Detecting Adverse Events in Clinical Trial Free Text

Todd G. Lingren

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2013

Committee:

Imre Solti

Fei, Xia

Program Authorized to Offer Degree:
Department of Linguistics

Acknowledgments

Without the help of many people this thesis would not have been finished. First I'd like to thank my wife Nataline Lingren and four wonderful children who supported me through my absence while working many nights and weekends. I appreciate the encouragement, resources, mentoring and opportunity given to me by my adviser, Dr. Imre Solti. His helpful comments, changes and suggestions over many hours, nights and weekends have helped shaped and strengthened this work. The academic and professional environment at Cincinnati Children's Hospital Medical Center (CCHMC) is second to none. Generous support was given by the Center for Technology Commercialization Innovation Fund and it has funded the larger project to which this thesis belongs.

Thanks go to all of the professors, teaching assistants and staff in the Computational Linguistics Masters of Science program at the University of Washington who helped me to broaden my horizons into computational linguistics and develop skills that have served me well. These include, but are not limited to, Dr. Emily Bender, Dr. Scott Farrar and Dr. Gina Levow. I would like to especially thank Dr. Fei Xia for the mentoring and guidance that she provided me with in class and during the selection of thesis topics and my thesis. Her contributions have been invaluable.

At CCHMC I would also like to thank Dr. John Perentesis who provided insight from his extensive experience as a clinical oncologist to help me understand the clinical trial process. Lori Backus, clinical research coordinator, patiently explained her own experience in coding adverse events through many hours and many dozens of emails. Laura Mayer, clinical research manager, granted me and the annotators access to working spaces in order to annotate the clinical notes. Special thanks are due to the three annotators, Megan Kaiser, Laura Stoutenborough and Jessica Robbins, who tirelessly and patiently put up with the many hiccups and difficulties of the annotation process.

Finally I would like to thank my wife specifically for her love and encouragement. She also lovingly proofread the final copy, but any errors are entirely mine.

University of Washington

ABSTRACT

Detecting Adverse Events in Clinical Trial Free Text

Todd G. Lingren

Chair of the Supervisory Committee:

Imre Solti, M.D., Ph.D

Introduction

In pharmacotherapy cancer clinical trials patients receive frequent outpatient evaluation and monthly inpatient evaluation, as required by the protocol or institutional guidelines. Detection of adverse events (AEs) and adverse drug events (ADEs, caused by the therapy drug) is a manual and costly process and involves chart review. The goal of this thesis is to save resources needed to support a clinical trial by improving the automatic classification of ADEs of clinical notes that document the patient evaluation. To improve the classification I propose using the informativeness of a sentence. The definition of informativeness in this context is any sentence which contains reference to one or more medical conditions. The null hypothesis states that "Classifying sentences into informative and non-informative in the first step of ADE detection will not improve the performance of the ADE classifier".

Data

The 1391 notes from ten patients enrolled in Cincinnati Children's Hospital Medical Center pediatric clinical trials are double annotated for ADEs with adjudication by experienced annotators following the guidance of clinical research coordinators.

Methods

Using the sentence as the base unit for processing, first step of identification of ADE involves the classification of the sentences into informative and non-informative categories. Over 1,200 of the 29,232 sentences contain at least one ADE (positive sentence) and 80% of positive sentences are informative. The results of three support vector machine (SVM) classifiers are compared with one rule classification baseline and one SVM baseline. Three feature selection methods are compared and the chi-square-based approach performs best on the training data.

Results

The experiment classifiers using informativeness of the sentence are significantly better performing than either baseline method. Experiment 2, which used a four-class SVM had a better positive predictive value (PPV) than experiment 1 (80.4% vs. 70.3 %, respectively) which combined results from two classifiers, one for informative and the other for noninformative sentences. All classifiers (experiment and baseline) showed improved results with chi-square feature selection over a naïve feature selection method.

Conclusion

Automated ADE detection in pharmacotherapy clinical trial notes is improved by classifying the sentences by informativeness as a first step.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Table Number Page

1 Introduction

An adverse event is defined as "any new finding or undesirable event that may or may not be attributed to treatment."(Trotti, et al. 2003) The term adverse drug event (ADE) is an adverse effect brought about by a particular medication or drug. In clinical cancer trials, specifically here pharmacotherapy cancer trials, patients are on a course of medication for a set period, usually three to four weeks. During that period patients receive frequent outpatient evaluation and monthly inpatient evaluation, if required by the protocol or institutional guidelines. If the adverse event (AE) experienced by the patient is attributable to the medication course and determined to be too severe, the patient might be taken out of the clinical trial; the ADE would be determined to be an adverse drug reaction (ADR). As depicted in Figure 1, ADEs subsume ADRs. According to Bhavani et. al (2006), "unforeseen adverse effects exhibited by drugs contribute heavily to late-phase failure and even withdrawal of marketed drugs." The subject of this thesis is prediction of an ADE from clinical text; attribution of an ADE to an effect caused by a particular medication is outside of the scope of this thesis.

