

We believe the highly self-similar nature of our corpus to be contributory to this. In comparison to the fully-supervised system, the performance is worse – particularly the performance on the minority classes. Consequently the macro F_1 score is not as high as its fully-supervised counterpart.

1.3 Contributions

To address this lack of performance in the minority classes, we present a new technique called 'groundtruth budgeting'. With this technique, we set aside a portion of the groundtruth at the beginning of bootstrapping and gradually add it in as the classifier is bootstrapped. The samples our classifier chooses to be added are the samples on which it has the lowest confidence in its predictions. Using this technique, we demonstrate improvements over regular bootstrapping and fully-supervised models with equivalent amounts of training data.

Additionally, we present the results of a variety of experiments we have conducted in the exploration of this problem domain. In particular, we have found that our attempts to diversify the seed of labeled data used in bootstrapping have not produced anything better than the choice of a random seed. Finally, we compare our results to other recent relation extraction results from the Proceedings of the 2010 Fourth i2b2/VA Shared-Task and Workshop.

1.4 Thesis Structure

We begin in Chapter 2 with a description of the problem and the related work we reference. We follow this in Chapter 3 with a description of the experimental infrastructure we have designed. In Chapter 4 we describe our experimental methods, experiment parameters, and our proposed technique, groundtruth budgeting. We present our results in Chapter 5. We follow this with a discussion of our results in Chapter 6, a set of next steps in Chapter 7, and a summary of our contributions in Chapter 8.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Background and Related Work

2.1 Problem Overview

The learning task we address is relation extraction on medical language. The medical documents we are focusing on are patient discharge summaries provided from two corpora. A small subset of patient discharge summaries in these corpora have been annotated to label every concept in the text with an appropriate medical concept.

For example, words that indicate a medication have the semantic concept `med`. Medical problems in the text are all tagged `problem`. The task is to determine the relationship between the concepts in the medical text.

The medical discharge summary given shown in Figure 1-1 has a number of medical concepts contained within it. The same discharge summary with each of its concepts labeled can be found in Figure 2-1.

$$\texttt{A } \underbrace{\texttt{culture}}_{test-1} \texttt{ taken from the } \underbrace{\texttt{lumbar drain}}_{med-2} \texttt{ showed } \underbrace{\texttt{Staphylococcus aureus}}_{prob-3}$$
$$\texttt{resistant to the } \underbrace{\texttt{Nafcillin}}_{med-4} \texttt{ which he was receiving and he was therefore}$$
$$\texttt{placed on } \underbrace{\texttt{Vancomycin}}_{med-5}\texttt{.}$$

Figure 2-1: An example of a sentence from a patient discharge summary with semantic concepts labeled.

As discussed in §1.1.2, the concepts within Figure 1-1 have a number of relations between them. The learning task is to build a model that can predict these relations given a seed of human-annotated training data. The labels for the sentence in Figure 1-1 are listed in Table 2.1.

Relations

Treatment Addresses Problem (TAP)	`Staphylococcus aureus, Vancomycin`
Treatment Does Not Address Problem (TNP)	`Staphylococcus aureus, Nafcillin`
Treatment Addresses Problem (TAP)	`Staphylococcus aureus, lumbar drain`
Test Reveals Problem (TRP)	`Staphylococcus aureus, culture`

Table 2.1: Relations between concepts appearing in Figure 2-1

Our system is capable of taking sentences from discharge summaries and predicting the relationships that occur between pairs of concepts within its sentences. For each corpus of discharge summaries, we split each discharge summary into sentences and annotate the medical concepts occurring within each sentence. Since we aim to evaluate the performance on relation extraction, we assume that the concepts are labeled with 100% accuracy by humans.

2.1.1 The 2010 i2b2/VA Challenge Corpus

Our reference corpus, the 2010 i2b2/VA challenge corpus is a human-annotated corpus of concept and relation annotations on medical discharge summaries taken from 3 different hospitals: Beth Israel Deaconess Medical Center, Massachusetts General Hospital, and the University of Pittsburgh Medical Center. The concept types are outlined in Table 2.1.1 and the relation groups with their corresponding relations are outlined in Table 2.1.1. The relations and concepts from our training set are taken from a total of 349 hospital discharge summaries and progress notes and comprise over 30673 lines and 3033 relation

Concept Type	Count
Test	7365
Problem	11967
Treatment	8497

Table 2.2: Occurrences of concept types in the 2010 i2b2/VA challenge corpus

Relation Group	Relation Type	Count
Problem – Treatment	Treatment administered for problem (TrAP)	1423
Problem – Treatment	Treatment causes problem (TrCP)	296
Problem – Treatment	Treatment improves problem (TrIP)	107
Problem – Treatment	Treatment not administered because of problem (TrNAP)	106
Problem – Treatment	Treatment worsens problem (TrWP)	56
Problem – Test	Test reveals problem (TeRP)	1734
Problem – Test	Test causes problem (TeCP)	303
Problem – Problem	Problem indicates problem (PIP)	1240

Table 2.3: A list of each relation group, their relations and occurrences in the 2010 i2b2/VA challenge corpus.

annotations. Our gold-standard test data set is taken from 477 discharge summaries and is composed of 45053 lines with 5142 relations.

2.2 Related Work

Concept extraction in the domain of medical language has been explored by Sibanda et al. in "Syntactically-Informed Semantic Category Recognizer for Discharge Summaries"[27]. Furthermore, there are production systems such as `MedLEE`[18] capable of concept extraction within medical text such as these discharge summaries. Due to the significant body of work in this area, we are not addressing the task of concept extraction in this domain. Instead, we assume that the concept extraction task has succeeded perfectly on our corpora and focus our attention on the problem of relation extraction.

There are two main examples of generalized relation extraction in the literature. In [2], Agichtein and Gravano describe a novel technique for generating patterns in documents and a system for extracting relations from large collections of documents called `Snowball`. `Snowball` is a weakly-supervised approach that takes a small number of human-annotated examples and generates patterns based on those which are capable of locating and extracting relations in a corpus of documents. They build upon the work of Brin's Dual Iterative Pattern Relation Expansion (`DIPRE`). [10] While general-purpose, both `Snowball` and `DIPRE` are vulnerable to the generation of invalid patterns, since both automatically expand their database of patterns based on regular expressions. Zelenko, et al. [33] explore kernel methods for relation extraction on `person-affiliation` and `organization-location` relations.

The foremost examples of weakly-supervised relation extraction in the literature are evaluated on the Automatic Content Extraction (ACE) program's Relation Detection and Characterization (RDC) task. Relevant to our task, Zhang [34] presents multiple bootstrapping classifiers for relation extraction, `SelfBoot`, `BootBagging`, and `BootProject` evaluated on the ACE-2 RDC corpus. We use `SelfBoot` as the basic implementation of our bootstrapping system. Additionally, the author proposed a maximum entropy-based confidence metric for multi-class relation extraction. We have implemented this and compare it to a number of other metrics in §4.4.3.

Others have addressed the task of relation extraction in the field of medical text. For example, Abulaisha and Deyb[1] have built a relation extraction system for biological relations using abstracts from MEDLINE, an index of scientific journal articles. In the same area, Bundschus, et al. [11] implement a relation extraction system using Conditional Random Fields (CRF) which produces good performance on the extraction of semantic relations between diseases and treatments in PubMed abstracts. While both of these systems deal with medical language, the language is well-formatted and grammatical since it is taken from scientific paper abstracts.

There are few existing attempts to do relation extraction from text in medical discharge summaries. Medical discharge summaries are often un-grammatical and jargon-rich, making them especially hard to use with normal Natural Language Processing (NLP) techniques. Following the 2010 Fourth i2b2/VA Shared-Task and Workshop, there are now examples of relation extraction attempts on our corpus including a maximum entropy and bootstrapping approach by de Bruijn, et al. [16]. We compare their findings to ours in Chapter 6.

We build on the work of Uzuner et al. [23] in which they present a semantic relation (SR) classification system for medical discharge summaries. They show fully-supervised macro-F scores of 74% to 95% on multiple corpora of pairs of concepts taken from sentences in medical discharge summaries. We extend their work by presenting weakly-supervised approaches to the same task, but with the 2010 i2b2/VA challenge corpus.

2.2.1 Bootstrapping

We use the bootstrapping method of incorporating a corpus of unlabeled data into a model based on a small seed of training data. One of the earliest NLP papers to use bootstrapping is Yarowsky's method [32] of weakly-supervised bootstrapping for word sense disambiguation (WSD). Using an initial seed of information from the dictionary, the algorithm records collocations of a word and words contained in each of its dictionary definitions. Then, an unlabeled corpus is used to expand its measurements of the occurrences of each sense of a word. A log-likelihood test can then be used to measure the probability of a given sense of the word occurring in a given context.

Another paper we are inspired by is Blum and Mitchell's description of co-training. [7] In co-training, two classifiers are trained on the same training set. Each classifier has a different set of features which it extracts from examples. The classifier's feature sets are formulated such that they offer a complementary picture of the data set. The classifiers then bootstrap each other by iteratively making predictions on unseen examples and passing those on to the other classifier. Since each classifier has a different 'view' on the data in question, they are able to accomplish more than a single bootstrapper on its own.