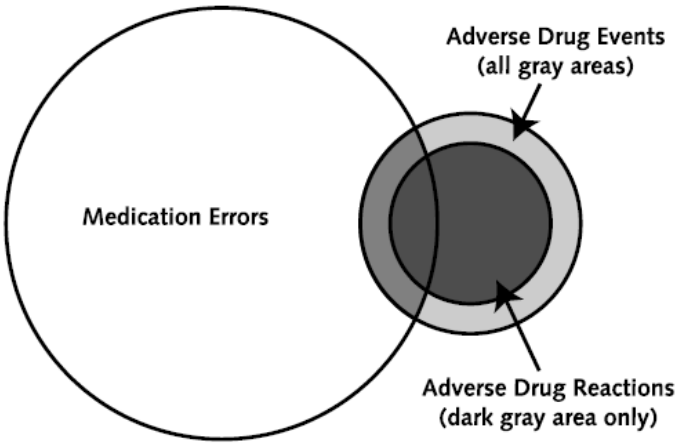

Figure 1: Adverse Drug Events and Adverse Drug Reactions (Nebeker et. al, 2004)

Detecting adverse events quickly from the clinical notes (CNs) has three potential advantages: 1) preventing harm to the patient, 2) saving time identifying the adverse event and 3) saving cost for unproductive trials. The cost to bring a new drug to market is approximately $800 million to $1.2 billion (DiMasi et. al., 2003). Approximately 58% of these drug development costs reflect the resources needed to support clinical trial testing; a large proportion of these costs reflect the expenses related to detailed clinical research coordinator (CRC) review of charts to assess side effects, adverse events, and related reporting. Further evidence for the need of an automated detection system was highlighted in a review of the inconsistent manual reporting methods of 49 National Cancer Institute centers (Belknap et. al., 2010). The primary aim of this thesis is to identify adverse events. The ultimate goal of this work is to reduce the cost of detecting ADE, that a particular pharmacotherapy is determined to cause too many adverse effects, and therefore be either the pharmacotherapy is discontinued for patient or the clinical trial.

To build an adverse event classification pipeline, I propose first identifying which sentences are informative.

H0: Classifying sentences into informative and non-informative in the first step of ADE detection will not improve the performance of the ADE classifier.

Ha: Classifying sentences into informative and non-informative in the first step of ADE detection will improve the performance of the ADE classifier.

The definition of informativeness in this context is any sentence which contains reference to one or more medical conditions (e.g. diseases, disorders, signs and symptoms) since adverse events are inherently medical conditions.

In the next chapter I review the literature relevant to this task. Chapter 3 details the data sets used for training and test purposes. Chapter 4 then describes my methodology and implementation in detail. Chapter 5 lays out the measures used for evaluating the system's results, which are detailed in Chapter 6. I conclude the thesis in Chapter 7 with a summary of my work and a discussion of future work planned on this topic.

2 Literature Review

My task is to build a classification pipeline to detect ADEs in CNs for patients in pharmacotherapy cancer clinical trials. I will accomplish this task by building a sentence-level ADE classifier. In the following section I discuss related work for AE classification. In Section 2.2 I discuss other work related to sentence-level classification.

2.1 Adverse Event Classification

AE detection focuses on a variety of clinical settings including outpatient and inpatient. Setting-specific approaches are more effective for classification (Morimoto et. al, 2004). Relevant literature was selected for automated methods based on inpatient and outpatient data because oncology clinical trial protocols usually include both settings. A survey of laboratory based ADE detection will be limited; while relevant to the larger study, the scope of this thesis only includes text of CNs. For evaluation in the thesis, I will be using positive predictive value (PPV). In a recent systematic search of ADE and ADR detection systems, not all studies reported results, but all 24 studies that reported results included the PPV (Forster et. al., 2012). Chapter 5 provides further details on evaluation methods.

Many ADE classification systems focus on rule-based methods and on data that is numeric or structured. The structured data provide the context to a possible AE. Active monitoring is an implementation of a ADE system and the focus of many reports from the literature.

Using drug-drug interaction as the basis of predicting an ADE, Classen et. al. (1991) developed a drug interaction surveillance program based on medication stop orders, laboratory results and certain antidote ordering. Designing an ADE monitor program from lab results which also determined if harm had occurred, Raschke et al. (1998) reported 53% PPV for the 37 ADEs targeted. An interesting

discovery is that 44% of the true positive alerts were unrecognized by the physician prior to the notification. Jha et al. (1998) combined manual chart review, computer-based monitoring and voluntary reporting for their system, which facilitated a three-way comparison. The automated system used an event module which included a wide variety of clinical events. The 52 ADE screening rules ran under this module and involved "simple medical conditions such as new medication orders, laboratory results above or below certain numeric thresholds, and medication orders associated with changes in laboratory values over time." The automated system captured 45% of all ADEs but only achieved 17% PPV. Ferranti et. al. (2008) wrote 57 rules including antidote drug orders, abnormal lab results and combined the occurrence of certain drug-laboratory combinations, where a given lab result coincided with exposure to a specific drug. Both the voluntary reporting system and automated surveillance system performed poorly, but the automated performed worse (11.0% vs. 5.1% PPV, respectively). Szekendi et al. (2006) developed an active surveillance program with lab results, high risk and antidote medications and sampled the results (8% of alerts) from a three month period. The PPV of preventable AEs was 62% and laboratory-triggered AEs had 44% PPV but the overall PPV was 20%.

In a retrospective evaluation of an ADE computer monitor alert program which included laboratory results, drug orders and discharge diagnosis codes, Hwang et al. (2008) found that the PPV was only 21% while the sensitivity was 79%. The relatively high sensitivity guaranteed that the system was good at identifying errors, but the low PPV required additional effort by a pharmacist to identify or rule out the ADE after the alert. The alert system also predicted the Naranjo severity score (Naranjo et al., 1981), which is a widely used probability scale by using manual gold standard of severity of ADRs.

Rule-based detection from structured data has limited application given the vast content available in the electronic medical record which can exceed what is structured. There remains work to be done on both the narrative clinical text as well as structured and numerical data. Using text-based triggers as a basis of a rule classification system can be effective (Resar et al. 2003). The electronic notes must be available so the text-based trigger systems generally were developed later than some of the previous examples. Trigger-term detection as applied to the clinical text has limited benefit alone. Cao et al. (2003) reported results between 3.4-24% PPV using five trigger terms but the poor performance may have been due to the paucity of triggers. A tool developed by Murff et al. (2003), using 95 triggers, achieved 52%. Focusing on a narrow type of ADEs (spironolactone-related), Huang et al. (2005) were able to achieve a relatively good PPV (63.5%).

Machine learning methods have been recently popular in the area of ADE detection. Some ADE classification systems (Aramaki et al. 2010; Gurulingappa et al. 2012) have focused on drug-condition relationships, where the condition is ostensibly produced by the drug. This is similar to ADR detection, but in these studies, the relationship is most often labeled an association, rather than conclusively determining causality. Cami et. al. (2011) created a drug-ADE association network representation enhanced with pharmacological properties to train a logistic regression model to predict new associations. Tatonetti et. al. (2012) also predicted identified novel drug-drug interactions using the FDA's Adverse Event Reporting System to build profiles of drug interactions. The classification targeted event evidence in electronic medical records based on the International Statistical Classification of Diseases and Related Health Problems (ICD9). He et. al. (2013) and Liu et al. (2012) studied the properties of drugs (e.g., chemical, biological and phenotypic) in order to predict the potential ADEs. For Liu et. al. (2012) the best performing binary classifier was Support Vector

Machines (SVM). They addressed the question of class imbalance by minority resampling. In the case of pharmacotherapy cancer trials, the same drug or combination of drugs is presupposed in the free text notes and contained elsewhere in the medical record (such as medication order information). In examining some of these clinical trial notes, mention of adverse events are far more frequent then the name of the medication required by the protocol and so it is more helpful to focusing on the ADE instead of the drug and should increase the sensitivity.

These methods discussed are useful, but using the clinical text to classify ADEs is extremely important. Bates et al. (2003) suggest that integrating information from physician narratives with automated surveillance methods would increase the number of AEs detected. Because 35% of reviewed ADEs were missed by computer surveillance systems although they were explicitly documented in the dictated reports, Tinoco et al. (2011) concludes that natural language processing is needed. An improved system design is a hybrid, including rules and machine learning to focus on the condition rather than being limited to a condition associated with a particular drug.

Honigman et al. (2001a) used a multimodal approach to detect ADEs in outpatients with diagnosis codes, allergy rules, computer event monitoring rules and text searching. They first developed a data mining tool Micromedix M2D2 (Honigman et. al., 2001b) which consisted of a vocabulary of medical concepts, drug terminology and associations between medical terms and events. The events discovered in the text, medications and laboratory results were linked semantically based on known AEs. The positive predictive value (PPV) for a predicted 25,056 incidents was 7.5% and the sensitivity was 58%. Gurwitz et al. (2003) used the same system as Honigman et al. (2001a) and also included ICD9 codes in a search for drug related events which were classified by severity. PPV

was not reported, but the sensitivity of the computer monitor was 28.7%. Melton et. al. (2005) also developed a hybrid model and wrote 45 rules to map medical conditions predicted by MedLEE to a New York state adverse event reporting classification system. Their individual AE PPV was 44% (sensitivity 25%).