Gr and Bengio [19] propose a system for weakly-supervised learning using maximum-entropy regularization, further motivating the use of max-entropy for the learning task. Erkan et al. [17] have implemented a bootstrapping system with transductive SVM's (TSVM) which makes use of both word vectors, edit distance, and cosine similarity in feature space as features for learning. Others have attempted graph-based weighted, k-Nearest-Neighbor (kNN), or label propagation (LP) approaches, as in Chen, et al.[12]; however, as we describe in §4.6.3, geometric approaches we attempted have all failed due to the highly redundant nature of the medical language in our corpus. Finally, Chen, et al. [13] demonstrate relation extraction using unsupervised clustering methods.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

System Overview

We extend the work of Dr. Özlem Uzuner and Tawanda Sibanda in "Semantic Relations for Problem-Oriented Medical Records" [23]. They present a system that achieves F_1-measures of 74% to 96%, depending on the relation group. The system makes use of a variety of lexical, syntactic, and surface features of the training corpus to train a soft-margin, fully-supervised SVM.

In [23], they present a system called the Category and Relationship Extractor (CARE). To evaluate and experiment with our weakly-supervised approaches to the same learning task, we have re-implemented the relation-extraction portion of the CARE system in Python, using the PyML [6] and libsvm [14] machine learning libraries. PyCARE is a flexible and modular machine learning system that can be easily configured to evaluate relation-extraction experiments on large corpora.

We run our system and all of our experiments using PyCARE on modern 64-bit x86 quad-core machine with four 2.4 gigahertz cores running a GNU/Linux derivative operating system.

3.1 PyCARE

The PyCARE library, our implementation of the relation-extraction parts of CARE using Python, PyML, and libsvm, is a modular system made up of the following components:

1. A number of parsing libraries for each specific corpus file format.
2. A number of tools for cleaning and pre-processing raw discharge summaries.
3. A processing pipeline for sending unlabeled data through knowledge sources such as the Unified Medical Language System (UMLS).
4. A library of tools for extracting lexical and surface features from sentences in a given discharge summary.
5. An interface for describing learning experiments to be run by defining the type of learning tool to use, the corpus to train or test from, and the feature extraction methods to use.
6. A set of tools for determining the outcomes and statistics of test results.
7. A set of tools for measuring the confidence of classifiers on unlabeled data for use in bootstrapping.
8. A set of tools for measuring the changes to a corpus as data is added, e.g. semantic drift tracking tools.

3.2 Pre-processing

For all of our experiments, both labeled and unlabeled corpora must be pre-processed in order to prepare them for use by the lexical and semantic feature extractors in `PyCARE`. The following steps are taken to pre-process our corpora:

- ☐ Sentence-Breaking / Tokenization
- ☐ Stemming
- ☐ Part-of-Speech Tagging
- ☐ Link-Parsing
- ☐ UMLS CUI and Relation Identification

3.2.1 Sentence-Breaking / Tokenization

All of the discharge summaries in our corpus were sentence-broken and tokenized by human annotators. We integrated `MetaMap` [5] for automated sentence breaking and tokenization; however, since our corpus has already been sentence-broken and tokenized, we

```
            +---------S--------+
+--AN--+-Mp-+-Js-+           +----Os----+
|      |    |    |           |          |
Chest.n x-ray on Monday revealed.v pneumonia.n
```

Figure 3-1: An example of the output of Daniel Sleator's Link Grammar parser on the sentence "Chest x-ray on Monday revealed pneumonia." The links indicate relationships between the connected words in the parse tree.

chose to use the provided annotations in order to reduce noise that automated approaches introduce.

3.2.2 Stemming

After sentence breaking, the stemmed version of every word in each sentence is fetched using the National Library of Medicine's lexical toolkit LVG [22].

3.2.3 Part-of-Speech Tagging

The part of speech of every word is determined by using Eric Brill's part-of-speech tagger [9].

3.2.4 Link Grammar Parsing

Many of our system features make use of Sleator, et al.'s Link Grammar parsing system [28] for ascertaining a deep-parse of every sentence in our corpora. Therefore, we run the Link Grammar parser on every sentence as a pre-processing step.

An example of a sentence parsed using the link grammar parser is shown in Figure 3-1. The Link Grammar parser often produces multiple parses for a given sentence. To resolve these cases we simply take the first result returned by the parser.

3.2.5 UMLS CUI and Relation Identification

Since we have experimented with using Unified Medical Language System (UMLS) relations as features to our system, another pre-processing step is the determination of all

UMLS concept unique identifiers (CUIs) associated with each concept (e.g. medication, problem, test) in our corpora. For this, we use `MetaMap`. Next, to find all the UMLS relations between CUIs in our corpus, we use a batch Structured Query Language (SQL) query using the `2010aa` version of the UMLS.

Chapter 4

Methods

The main problem with the fully-supervised system presented in [23] is that it relies solely on labeled data. Labels for medical data are expensive to produce. Unlabeled data, on the other hand, is in abundance. Therefore a weakly-supervised approach to the task – one that makes use of both labeled and unlabeled data – is ideal. In order to develop a feasible approach using this model, we have evaluated a number of relation-extraction experiments which make use of unlabeled data combined with a small kernel of labeled data. The unlabeled data we use has 100% accurate concept labelings, since we aim to isolate and study the performance of relation-extraction only.

4.1 Support Vector Machines

Our system makes heavy use of Vladimir Vapnik's Support Vector Machine (SVM) learning model. [30] We use the standard soft-margin, linear formulation of the SVM defined by the following optimization problem:

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}||\underline{\theta}||^2 - C\sum_{i=1}^{n}\xi_i \\
\text{subject to} \quad & y_i\left(\underline{\theta_i}\cdot\underline{x_i}+\theta_0\right) \geq 1-\xi_i, \quad i=\{1,\ldots,n\} \\
& \xi_i \geq 0, \quad i=\{1,\ldots,n\}
\end{aligned}
$$

This soft-margin formulation of the SVM was given by Cortes and Vapnik in [15]. We use a simple linear kernel with unity slack cost to avoid over-fit issues.

SVMs are traditionally binary classifiers. However, our system must classify multiple relations at once. Duan and Keerthi [8] present an empirical study of the benefits and disadvantages of various SVM multi-class classifier configurations. We use the One vs. All (OVA) method of training our SVM first proposed by Vapnik in 1995 [30]. To produce a single classifier for n different classes on a data set L, the OVA formulation of the SVM trains n SVMs. For each SVM, L is partitioned into those examples that have label i, and those that do not. SVM i is then trained on that partition of L. When all n SVMs are trained, then to give a prediction for the OVA classifier, all n SVMs are used to predict the labels for an unseen example. The example is labeled by the classifier that both positively identifies the example and for whom the distance to the hyperplane for the example in feature-space is maximized.

4.2 Lexical and Syntactic Features

The following are the lexical and syntactic features used by the fully-supervised system described in [23] which we have re-implemented as part of `PyCARE`.

For any two candidate concepts in the text c_1 and c_2, the following binary features are applied:

- ☐ Existence of concepts occurring between c_1 and c_2 in the text.
- ☐ The words occurring between c_1 and c_2 in the text.

- ☐ The verbs occurring between c_1 and c_2 in the text.
- ☐ Left and right lexical trigrams of each concept.
- ☐ Verbs preceding each concept.
- ☐ Verbs following each concept.
- ☐ Syntactic paths between the concepts via the Link Grammar Parser.[28]
- ☐ Left and right syntactic bigrams connected to each concept via the Link Grammar Parser.

Each feature that indicates the presence or absence of a word or group of words augments the feature vector for the c_1 c_2 pair by a number of binary features the size of the word vocabulary. Any words that are indicated by the feature take a value of 1 in this array. Since each one of these features expands the total feature space by a magnitude at least equal to the vocabulary size, the combination of all of these features results in a feature-space of over one million dimensions.

4.3 Bootstrapping

The core weakly-supervised technique we use is the bootstrapping method. We configure `PyCARE` to train an SVM classifier S on a small seed of labeled data L. The size of this seed is dynamically configurable, as described in §4.4.2. The kernel we use is described in §4.4.1. Using S, we make predictions on unseen examples taken from a large, preprocessed, corpus of unlabeled data, U.

In order to evaluate relation extraction on its own, we must have 100% accurate concept annotations on our unlabeled data. This is not true of the unlabeled corpora available to us, so in order to facilitate we split the entire corpus of labeled data available to us into a training set L and an unlabeled (but concept-annotated) set, which we use as U. All the known relation labels for U are erased prior to running the system. The split size is a parameter to our system.

Each unlabeled example in U is labeled by S and ranked by S's confidence in the label it assigns. Since S is a multi-class SVM, there are a variety of different ways to judge the confidence of the classifier. We call this parameter the 'confidence metric', and detail the various metrics we have implemented in §4.4.3.

After rating the examples by confidence, we then filter S's predictions on the unseen examples by some function of their confidence score. The choice of function is a parameter, and we call it the 'example filter metric'. We describe the various filter metrics we have implemented in §4.4.4.

The examples remaining after filtering along with their predicted labels are then merged with L into a new training corpus L'. A new classifier S' is trained on L' and the performance of S' is evaluated by either 10-fold cross-validation or evaluation on a held-out portion of the training data.

After each round of bootstrapping, various metrics are measured in order to determine when the bootstrapping process should cease. We call these 'stopping criteria', and describe them in §4.4.5. Once a stopping criterion has decided the classifier will not benefit from addition of more unlabeled data, the bootstrapping process halts and the final classifiers S' is evaluated on the held-out test set. For our experiments, our held out test set is the 2010 i2b2/VA challenge corpus gold-standard test set.

Using this setup, we are able to quickly prototype and execute a variety of bootstrapping experiments.

4.4 System Parameters

4.4.1 Choice of Kernel

In our experiments, we seek to examine the effect of different classifier configurations on weakly-supervised relation extraction from the text. Since searching for an optimal setting of system parameters is combinatorial in the number of parameters, we seek to minimize the number of parameters to our system. To help ease the combinatorial explosion of parameters, we choose a fixed linear SVM kernel with unity slack cost. This has an added benefit of allowing us to avoid the over-fit issues to which polynomial and radial-basis function kernels are vulnerable. We use the same kernel parameters as those in [23]. We

Kernel Function	$(\gamma \cdot \underline{u}' \times \underline{v} + \text{coeff0})^{\text{degree}}$
degree	1
γ	1
coeff0	1
Slack Cost (C)	1

Table 4.1: A summary of our kernel and its parameters used with `libsvm`

summarize these kernel parameters as the arguments we supply to `libsvm` in Table 4.1.

4.4.2 Seed Size

The seed-size parameter describes the amount of training data provided to a bootstrapping classifier as a percentage of the initial training set. We represent this parameter as the percentage of labels taken from a training corpus. Labels are selected from the training corpus proportionally to their distribution within that corpus. This is accomplished for each relation by setting a goal number of samples for the relation based on the relation's presence in the initial training corpus, and then using reservoir sampling [20] to select the samples at random from the corpus.

4.4.3 Confidence Metrics

Active learning, originally proposed by Angluin [3], is a technique founded on the key realization that a machine learning model can use less training data and require fewer training iterations if it is able to characterize the type of examples about which it needs more information, and receive labels for those examples from an oracle. Thompson, et al. [29] explore active learning in the context of natural language processing and information extraction.

The most common method by which active learning is accomplished is the creation of a confidence metric for the learning model to apply to unseen examples. In SVM's, this is usually a function of the distance from the hyperplane in feature-space. For a binary classifier, the closer an example is to the hyperplane in feature-space, the less confident the classifier is about that example. Since we make extensive use of multi-class classifiers, we must find a way to extend this to multiple relations. There is a lot less writing in the literature about how to combine their parameters into a stable confidence metric. Vlachos [31] presents a variety of confidence metrics for active learning with multi-class classifiers.

During bootstrapping, predictions of the classifier on unseen examples are ordered by the classifier's 'confidence' in the label it has assigned to each example. All of our relation classifiers are One vs. All classifiers [30], therefore each prediction on an example is nothing more than n distances to the decision boundaries of each of the n relation versus all sub-classifiers. The confidence metrics we have evaluated are all functions of the distances of examples from these decision boundaries.

We have evaluated 9 different confidence metrics. Vlachos [31] proposes a number of confidence metrics based on the n One vs. All distances-to-hyperplanes. They use these confidence metrics for Active Learning with multi-class SVMs. We evaluated five of these confidence metrics with our bootstrapping approach. Additionally, we evaluate two other metrics from Zhang [34]: confidence based on the entropy of the label probability distribution and a confidence score based on the cosine similarity of the example to other labeled examples in feature space. The entropy of the label probability distribution, as described in Zhang [34], is given for a sample x in terms of $p_i(x)$, or the estimated probability that example x has label i.

$$H(x) = -\sum_{i}^{C} p_i(x) \log p_i(x)$$

We describe each confidence method in detail in Table 4.2.

4.4.4 Example Filter Metrics

After the classifier has rated each unseen example with a label and confidence score, the unseen examples must be filtered down to the set of examples that will be incorporated into the training set. Since there are a number of ways to accomplish this, it is a parameter of our system. Table 4.3 details the various filter metrics that are available to our system.

Name	Description
random	Confidence scores are chosen randomly in the range $[0.0, 1.0]$. Used as a baseline.
hyperplane-min	Confidence scores are the value of the minimum distance to the decision-boundary for the n classifiers that comprise the One-Vs-All multi-class classifier.
hyperplane-max	Confidence scores are the value of the maximum distance to the decision-boundary for the n classifiers that comprise the One-Vs-All multi-class classifier.
hyperplane-sum	Confidence scores are the sum of the distances to the decision-boundary for the n classifiers that comprise the One-Vs-All multi-class classifier.
hyperplane-product	Confidence scores are the product of the distances to the decision-boundary for the n classifiers that comprise the One-Vs-All multi-class classifier.
hyperplane-difference	Confidence scores are the difference between the maximum distance to the decision-boundary and the sum of the other distances for the n classifiers that comprise the One-Vs-All multi-class classifier.
label-entropy	The confidence scores are the entropy of the label probability distribution as described in [34].
centroid-cosine	The confidence scores are calculated by the cosine of the feature vector for the unseen example with the centroid in feature-space of every example with the given label.

Table 4.2: A breakdown of the different confidence metrics we have evaluated.

Top-N	The top-n rated examples by confidence are incorporated into the training set
Ratio-N	n examples are taken from the unseen examples, the label composition of these n examples are selected to match the distribution in the original training set
Random-N	n examples are chosen uniformly at random from those available.

Table 4.3: A list of the example filter metrics we have evaluated.

Potential Drift	$0 \leq \text{dist} \leq \min(r_{old}, r_{new})$
Drift	$\min(r_{old}, r_{new}) \leq \text{dist} \leq \max(r_{old}, r_{new})$
Abrupt Drift	$dist \geq \max(r_{old}, r_{new})$

Table 4.4: A summary from [21] of the various types of semantic drift in bootstrapping classifiers. Potential drift indicates the possibility of drift, drift indicates that there was drift, and abrupt drift means that the drift was significant.

4.4.5 Stopping Criterion

While augmenting the training set with examples from the unlabeled corpus, there will be examples that do not contribute to the performance of the system. We must determine when to stop the bootstrapping process so that we do not incorporate these examples into the training set. To this end we have examined a number of criteria by which we may decide to stop the bootstrapping process. The first of these is the detection of classifier 'drift'. The general idea is that while adding unseen examples to the training data, the meaning of the classifier labels may shift from what the ground-truth reflects.

Li, et al. [21] describe this by characterizing three forms of drift: 'potential drift', 'drift', and 'abrupt drift'. These 3 metrics are determined by treating the labeled examples of a given label in the training corpus as a cluster, and by measuring the change in both the cluster's centroid in feature-space and the distance to the example furthest from the centroid. We consider the centroid and max-radius of each relation cluster before and after each round of bootstrapping, and the distance between the old cluster centroid and the new centroid. The three cases of drift are described in Table 4.4.

In addition to the techniques described by Li, we measure the relative change in similarity between each label. We measure this by detecting overlapping radii of clusters in feature-space, as well as by measuring the cosine similarity between the cluster centroids.

4.5 Seed Selection

We have evaluated a variety of techniques for diversifying the initial seed provided to a weakly-supervised classifier. In systems such as Zhang's [34], the seed is naively calculated by random sampling of the unlabeled set. Since a bootstrapping classifier will start with drastically fewer training examples, it is crucial that these samples are selected in order to maximize performance of our system. We have referred to a number of techniques in the literature for inspiration in this area.

Qian, et al. 2009 [25] present a technique for splitting up their initial unlabeled set into various strata using a stratification variable, such as the relation class, using the label probability distribution as a prior probability of examples taking on each relation. In Qian, et al. 2010 [24], they present a refined version of this technique which does not rely on the label probability distribution, but instead determines the stratification using various clustering methods.

Since weakly-supervised approaches to relation extraction use far less training data than fully-supervised, it is crucial that we choose the most effective samples from the data available to us to be labeled and used as the starting seed. We have experimented with multiple different ways of composing a seed in search of a technique that produces results better than randomly selecting it.

4.5.1 Maximal Cosine Diversity

One method we have developed is to maximize the diversity of samples added to the seed in feature-space. As we select samples, we take their feature-space representation and calculate the centroid of the entire seed. From this centroid, we select samples whose similarity to the centroid is minimal. The similarity metric we use is the cosine between the two vectors in feature space. By building a seed this way, we aim to build up a collection of diverse examples so that our seed has good coverage of feature space.

4.5.2 Maximal Geometric Diversity

This approach is identical to the maximal cosine diversity approach except the similarity metric we use is geometric distance in feature space.

4.5.3 Random Seed

As a baseline for comparison, we also built in the capability to select a seed fully at random.

4.6 Parameter Tuning

To determine the optimal setting of the parameters described in Sections §4.4.2, §4.4.3, §4.4.4, §4.5 and §4.4.5, we evaluate every possible setting of each feature while holding all others constant. While we have often watched how parameters such as seed size and confidence metric vary together, we have never evaluated every setting of each parameter, because this would take a prohibitive amount of processing time for each experiment we run. Instead, we identify overall winners among the confidence metrics, filter metrics, and stopping criterion. We always examine the system at different seed sizes because this is essential to ensure we understand how the system changes as we change the amount of labeled and unlabeled examples fed to the system.

4.6.1 Confidence Metric

To determine the best confidence metric, we evaluated a plain bootstrapping model in which all parameters were held equal except the confidence metrics. We tested every confidence metric at varying seed sizes, and the `hyperplane-difference` metric produced the best overall performance. In Figure 4-1, we plot the performance of bootstrapping on a training corpus of 50% the size of the 2010 i2b2/VA challenge corpus, with a held out set of 15% and an unlabeled set of 35%. The figure shows a significant spike in the performance of `hyperplane-difference` as compared to the rest of the confidence metrics. Additionally, in all of our other tests, `hyperplane-difference` has generally had even and positive performance. Accordingly, we will use `hyperplane-difference` as the best setting for our confidence metric parameter. We find this metric to be a straightforward generalization of the binary SVM case.