The Cancer Automated Lab-based Adverse Event Grading Service (CALAEGS) is an open source rule based ADE detection system developed by the City of Hope hospital in Los Angeles, CA (Niland et al., 2012) for application in clinical trials. It is closely related to my work because the goal was to detect AEs defined by the Common Terminology Criteria for Adverse Events (CTCAE, Trotti et al., 2003) which is the same goal of the thesis. In retrospective study evaluation for 10 trials (40 patients, 18,603 lab results), CALAEGS detected 99.5% sensitivity of true ADE and also detected that the original manual method missed 15% of ADEs. The scope of the CALAEGS was limited due to the fact that it focuses only on laboratory based AEs (13% of the CTCAE) and 39 laboratory values from structured data. CALAEGS used version 3 of CTCAE guidelines and due to the recent update to version 4, have discontinued development of the system.

2.2 Sentence Classification

In text classification meaningful units in a document are tokens. While in document classification word distribution is important for the document's class prediction, the sentences in a CN represent more meaningful units than tokens. Linguistically, the sentence represents an assertion by the author. As such, it is similar to a dialog act or communicative act and in the case of CNs, represents an assertion of an observation by a clinician. Therefore I have chosen to represent ADE detection as a sentence classification problem. There is a tremendous amount of work done on textual classification as a sentence classification problem, and so I will highlight only the most relevant works.

Beyond bag of words for sentence classification features (McKnight et. al., 2003; Gurulingappa et al., 2012), a “normalization” technique was employed by Naughton et al. (2008). Stop words were removed and in addition to words, noun chunks and POS tags, the authors used higher dimensionality of features to normalize all “numeric references, locations, person names and organisations (sic) to “DIGIT”, “LOC”, “PER”, and “ORG” respectively”. However omitting the original words could cause a loss of basic lexical information. Zhang et. al. (2008) investigated “multi-word” units in order to find “a more meaningful and descriptive lexical unit than the individual word from documents”. Syntax of the sentence is also important and has been used part a of a “fingerprint” algorithm to detect copy-paste errors (Cohen et. al., 2013).

Naughton et. al. (2008) examined sentence-level events with an SVM classifier using modality and words as features. The best F-score used Information Gain (IG) for feature selection and reported 90% for positive events, however their classification used a balanced data set with an approximate frequency of 1.08 events per sentence. Medical abstract structure sentence classification (e.g. introduction, methods, results, conclusion) is a popular sentence classification task which can suffer from the unbalanced positive-negative data when framed as a binary classification (one class vs. rest). Xu et. al (2006) used Hidden Markov Models to detect the most likely inter-sentence structure and McKnight et. al. (2003) found positional information helpful as features. Ko et. al. (2004) used informative measure of sentences in a document to assign feature weights. The term frequency/inverse document frequency was calculated with chi-square values. The similarity of the sentence with the title of the document was combined to apply additional weight to terms that occur in sentences that are more informative.

In a task on a clinical corpus (Cohen et al., 2013) researchers demonstrated success with filtering target text (in this case CN) based on informativeness which was defined based on the content of semantic entities in the text (such as medical condition). Mitchell et al. (2005) implemented an informativeness filtering step by classifying sentences from manually curated protein family reports. Ko et. al. (2004) used text summarization, based on similarity with the title of the abstract, to provide informativeness features for sentence classification.

2.3 Medical Condition

The medical condition classifier that I will use for determining informative sentences was developed by Li et al. (2013), to work on disease, disorder, sign and symptom named entities in Federal Drug Administration (FDA) drug labels. The drug labels (side-effect sections) were double annotated and adjudicated according to the Strategic Health IT Advanced Research Project: Area 4 project (SHARPn) guidelines (Cairns et al., 2011) for disease/disorder and sign/symptom entities. The classifier is a pipeline which preprocesses the text through the clinical Text And Knowledge Extraction System (cTAKES, Savova et al., 2010) and generates feature vectors for a sequence labeling approach using conditional random field (CRF) classifier (MALLET) with default parameters. The feature vectors include n-grams, UMLS entities and semantic types recognized by cTAKES. The performance of the classifier is 89% F-score (92% Precision, 87% Recall) for the combined entitites.

3 Data

Included in the data set for the thesis are CNs for ten pediatric patients (1 to 21 years old) in pharmacotherapy cancer clinical trials at Cincinnati Children's Hospital Medical Center (CCHMC). There are eight separate clinical trials represented which are all Children's Oncology Group (COG) phase 1 or phase 1-2 trials. COG is National Cancer Institute supported clinical trials group, with more than 200 hospitals participating, including CCHMC. Seven of the clinical trials included in the data set are 'small molecule' clinical trials which target certain enzymes in order to inhibit cancer cell growth (Imai et. al., 2006).