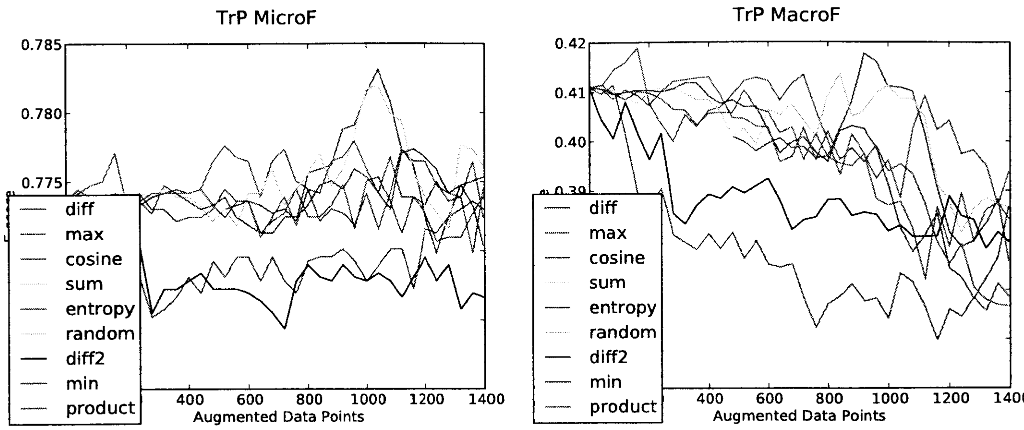

Figure 4-1: Plots of the micro F_1 and macro F_1 scores in the Treatment-Problem (TrP) relation category versus the number of examples bootstrapped for each confidence metric listed in Table 4.2.

	TrAP	TrCP	TrNAP	TrWP	TrIP
TrAP	1.00	0.98	0.98	0.97	0.96
TrCP	0.98	1.00	0.97	0.96	0.96
TrNAP	0.98	0.97	1.00	0.96	0.95
TrWP	0.97	0.96	0.96	1.00	0.96
TrIP	0.96	0.95	0.95	0.96	1.00

Table 4.5: A table describing the cosine similarity in feature-space between each pair of relation clusters in the Treatment-Problem (TrP) relation group.

4.6.2 Filter Metric

In our experiments, the most straightforward example filter metric, `Top-N` performed significantly better than `Random-N` and `Ratio-N` by over .03. In Figure 4-2, we plot the performance of bootstrapping on a training corpus 40% the size of the 2010 i2b2/VA challenge corpus with a held out test set of 20% and an unlabeled set composed of the remaining 40%. The confidence metric employed in this comparison is the `hyperplane-min` metric. Based on the results shown in Figure 4-2, we chose the `Top-N` metric as the best setting for the filter metric parameter.

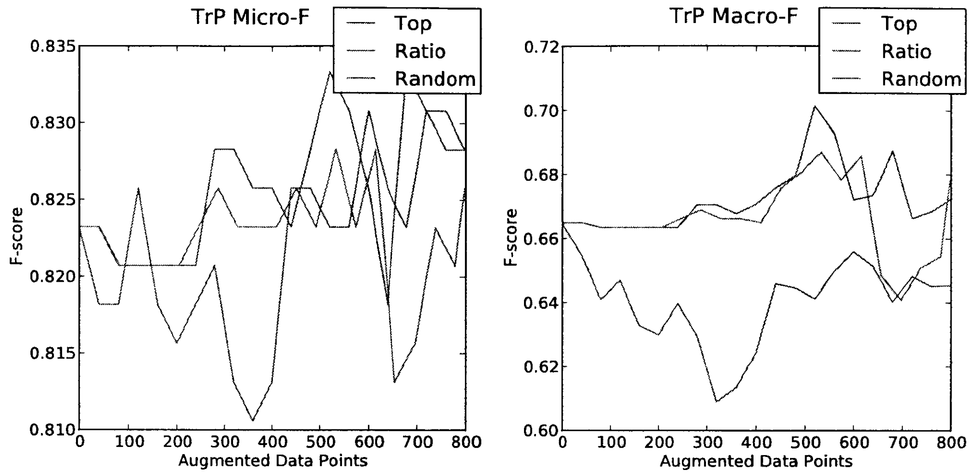

Figure 4-2: Plots of the micro F_1 and macro F_1 scores in the Treatment-Problem (TrP) relation category for the `Top-N`, `Ratio-N`, and `Random-N` versus the number of examples bootstrapped.

4.6.3 Stopping Criterion

As we have mentioned in §1.2, the 2010 i2b2/VA challenge corpus is highly self-similar. We did not realize the extreme nature of this until we began to look at how a stopping criterion could be implemented to detect semantic drift of the classifier and halt it before performance degrades. In Table 4.5, we tabulate the pairwise feature-space cosine similarity between each relation pair of relation clusters in our data.

Each value in Table 4.5 is calculated as follows: We measure the centroid of every relation cluster by taking the vector sum of every example in feature space with each label. Once we have the centroid of every relation cluster in feature space, we then take the pairwise cosine by calculating the vector dot product of every pair of relations, and divide by the magnitude of each vector. We use this value as the cosine similarity metric between the two relation clusters.

Geometrically, the clusters fully overlap with each other in feature-space. Geometric approaches are not applicable in very high-dimensional spaces; however, the cosine similarity metric is commonly accepted as a reasonable method to judge similarity in high-dimensions. Every relation pair has a cosine similarity of 0.95 or greater. The self-similarity of the 2010 i2b2/VA challenge corpus prevented us from implementing any reasonable stopping criterion based on the drift of relation clusters. Because of this, in every experiment we do not implement a stopping criterion. We run the experiment until all of the unlabeled data is exhausted, recording the performance at each round of the experiment.

4.6.4 Seed Selection

As described in §4.5, we implemented and evaluated a couple of techniques for diversifying the initial seed chosen for bootstrapping. We implemented a seed generator which would attempt to maximize diversity in feature-space by iteratively growing the seed, and at each step choosing samples that will most increase the diversity. We tried to pick samples that were the least similar to those already in the seed via cosine and geometric approaches.

In our testing, we found that a randomly generated seed always outperformed diverse seeds that we attempted to construct. Qian et al. 2009 [25] and Qian et al. 2010 [24] both show that diversifying the seed is a good way to improve bootstrapping performance. We believe the discrepancy to be related to the self-similarity we demonstrated in Table 4.5. If we are choosing samples with the smallest cosine similarity to the centroid of the seed, and all of the examples are highly similar to the centroid of the seed, then the samples chosen will not achieve the goal of diversity. This ultimately suggests a problem with our feature-set and perhaps the learning model we have chosen. We describe this in more detail in §6.3.

4.7 Groundtruth Budgeting

We present a new technique called 'Groundtruth Budgeting' which under certain conditions can improve the performance of weakly-supervised bootstrapping by applying techniques inspired from active learning.

Groundtruth budgeting is similar to straightforward bootstrapping, except we begin by setting aside a portion B of our training set L to be 'budgeted'. During the bootstrapping process, we proceed as described in §4.3. We iteratively train a model S on our training set L. Using S, we make predictions on our unlabeled corpus U, and incorporate the most confident examples into our training set L' as groundtruth. After making predictions on samples taken from the unlabeled set, we then take a group of examples from our budget set B, along with their groundtruth labels, and insert them into L' as well. The order in which samples are chosen from B is also decided by the confidence metric, except the most confident examples are chosen instead of the least confident. We then train a new classifier S' on L', and the process continues.

The main difference between regular bootstrapping and groundtruth budgeting is that in the budgeting case, groundtruth information about the corpus which is the least noisy (as determined by the confidence metric) is added in with the predictions on the bootstrapped data. This helps keep the classifier from diverging as the bootstrapping process is prone to cause. The budgeting process omits uninformative or noisy groundtruth examples because they will not produce good confidence signals from the SVM and will thus be left in the budget reservoir.

As mentioned in §4.6.3, we did not find a suitable stopping criterion. Of course, in both the bootstrapping and budgeting case having an effective stopping criterion is important to save on training time. Otherwise, we must evaluate the entire unlabeled corpus while budgeting or bootstrapping and pick the resulting model which maximizes performance on the held-out set.

4.8 Evaluation Metrics

$$\mathrm{P}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i} \tag{4.1}$$

$$\mathrm{R}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} \tag{4.2}$$

$$\mathrm{F}_{1,i} = \frac{(1+\beta^2) \times \mathrm{P}_i \times \mathrm{R}_i}{\beta^2 \times \mathrm{P}_i + \mathrm{R}_i} = \frac{2 \times \mathrm{P}_i \times \mathrm{R}_i}{\mathrm{P}_i + \mathrm{R}_i} \tag{4.3}$$

$$\mathrm{P}_{\mathrm{micro}} = \frac{\sum_{i=1}^{M} \mathrm{TP}_i}{\sum_{i=1}^{M} \mathrm{TP}_i + \sum_{i=1}^{M} \mathrm{FP}_i} \tag{4.4}$$

$$\mathrm{R}_{\text{micro}} = \frac{\sum_{i=1}^{M} \mathrm{TP}_i}{\sum_{i=1}^{M} \mathrm{TP}_i + \sum_{i=1}^{M} \mathrm{FN}_i} \tag{4.5}$$

$$\mathrm{F}_{1,\text{micro}} = \frac{2 \times \mathrm{P}_{\text{micro}} \times \mathrm{R}_{\text{micro}}}{\mathrm{P}_{\text{micro}} + \mathrm{R}_{\text{micro}}} \tag{4.6}$$

$$\mathrm{F}_{1,\text{macro}} = \frac{\sum_{i=1}^{M} \mathrm{F}_{1,i}}{M} \tag{4.7}$$

To evaluate our system, we measure the precision, recall, and F_1-measure for each label i, as shown in Equations (4.1), (4.2), and (4.3) respectively. Since we are using multi-class classifiers, each label's performance, characterized by P_i, R_i, and $\mathrm{F}_{1,i}$, must be combined into an overall system performance metric. For this we follow [23] by using both the micro-averaged F_1-measure and the macro-averaged F_1-measure as given in Equations (4.6) and (4.7). We use the macro-averaged F_1-measure to judge how well our system performs as an average of the performance of every category, while the micro-averaged F_1-measure allows us to judge how well our system performs as the average of every example.