Table 1: CN Types in ADE Annotation

Note Type	# of Notes	Sentences with ADE
Consult Note	66	89
Discharge Summaries	20	3
Emergency Department (ED) Notes	23	11
History & Physical Note	45	53
Operative Report	1	1
Oncology Program Treatment Plan Note	11	0
Operating Room Note	13	0
Patient Instructions	103	0
Pharmacy Note	20	5
Plan of Care	155	48
Pre-Op Evaluation	2	0
Procedure Note	7	0
Progress Note	653	993
Referral Image	7	8
Telephone Encounter	265	25
Total	1391	1236

The CNs were created within the EHR system and include progress notes, discharge summaries, emergency department (ED) notes and telephone

encounters. The distribution of the notes is in Table 1. For the ten patients, there are 1,391 notes in aggregate and 1,804 adverse events. There are nearly three quarters of a million tokens in the CNs (726,270) with an average note length of 549 tokens. After preprocessing, the CNs are divided into 29,232 sentences, 1,236 positive for ADE and 27,996 negative. Although many sentences may have more than one ADE, the sentence classification task considers one ADE per sentence. The division into sentences, generation of the features and annotation of the gold standard ADE is described in the Chapter 4.

4 Methodology

For this thesis, I will be focusing on aggregate sentences across all patients available for classification. In clinical application of an automated tool for discovering adverse events, the classification would be targeted for a single patient or more specifically, a selected course for the patient. The reason for focusing on sentences as the base unit, rather than courses, is that the number of courses of treatment for a particular patient varies widely. The number of courses is based on a combination of factors including the protocol requirements and the response of the patient to the medication; especially important is the toxicity monitoring and the attribution to the study medication. For example, a patient may only have one course because the protocol dictated removing them from the trial because of too many ADRs. However, some patients who are able to tolerate the medication could have five, six or more courses. Because the number of courses is determinative for the time (and documentation available) for a patient, there is also a high variability in number of notes per patient. Additionally aggregating sentences from all the patients provides more positive training examples. Using sentences as the base unit for classification still presents challenges with regard the variation of ADE density per patient, but allows for a more balanced classification instance that can be easily judged.

4.1 Adverse Event Gold Standard

For training and evaluation, the CNs have been annotated by clinical research coordinators (CRCs), identifying sentences which contain adverse events. Each protocol or clinical trial has one CRC assigned. The role of the coordinator is to identify toxicities in the electronic health record (EHR) text, communicate with the other study participants, and report severe AEs to COG. After the clinical visit, the CN is authored and entered into the EHR. The CRC accesses this information and prints the documentation (e.g. notes, lab results, radiology

reports, telephone encounter notes) from that visit and collects the printed material into a binder. The sentences or phrases that contain AEs are highlighted as displayed in Figure 2.

PLAN:

-ONC: Continue treatment per ADVL0815, Pazopanib. Currently Cycle 2, Day 13. Obtain weekly labs and will have repeat scans in two weeks and then be seen in clinic on 3/9/11.

-RASH: New left anterior thigh rash. May be irritation from compression stocking. No pain or pruritis, but patient has loss of sensation in this area. Rash was already fading when re-examined later in this visit. Patient and mom will monitor at home and we will re-evaluate at next visit.

=ELEVATED TSH: New finding on labs done 2/9/11. Free T4 remains pending. Possible new onset Hypothyroidism vs Sick Euthyroid Syndrome vs Thyroiditis. Will repeat TSH and free T4 with his end of cycle 2 labs.

PAIN: Palliative Pain Team following. Currently only having pain in LLE, buttock, and groin. Previous pain sites include his chest, right scapula, and right shoulder. Pain has been increasing. Currently using Oxycodone and Valium, but does not like Valium b/c it makes him tired. Seen by Dr. [redacted] from Palliative Pain Service and started on Flexeril PRN, with strong encouragement to at least use it at bedtime. Rx for Oxycodone also renewed.

-CONSTIPATION: Intermittent reports of constipation. Previously taking Miralax, but at present, is able to control his symptoms by taking apple juice when he has not stooled in a couple of days. Stools are not hard or painful at this time.

-SCAR NODULES: Cheloid vs. Tumor. Some of the nodules appear to be disappearing, although the overall size of the affected area does not seem to have changed. . Not appear avid on PET CT.

-VISION CHANGES: Resolved once Crizotinib was discontinued. Now with new complaint of difficulties on focusing on near small objects. No change from two weeks ago. Will continue to monitor and consider referral to Ophthalmology.

Figure 2: ADEs Highlighted by Clinical Research Coordinator

Severe AEs are immediately communicated to the treating physician and PI and will be reported to the COG. All AEs are evaluated in relation to the pharmacotherapy that the patient is receiving. The CRCs assign the adverse event (or toxicity) category and subcategory using the CTCAE as a rulebook. They also assign a grade level (1-5, 5 being death) of the AE and under the guidance of a physician, an attribution severity score. As per COG guidelines, the treating physician must agree with the severity and assign the attribution. The attribution

severity score is the likelihood (0-5) that the adverse event is related to the particular clinical trial drug. The AEs are entered on the toxicity grading report (excel spreadsheet) by date of visit and the severity and attribution is recorded. A portion of a toxicity grading report is shown in Figure 3.