4.8.1 Evaluation Strategy

In order to evaluate the performance of groundtruth budgeting, we designed the following experiment. First, the full training corpus is partitioned into 3 non-overlapping pieces – an initial training seed r, an unlabeled reservoir of samples with which to bootstrap u, and a groundtruth budget reservoir g. We evaluate with the 2010 i2b2/VA challenge corpus test set, e. Two systems are trained in parallel – one bootstrapping classifier and a bootstrapping and budgeting classifier. The bootstrapping learner begins with training set $r + g$, while the budgeting learner begins with r as its training data and a groundtruth budget of g. Therefore, the bootstrapping learner starts with $\frac{g}{r}$ more training data than the budgeting learner.

Both learners bootstrap in rounds. In each round, both classifiers label X new examples from their unlabeled reservoir u and incorporate them as groundtruth. The examples chosen are picked as those on which either learner is most confident. Additionally, the budgeting learner takes Y examples from b, its groundtruth budget. The examples taken from b are those on which the budgeter's label predictions are the least confident. The budgeter then incorporates the Y examples with their groundtruth labels into its training set. At the end of each round, each learner retrains a new classifier based on its new training set, and tests the classifier's performance on the held-out set. The process continues until the entire unlabeled reservoir has been depleted. The ratio $\frac{Y}{X+Y}$ is the fraction of examples in each round that are added from the groundtruth budget versus the total examples added in each round.

This process allows us to accurately compare regular bootstrapping and groundtruth budgeting side by side, since the only difference between the groundtruth budgeting system and the bootstrapping system is that at each round, the budgeter incorporates more ground-truth into its training set. Since the bootstrapping classifier starts out with $r + g$ as its training set, every example in the groundtruth reservoir g that the budgeting system incorporates into its training set is already in the training set for the bootstrapping classifier. Therefore, the test setup allows us to closely examine the effect on adding the groundtruth budget before bootstrapping or during. By the end of the evaluation, both classifiers will have an identical training set, except for the labels that they have chosen for each unlabeled item that is bootstrapped.

Using groundtruth budgeting, we have achieved improvements of up to 2 percentage points in the performance of our system as compared to a baseline bootstrapping system operating with an equivalent amount of training data, and in some cases, operating with even more training data than the budgeting system. We present and discuss these improvements in Chapter 6.

Chapter 5

Results

To evaluate the performance of groundtruth budgeting, we show comparison between a groundtruth budgeting classifier and a bootstrapping classifier for varying seed sizes with the system parameters described in §4.4 all set to their optimal settings as listed in §4.6.

In Figures 5-1 through 5-8, we plot the results of our evaluation for a number of different seed sizes. The figure title indicates that we are plotting the Treatment-Problem (TrP) relation group, and the 3 percentages indicate the percentage split of the available labeled data into training corpus, groundtruth budget, and unlabeled corpus, respectively. In the y-axis, we show the F_1 macro and micro scores, and in the x-axis, we show the number of unlabeled examples bootstrapped by both systems at each round. The graph shows the bootstrapping and budgeting process continuing until completion, so the final data point listed in each graph is the total number of unlabeled examples that were bootstrapped. The F_1 macro and micro scores reported are obtained by evaluating the resulting model on the 2010 i2b2/VA challenge corpus test set. The performance results on the test set are not fed-back into the system with each round, they are simply logged as an evaluation measure.

The results follow a pattern: For small training sets such as in Figure 5-1, the performance of both the bootstrapping classifier and budgeting classifier diverge, and performance throughout bootstrapping gets worse and worse. This happens because there is not enough training data present at the beginning of bootstrapping for our classifier to be accurate enough to increase its performance via bootstrapping.

For large initial training sets, such as in Figure 5-8, the benefits of groundtruth budgeting compared to bootstrapping are similarly diminished because with a large initial training seeds, both models begin with very good initial performance, so there is not as much room for improvement. Note that the variance between budgeting and bootstrapping in the graph is on a very small performance scale.

However, in the middle ranges of initial seed size, we are able to see the benefits of groundtruth budgeting. For example, in Figure 5-3. Initially, the performance of the budgeter starts over 0.01 points below the performance of the bootstrapper. Since the bootstrapper begins with 11% more training data than the budgeter, this is to be expected.

However, we see that the budgeter is able to quickly rise to levels higher than the bootstrapper within the first 200 examples that are added to the system. This is surprising, since the budgeter achieves over .01 points higher performance than the bootstrapper while it still has far fewer training examples in its training set than the bootstrapper. In the case of Figure 5-3, the budgeter achieved performance levels that matched and exceeded the bootstrapper with roughly 11% less training data in use.

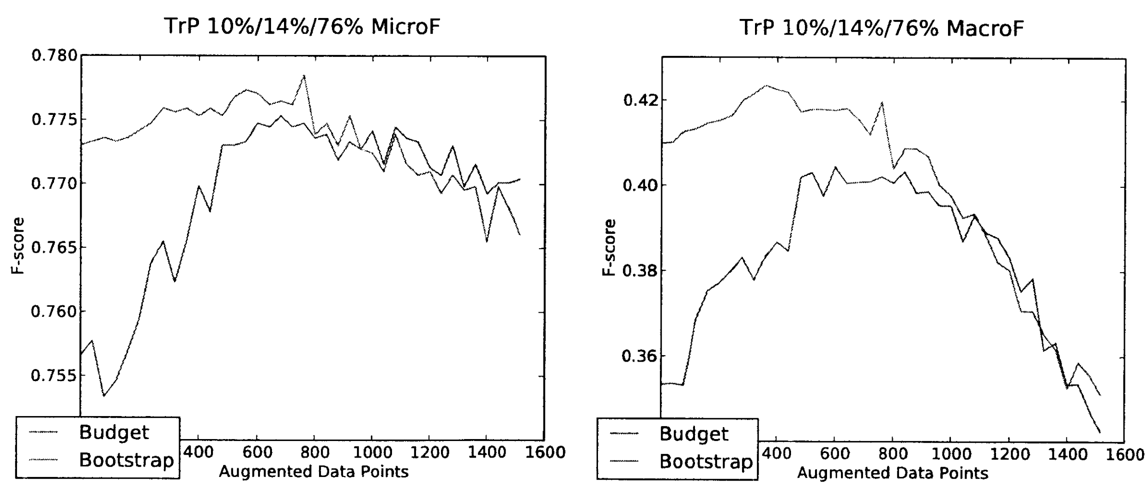

Figure 5-1: Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 10% training data and budgets 14%, while the bootstrapper starts with all 24%. Each use an unlabeled corpus of 76%

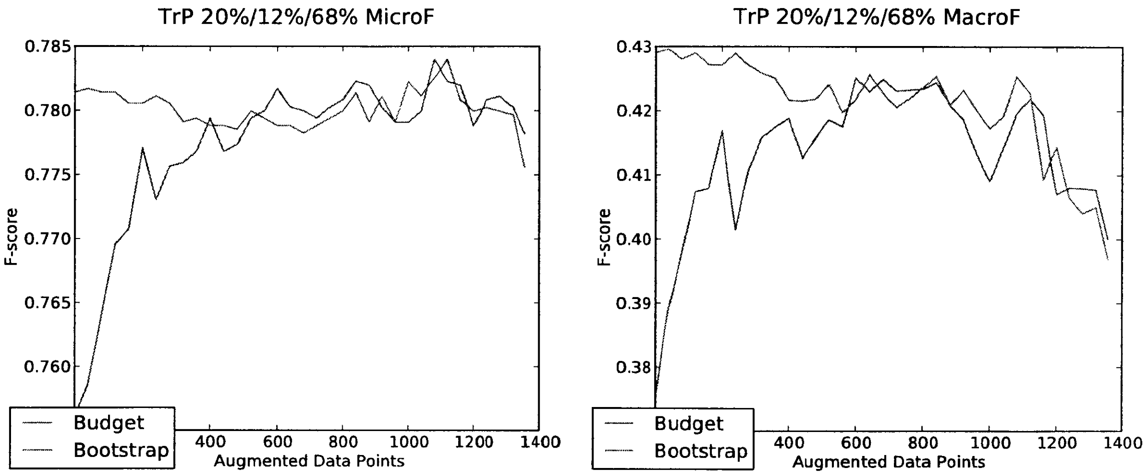

Figure 5-2: Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 20% training data and budgets 12%, while the bootstrapper starts with all 32%. Each use an unlabeled corpus of 68%

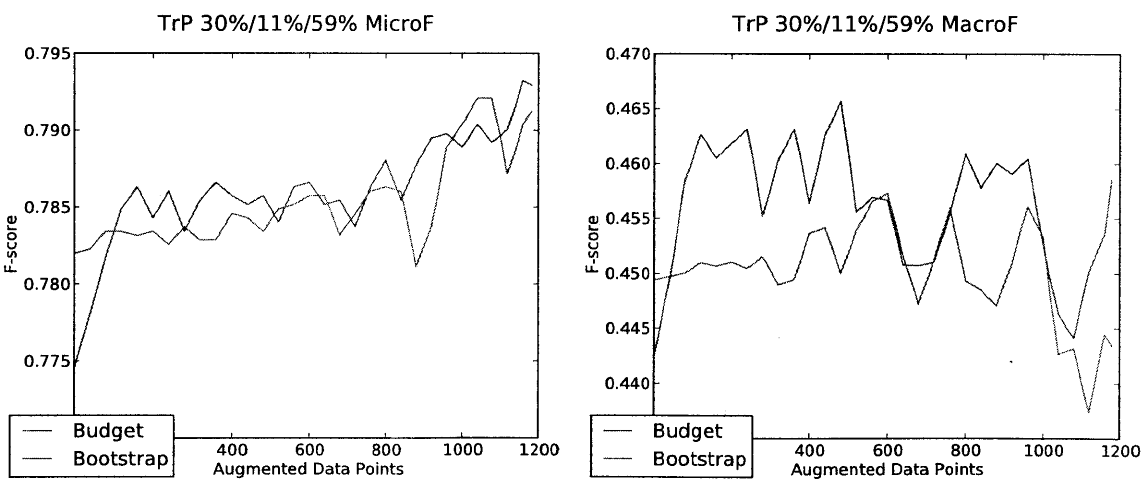

Figure 5-3: Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 30% training data and budgets 11%, while the bootstrapper starts with all 41%. Each use an unlabeled corpus of 59%

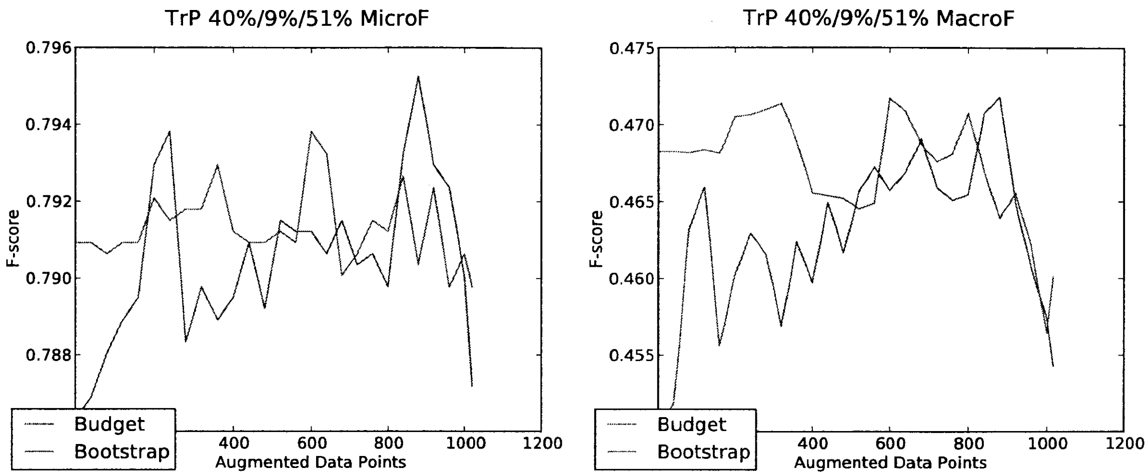

Figure 5-4: Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 40% training data and budgets 9%, while the bootstrapper starts with all 49%. Each use an unlabeled corpus of 51%

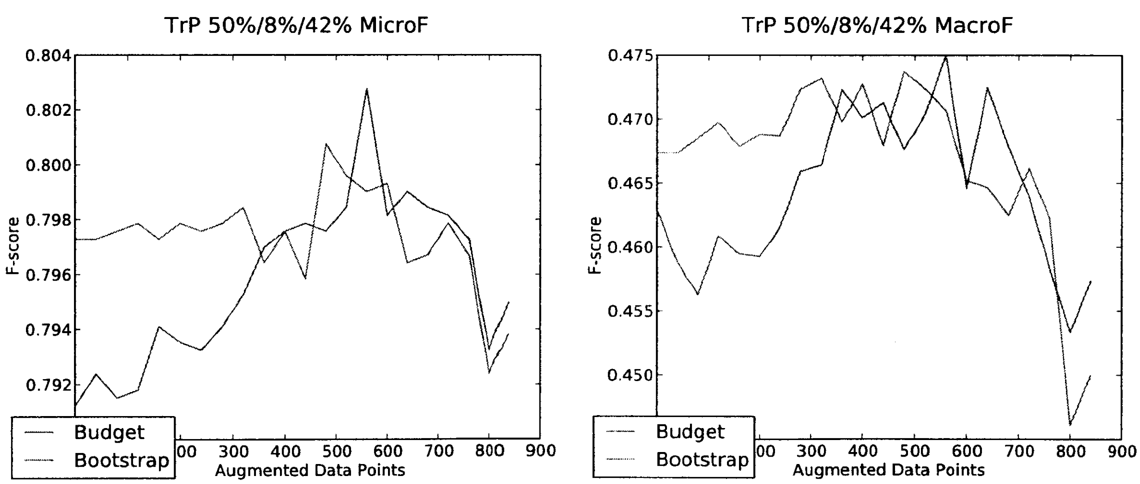

Figure 5-5: Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 50% training data and budgets 8%, while the bootstrapper starts with all 58%. Each use an unlabeled corpus of 42%

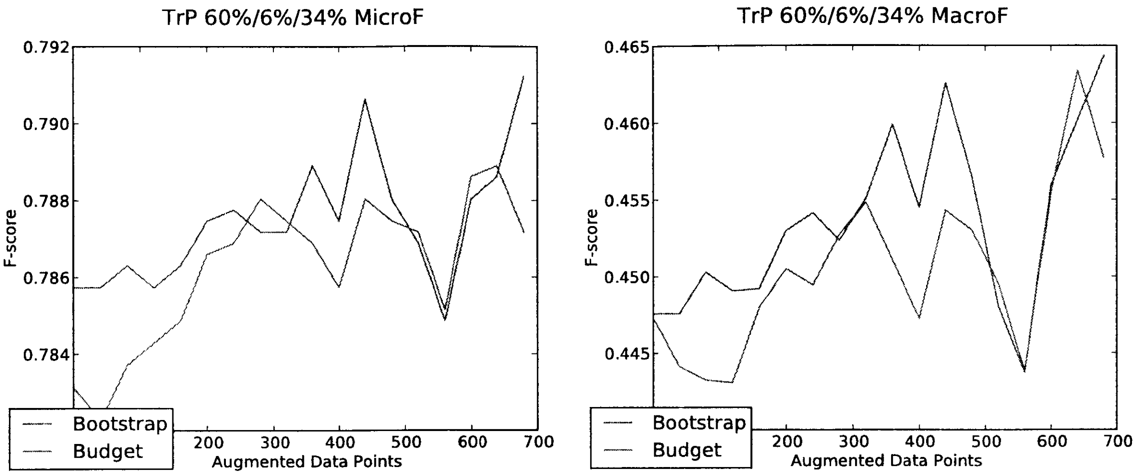

Figure 5-6: Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 60% training data and budgets 6%, while the bootstrapper starts with all 66%. Each use an unlabeled corpus of 34%

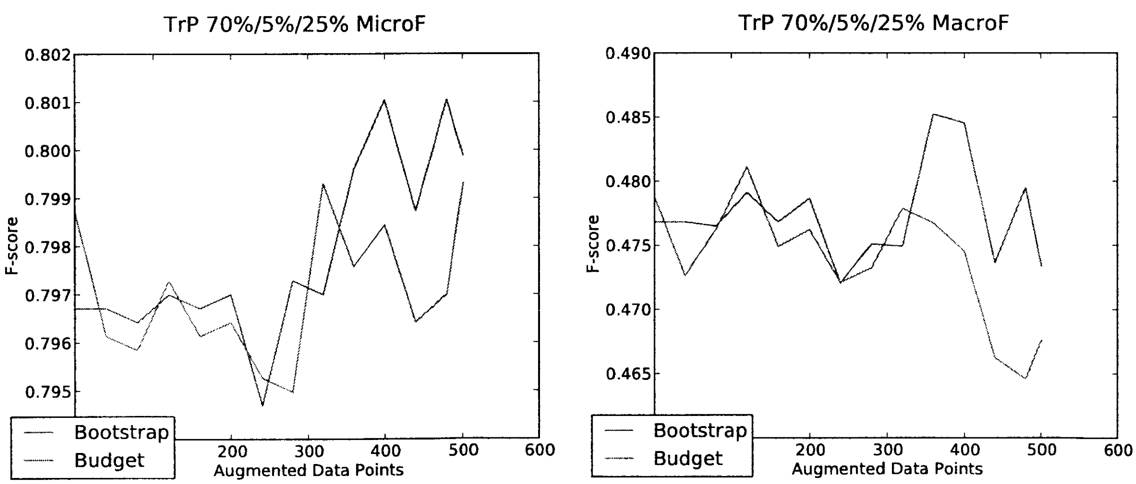

Figure 5-7: Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 70% training data and budgets 5%, while the bootstrapper starts with all 75%. Each use an unlabeled corpus of 25%