VISIT TYPE:	From C2	OPC	Pt report	Admit
Date	3/2/11	3/9/11	3/12/11	3/14/11
Course		Follow up	Follow up	Follow up
Week		1	1	1
DAY		1	4	6
Performance Score		90		70
Weight, kg		77.1		84.3
% weight difference		8.21%-		0.35%+
Temp Max, C				36.5
Blood Pressure (max: 140/89)		143/93		150/92
CTCAE v. 3.0 TOXICITY:	3/2/11	3/9/11	3/12/11	3/14/11
Fever				
Dermatology/Skin				
Rash/desquamation		0		
Endocrine				
Gastrointestinal				
Constipation		0		
Anorexia				1
Gastrointestinal-other, mouth sores				
Hemorrhage				
Hepatobiliary/Pancreas				
Infect/Febrile Neutropenia				
Inf w nl ANC- Upper airway NOS				
Lymphatics				
Edema- limb			2	
Edema-trunk/genital			2	
Metabolic/Laboratory				
Magnesium- serum high	1	1		
Proteinuria		1		2
Calcium- serum high	1	0		
AST	1	0		
Glucose, high		1		0
Albumin, low				
Sodium, low				
Musculoskeletal				
Joint-function				
Fracture				2
Neurology				
Mood alteration: depression		1		
Neuropathy: sensory				
Mood alteration: anxiety				
Ocular/Visual				
Vision- blurred vision		1		
Ocular/Visual-other, redness				

Figure 3: Toxicity Grading Report

In Figure 4, a screenshot of the CTCAE guide, each adverse event lists criteria for severity as measured in grades 1-5; in all cases, a result of severity of grade 5 is death. The majority of criteria for each grade is in text description of a condition such as, *Hypotension*, grade 3,"Sustained (≥ 24 hours) therapy, resolves without persisting psychologic (sic) consequences". In the other criteria, numerical measurements are given, such as *Left ventricular systolic dysfunction*, grade 2, "Asymptomatic, resting EF < 50-40%; SF <24-15%." The severity of an ADE is important to assessing harm to the patient.

CARDIAC GENERAL						Page 2 of 3
		Grade				
Adverse Event	Short Name	1	2	3	4	5
Hypotension	Hypotension	Changes, intervention not indicated	Brief (<24 hrs) fluid replacement or other therapy; no physiologic consequences	Sustained (≥24 hrs) therapy, resolves without persisting physiologic consequences	Shock (e.g., acidemia; impairment of vital organ function)	Death
ALSO CONSIDER: Syncope (fainting).						
Left ventricular diastolic dysfunction	Left ventricular diastolic dysfunction	Asymptomatic diagnostic finding; intervention not indicated	Asymptomatic, intervention indicated	Symptomatic CHF responsive to intervention	Refractory CHF, poorly controlled; intervention such as ventricular assist device or heart transplant indicated	Death
Left ventricular systolic dysfunction	Left ventricular systolic dysfunction	Asymptomatic, resting ejection fraction (EF) <60 – 50%; shortening fraction (SF) <30 – 24%	Asymptomatic, resting EF <50 – 40%; SF <24 – 15%	Symptomatic CHF responsive to intervention; EF <40 – 20% SF <15%	Refractory CHF or poorly controlled; EF <20%; intervention such as ventricular assist device, ventricular reduction surgery, or heart transplant indicated	Death

Figure 4: Common Terminology Criteria for Adverse Events (CTCAE)

The expert annotations have been applied to printed text as in Figure 2 and translated to the electronic notes by experienced annotators using traditional adjudicated double annotation. Four elements were captured in the translation to electronic annotation: AE category (e.g. Gastrointestinal, Cardiac General, Blood/Bone Marrow), subcategory (e.g. nausea, hypertension, white blood count increased), severity (1-5), and attribution to study medication (0-5). Under the guideline of the highlighted printed notes (Figure 2) and the toxicity grading

report the annotators converted the AE annotation using Knowtator, a plug-in for Protégé. (Ogren, 2006) A screenshot of the electronic annotation is shown in Figure 5. On the left side of the image is a partial listing of the categories of ADEs (a full list is shown to the annotators). Displayed in the center is the CN text and on the right the grade, subcategory and attribution fields. The annotator highlights the appropriate portion of text (as in the highlighted printed CN page), selects the category of AE and enters the grade, subcategory (if any) and attribution severity according to the toxicity grading report. The gold standard used in the thesis is the presence (positive class) or absence (negative class) of one or more ADEs in a sentence.

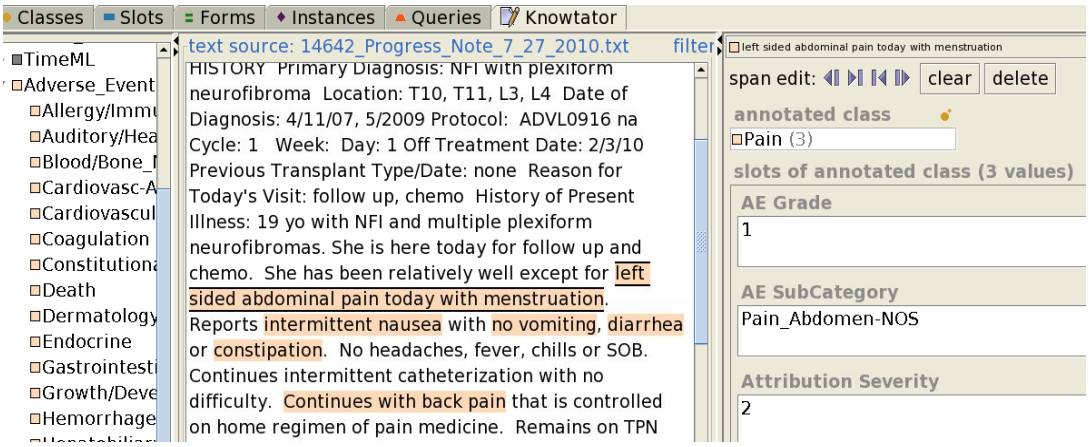

Figure 5: ADE Annotation with Knowtator

Three annotators worked in concert to provide double annotation for every CN. All three annotators are native English speakers. Two annotators had a minimum of one year of linguistic annotation experience on clinical text (clinical trial announcements and CNs) and bachelor's degrees. One annotator (Annotator 2) had clinical experience as a registered nurse (RN) and a nursing degree (Bachelors of Science in Nursing). The third annotator (Annotator 3) had an

associates degree in health information management technology and a certificate as a Registered Health Information Technician (RHIT) from the American Health Information Management Association (AHMIA). A copy of the annotation guidelines is in Appendix A.

I guided the consensus sessions between the annotators. A senior CRC provided advice and clarification when questions were raised by the annotators or when we were unable to come to a decision in the consensus sessions. We asked questions to clarify annotations, determine the relevant spans (e.g. which words and number of words were important for the ADE) and provide adjudication when the subclass of the ADE (e.g. Neurology, Pain, Gastrointestinal) was in question. A goal of future work will be to assign these grade levels to the adverse event detected, but it remains outside the scope of this thesis. In my thesis I will focus only on the binary classification sentences in CNs for ADEs.

Table 2: Inter-annotator Agreement for ADE Annotation

Patient	IAA	# of Sentences	# of ADEs
1	0.588	1,896	247
2	0.533	732	34
3	0.715	880	82
4	0.719	4,688	524
5	0.768	3,953	46
6	0.544	3,872	69
7	0.706	4,886	373
8	0.846	2,395	91
9	0.743	1,759	92
10	0.850	4,171	246
Total	0.748	29,232	1,804

For each patient, a pair of annotators worked to provide double annotation. The per-patient inter-annotator agreement IAA (F-score) and total IAA for all ten

patients is given in Table 2. The per-patient IAA ranges from 53.3% (patient 2) to 85.0% (patient 10).

4.2 Medical Conditions/Informativeness

Using the medical condition classifier pipeline as described in Section 2.3, I supplied a training set of 1,050 CNs. The new training set was used to represent a broad base of CNs similar to the thesis data set. These notes include 100 pharmacotherapy clinical trial notes, which are not part of the thesis data described in Chapter 3 and 950 other clinical notes selected from a random sampling of five million CCHMC clinical notes. The distribution of the 950 notes was similar to those in the project annotation (Table 1) with a few differences.

Table 3: Distribution of CNs in Medical Condition Classifier

Note Type	# of Notes
Asthma Action Plan	22
Communication Body	23
Consult Note	23
DC Summaries	230
ED Medical Student	23
ED Notes	204
History & Physical Note	12
Med Student Note	12
Operative Report	12
Operating Room Note	35
Patient Instructions	19
Pharmacy Note	12
Plan of Care Note	43
Pre-Op Evaluation	12
Procedure Note	12
Progress Note	177
Referral	12
Telephone Encounter	73

There were no Oncology Program Treatment Plan or Referral Image notes in this set. There were additional note types of Asthma Action Plan, Medical Student notes and Communication Body (Table 3).

The gold standard for the training set was annotated in accordance with the SHARPn guidelines and the notes were double annotated with adjudication for two classes of medical conditions (disease/disorders and sign/symptoms). The notes were annotated by the same annotators who worked on the ADE gold standard. In the gold standard there are 8,715 disease/disorder and 8,824 sign/symptom entities, more than twice as many entities as Albright et. al. (2013). Following the SHARPn guidelines, Albright et. al. annotated 4,208 disease/disorder and 3,556 sign/symptom entities to develop training data for NLP algorithms.