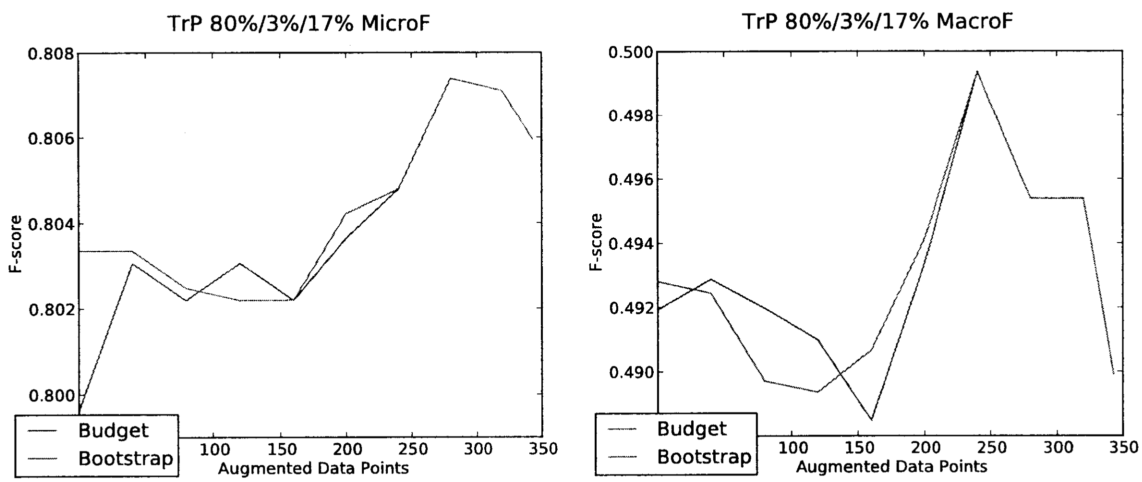

Figure 5-8: Plots of the micro F_1 and macro F_1 scores for bootstrapping and groundtruth budgeting in the Treatment-Problem (TrP). The budgeter starts with 80% training data and budgets 3%, while the bootstrapper starts with all 83%. Each use an unlabeled corpus of 17%

Chapter 6

Discussion

6.1 UMLS Metathesaurus Performance

As described in §3.2.5, we experimented with a system feature that indicated whether the two candidate concepts had any discernible relationship in the UMLS. Unfortunately, the results of this evaluation were not valuable because the recall of concepts within our corpus was too low to accurately evaluate the quality of the UMLS relations for relation extraction. Our experiences in this regard are corroborated by other reports from the 2010 Fourth i2b2/VA Shared-Task and Workshop, such as in Anick et al. [4].

6.2 Corpus Similarity

As discussed in §4.6.3 and §4.6.4, the 2010 i2b2/VA challenge corpus is composed of terse jargon and sentence structures which result in a high degree of self-similarity. Due to these issues, we were unsuccessful in developing techniques for both diversifying the seed selection process and finding useful stopping criteria.

We suspected that another potential cause for the similarity between relation clusters is that sentences from the discharge summaries can have multiple relations per line. In this case, pairs of concepts in the same sentence will have a large amount of redundant context captured in both of their feature spaces. To evaluate this, we measured the cosine similarity of relation clusters, like in Table 4.5, except with only one relation per

	TrAP	TrCP	TrNAP	TrWP	TrIP
TrAP	1.00	0.97	0.95	0.97	0.96
TrCP	0.97	1.00	0.94	0.95	0.94
TrNAP	0.95	0.94	1.00	0.93	0.91
TrWP	0.97	0.95	0.93	1.00	0.95
TrIP	0.96	0.94	0.91	0.95	1.00

Table 6.1: A table describing the cosine similarity in feature-space between each pair of relation clusters in the Treatment-Problem (TrP) relation group limited to only one sentence per line.

line chosen. The results of this experiment are show in Table 6.1. The results of this experiment show that while the similarity's are certainly lower, they still are all greater than 0.9.

6.3 Model Selection

One major weakness in our analysis is that we have not done any model selection. We have used a fixed linear, unity-slack SVM for everything. It is possible that there are other models that could provide more competitive performance, or even yield a different set of tuned parameters that would also not result in a classifier that is significantly over-fit to the data. We were shortsighted in this regard and going forward would like to devote time to this.

For example, de Bruijn, et al. [16] submitted a maximum-entropy based classifier to the 2010 Fourth i2b2/VA Shared-Task and Workshop which achieves performance. Other submissions used different SVM kernels, such as the radial basis function (RBF) kernel.

6.4 Seed Size

We evaluate each bootstrapping approach with a variety of seed sizes. In our experiences, we have found that seed sizes of 20% of our corpus and below often produce divergent bootstrapping behavior in which the classifier gets progressively worse and worse as bootstrapping goes on. This is because the initial model is trained with too little data for it to make accurate predictions. Since the predictions are wrong, a large number of the bootstrapped examples are labeled incorrectly, and the performance of the classifier on the groundtruth set worsens as the meaning of each relation in the classifier drifts.

We find that anywhere between 30% to 45% of our corpus produce positive groundtruth budgeting results. These seed sizes were chosen empirically by comparing the results of many different runs of bootstrapping and groundtruth budgeting.

6.5 Groundtruth Budgeting

Active learning systems are what inspired us to implement groundtruth budgeting. These systems collaboratively interact with a human annotator to produce a model that rapidly improves by pointing the annotator to examples that would be beneficial if the groundtruth were known. The general justification of groundtruth budgeting is this parallel to active learning: Our budget reservoir emulates the case in which a human annotator has annotated a variety of examples, but is no longer available.

6.5.1 Analysis of our Results

In Figures 5-1 through 5-8, we plot the results of our evaluation for a number of different seed sizes. From these plots, we conclude that within this seed range the budgeter is able to add more informative samples to its training set than the bootstrapper due to its dynamic choice of samples from the groundtruth budget via the confidence metric. Since budgeting is an iterative process, the budgeter is able to adapt to what its model is least confident about in each round, while the bootstrapper is stuck with the entire set g from the beginning.

The literature and our intuition both suggest that there should be no penalty to an SVM for starting with more data. On the contrary, it is generally accepted that the more data an SVM can make use of, the better. However, this counter-intuitive penalty is what we observed.

As discussed in §4.6.3, there are a number of properties of our data domain, medical language, which we believe to be contributory to this surprising result. The 2010 i2b2/VA challenge corpus appears to be highly self-similar in nature since it composed mostly of medical jargon and terse domain-specific language. The labels themselves are not incorrect, as our corpus was annotated at high cost by medical professionals; however, the sentences to which the labels apply are often sufficiently self-similar to those of different relation types that in the absence of a large corpus of training data, our classifiers are not able to accurately discern between the minority classes and majority classes in our corpus.

6.5.2 Error Analysis

To get a better sense of what examples are improved by groundtruth budgeting, we examine the system output for the following examples for both bootstrapping and budgeting. We aim to show that the budgeter has improved over the bootstrapped in how well it can recognize minority class examples. We closely examine the output for individual sentences for the system run shown in Figure 5-3. After 240 examples have been bootstrapped and budgeted, we examine the output on the 2010 i2b2/VA challenge corpus test set. In particular, we focus on examples where the bootstrapper got the right answer and the budgeter did not, and vice-versa.

In Figures 6.5.2 through 6.5.2 we present examples in which the budgeter got the correct answer and the bootstrapper did not. In each of these examples, the true relation is a minority class, e.g. TrIP, TrNAP, TrWP, and TrCP. The majority of the differences between the bootstrapper and budgeter in this run are disagreements over the minority classes. These examples would be fairly easy for a layman to discern on his or her own, so it is not unreasonable to assume that the bootstrapper could infer the correct relation from its training data. Why then does it classify them incorrectly?

```
However , <treatment-79> the thalidomide </treatment> was held for two
cycles secondary to <problem-80> atrial fibrillation </problem> .
```

Figure 6-1: An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 79 and 80 is TrNAP, while the bootstrapper claims it is TrAP.

```
We have increased his dose of <treatment-5387> Lantus </treatment> and
<treatment-5388> Humalog </treatment> , and the patient does not feel
<problem-5389> dizzy </problem> anymore .
```

Figure 6-2: An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 5388 and 5389 is TrIP, while the bootstrapper claims it is TrAP.

```
She has <problem-5728> allergies </problem> to <treatment-5729> Morphine
</treatment> , <treatment-5730> Percocet </treatment> , <treatment-5731>
Codeine </treatment> , <treatment-5732> Penicillin </treatment> ,
<treatment-5733> Xanax </treatment> and <treatment-5734> Toradol
</treatment> .
```

Figure 6-3: An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 5730 and 5728 is TrCP, while the bootstrapper claims it is TrAP.

We posit that the reason the bootstrapper cannot discern the correct relation in these examples is because of the aforementioned bias the bootstrapper has towards its majority classes. The same bias has been observed by multiple other entrants to the 2010 Fourth i2b2/VA Shared-Task and Workshop, as we describe in §6.6. The cases in which the budgeter gets the wrong answer and the bootstrapper gets the right answer, as in Figure 6.5.2, are also straightforward. It is unclear why the budgeter would get ones like this wrong; however, the number of examples that the budgeter mis-predicts as non-TrAP are far fewer than the minority class examples it gets right where the bootstrapper gets it wrong. This is evident in Figure 5-3, as after only 200 examples are added, the Macro-F_1 measure, or a measure of the mean of F_1 scores across each relation category, is significantly boosted over the bootstrapping performance.

6.6 Comparison to other i2b2 Challenge Entrants

Many entrants to the 2010 Fourth i2b2/VA Shared-Task and Workshop submitted results for systems designed to perform relation extraction on the 2010 i2b2/VA challenge corpus. To get a sense for how well this system performs in the context of those submissions we now compare our results to those. de Bruijn, et al. [16] submitted the overall best results to the

```
He was not a candidate for <treatment-6798> subcutaneous heparin
</treatment> as we were concerned of <problem-6799> an intracranial
hemorrhage </problem> .
```

Figure 6-4: An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 6798 and 6799 is TrNAP, while the bootstrapper claims it is TrAP.

```
He demonstrated <problem-11271> several asymptomatic electrolyte
deficiecies </problem> ; these were likely due to <problem-11272>
inadequate absorption </problem> following <treatment-11273> his surgery
</treatment> and he responded well to <treatment-11274> repletion
</treatment> .
```

Figure 6-5: An example of a case where the groundtruth budgeter is able to distinguish the minority relationship where the bootstrapper cannot. The budgeter correctly claims the relation for this sentence between concepts 11271 and 11274 is TrIP, while the bootstrapper claims it is TrAP.

relation challenge. They implemented a maximum-entropy based classifier and achieved overall Micro-F_1 performance scores of up to 0.77. They address the issue of imbalance among the relation classes in the 2010 i2b2/VA challenge corpus by downsampling the majority classes. Furthermore, they use MedLine abstracts as an external knowledge source to assess the relationship between concepts using Pointwise Mutual Information (PMI).

Another entrant to the challenge, Anick, et al. [4], show overall Micro-F_1 results of 0.663. They made use of SVM's combined with some hand-crafted rules and a standard set of NLP features to train a multi-class classifier. They report poor results from the integration of UMLS for the concept-annotation task. This confirms our observations in §6.1. Poor concept recall was the main reason we were unable to evaluate the improvements that the UMLS pairwise relation database could offer to our classifier. Anick, et al. [4] also resampled the training sets to avoid classifier bias due to the imbalance in the distribution of labels.

Roberts, et al. [26] train a multi-class SVM with a standard NLP featureset including tokens and bigrams between the concepts; however, they integrate features from external knowledge sources such as Wikipedia. They report Micro-F_1 scores of 73.7 on the relation

```
He still is <problem-18050> hypertensive </problem> and will likely need
<treatment-18051> better blood pressure control </treatment> .
```

Figure 6-6: An example of a case where the bootstrapper is able to distinguish the relationship where the budgeter cannot. The budgeter incorrectly claims the relation for this sentence between concepts 18050 and 18051 is TrCP, while the bootstrapper correctly claims it is TrAP.

extraction task. They too relate that the UMLS features they employed were not helpful for the relation extraction task.

Considering the various other entrants to the 2010 Fourth i2b2/VA Shared-Task and Workshop, we find that the results of both our bootstrapping and groundtruth budgeting classifiers are competitive. In particular, since we did not use the provided unlabeled set and instead opted to segregate our labeled set into a training set and unlabeled set, we see that our system produces roughly equivalent performance to the de Bruijn system while only using 30% of the provided training data. Our system performance with 80% of the training set exceeds their best reported result.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Next Steps

We have far from exhaustively explored the potential of relation extraction in this domain. Our results suggest that the models produced by weakly-supervised SVMs in this problem domain are heavily biased towards the majority class and tend to exhibit poor performance on minority classes.

As discussed in §6.3, we are not satisfied with the extent to which we explored tuning the basic parameters of our SVM – the kernel and its parameters. The use of a linear SVM with unity slack cost may have resulted in sub-par performance. Furthermore, we did not adequately explore feature-selection, since we simply replicated the feature set used in [23]. Due to the severe overlap of all relation clusters in our feature-space, it is possible that our feature-set is to blame for this issue.

We have presented a technique which improves the majority-class bias issues of weakly-supervised SVMs in our problem domain. However, we believe that other techniques should be explored, as our improvements were not enough to make systems like ours useful in practice. In particular, we are attracted to applying the ideas of Blum and Mitchell's co-training [7] to this domain. We built the functionality for this exploration into `PyCARE`, however due to time constraints, were unable to explore the possibilities co-training offered.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8

Contributions

We have demonstrated that given the properties of hospital discharge summaries, common machine learning and NLP techniques for multi-class, weakly-supervised relation extraction are prone to certain weaknesses – in particular a reduction in performance on classes that do not occur frequently in a corpus compared to the performance of the majority classes. We present a technique, 'groundtruth budgeting', which is able to expand the coverage of the resulting SVM models and exhibit improved precision and recall on minority classes during relation extraction.

Using a groundtruth budget while bootstrapping as described in §4.7 can produce better results with less training data than a naive bootstrapper with equivalent amounts of training data. We described our experimental setup for evaluating these results, and presented comparisons to other approaches to relation extraction on the same 2010 i2b2/VA challenge corpus from the proceedings of the 2010 Fourth i2b2/VA Shared-Task and Workshop. Our system performance was competitive with the range of scores from the proceedings of the challenge.

8.1 Summary of our Contributions

1. **Designed** an efficient, maintainable software platform for rapidly describing and developing weakly-supervised machine learning experiments called `PyCARE`.

2. **Presented** a new technique for improving precision and recall performance on minority classes in multi-class relation extraction called 'groundtruth budgeting'.

3. **Evaluated** a variety of different techniques for processes involved in the bootstrapping of multi-class SVMs and shared our findings in comparison with other competitive systems designed for the same purpose.

Bibliography

[1] Muhammad Abulaish and Lipika Dey. Biological relation extraction and query answering from medline abstracts using ontology-based text mining. *Data Knowl. Eng.*, 61:228–262, May 2007.

[2] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *In Proceedings of the 5th ACM International Conference on Digital Libraries*, pages 85–94, 2000.

[3] Dana Angluin. Inference of reversible languages. *J. ACM*, 29:741–765, July 1982.

[4] Peter Anick, Pengyu Hong, Xue Nianwen, and David Anick. I2B2 2010 Challenge: Machine learning for information extraction from patient records. the Proceedings of the Fourth i2b2/VA Shared-Task and Workshop Challenges in Natural Language Processing for Clinical Data, 2010.

[5] Alan R. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: The MetaMap program, 2001.

[6] Asa Ben-Hur. PyML – machine learning in python. http://pyml.sourceforge.net/.

[7] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pages 92–100, New York, NY, USA, 1998. ACM.

[8] Kai bo Duan and S. Sathiya Keerthi. Which is the best multiclass SVM method? an empirical study. In *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pages 278–285, 2005.

[9] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, ANLC '92, pages 152–155, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[10] Sergey Brin. Extracting patterns and relations from the world wide web. In *Selected papers from the International Workshop on The World Wide Web and Databases*, pages 172–183, London, UK, 1999. Springer-Verlag.

[11] Markus Bundschus, Mathaeus Dejori, Martin Stetter, Volker Tresp, and Hans-Peter Kriegel. Extraction of semantic biomedical relations from text using conditional random fields. *BMC Bioinformatics*, 9(1):207, 2008.

[12] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 129–136, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[13] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Unsupervised relation disambiguation using spectral clustering. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 89–96, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[14] Chih chung Chang and Chih-Jen Lin. `LIBSVM`: a library for support vector machines, 2001.

[15] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Mach. Learn.*, 20:273–297, September 1995.

[16] Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. NRC at i2b2: one challenge, three practical tasks, nine statistical systems, hundreds of clinical records, millions of useful features. the Proceedings of the Fourth i2b2/VA Shared-Task and Workshop Challenges in Natural Language Processing for Clinical Data, 2010.

[17] Gunes Erkan, Arzucan Ozgur, and Dragomir R. Radev. Semi-Supervised Classification for Extracting Protein Interaction Sentences using Dependency Parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 228–237, 2007.

[18] Carol Friedman. `MedLEE` - a medical language extraction and encoding system. `http://lucid.cpmc.columbia.edu/medlee/`.

[19] Yves Gr and Yoshua Bengio. Semi-supervised learning by entropy minimization .

[20] Donald E. Knuth. *The art of computer programming, volume 2 (3rd ed.): seminumerical algorithms.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[21] Pei-Pei Li, Xindong Wu, and Xuegang Hu. Learning from concept drifting data streams with unlabeled data. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.

[22] National Library of Medicine. Lvg - lexical tools package.

[23] Özlem Uzuner, Jonathan Mailoa, Russell Ryan, and Tawanda Sibanda. Semantic relations for problem-oriented medical records. *Artificial Intelligence in Medicine*, 50(2):63 – 73, 2010.

[24] Longhua Qian and Guodong Zhou. Clustering-based stratified seed sampling for semi-supervised relation classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 346–355, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[25] Longhua Qian, Guodong Zhou, Fang Kong, and Qiaoming Zhu. Semi-supervised learning for semantic relation classification using stratified sampling strategy. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1437–1445, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[26] Kirk Roberts, Bryan Rink, and Sanda Harabagiu. Extraction of medical concepts, assertions, and relations from discharge summaries for the fourth i2b2/va shared task. the Proceedings of the Fourth i2b2/VA Shared-Task and Workshop Challenges in Natural Language Processing for Clinical Data, 2010.

[27] Tawanda Sibanda, Tian He, Peter Szolovits, and Özlem Uzuner. Syntactically-Informed Semantic Category Recognizer for Discharge Summaries.

[28] Daniel D. K. Sleator and Davy Temperley. Parsing english with a link grammar. In *In Third International Workshop on Parsing Technologies*, 1991.

[29] Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 406–414, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[30] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[31] Andreas Vlachos. Active learning with support vector machines. Technical report, 2004.

[32] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95, pages 189–196, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.

[33] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 71–78, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[34] Zhu Zhang. Weakly-supervised relation classification for information extraction. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 581–588, New York, NY, USA, 2004. ACM.