As described above, an informative sentence is one which contains a mention of one or more medical conditions. Using the medical condition classifier is a more computationally tractable than identifying key words from a medical dictionary and therefore provides an efficient prediction of medical conditions in the text.

The medical condition classifier was evaluated against a gold standard annotation of 125 notes of one of the patients in the thesis data set. There was no overlap between the 1050 training notes used to build the classifier and the 125 notes of the testing set. The results (F-score) of each and all entity types are shown in Table 4.

Table 4: Performance of Medical Condition Classifier

Entity Type	Total GS	F-score
disease/disorder	2530	0.841
sign/symptom	3578	0.871
All	6108	0.858

4.3 Pipeline

The pipeline system (Figure 6) uses the medical condition classifier to provide informativeness input and cTAKES preprocessing to generate features. Based on informativeness classification, the pipeline builds a candidate list of informative sentences and non-informative sentences to classify for adverse events. The pipeline is graphically depicted in Figure 7, including preprocessing, feature generation, medical condition classifier and final sentence vector creation. More detail follows in Sections 4.3.1 – 4.3.5.

```
Preprocess Text with cTAKES
Detect medical conditions with medical condition classifier
Extract gold standard annotation of ADE
For each sentence
    Collect features
    if sentence contains medical condtion
        Mark as informative
        Add feature to include # of entities predicted
    else
        Mark as non-informative
        Add feature to include 0 entities predicted
    if sentence contains ADE
        Mark as positive
    else
        Mark as negative
  Output all sentences as baseline vector
  Output informative sentences as informative vector
  Output noninformative sentences as noninformative vector
Split vectors into training, development and testing
On each training set perform Feature Selection
    chi-square feature selection
For each classifier
    Train model
    Test model on classifier test set
```

Figure 6: Pipeline Pseudo-Code

4.3.1 Preprocessing

The first step in the pipeline is preprocessing the CNs. This step is performed with cTAKES and accomplishes two goals: sentence detection and creation of an input for the medical condition classifier. The sentence detection module of cTAKES is built using clinical notes, and so it is appropriate for this task. In addition to structural sentence detection, the output of cTAKES includes linguistic and semantic features as a base for feature generation.

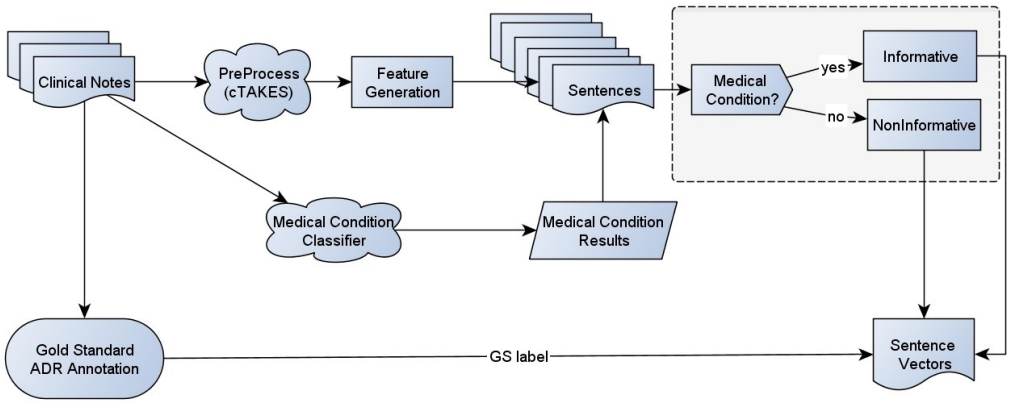

Figure 7: ADE Pipeline - Vector Creation

cTAKES tokenizes the text and generates an xml file which includes entities related to the tokens, sentences, named entities, POS tags, and the named entities also have associated UMLS CUIs, SNOMED-CT codes, and TUIs (semantic types). I wrote a parsing program that generates bigram, trigram token and POS tag sequences. The parsing program takes the sentence prediction of cTAKES and collects the sentence features to create a vector for each CN and for each sentence. Section 4.3.3 describes in detail the features used. There are 29,232 sentences in the CNs, 1,236 are positive for ADE and 27,996 are negative (see Figure 8).

4.3.2 Medical Condition Classification

The second step (line 2, Figure 6) is to detect the disease/disorder and sign/symptom entities with the medical condition classifier. After extracting the gold standard annotation of ADEs, the pipeline collects the features for each sentence and determines the candidate list of informative sentences which have one or more medical conditions detected (Figure 6, lines 4-11) based on the output of the medical condition classifier (Figure 3) which is described in Sections 2.3 and 4.2. After the medical condition classification and assignment of

informative labels, the sentences are logically considered members of one of four groups: informative-positive, noninformative-positive, informative-negative, noninformative-negative. Informative and noninformative refer to the presence of a medical condition in the sentence and positive and negative refer to the binary classification from the gold standard. Positive indicates that one or more ADE is present in the sentence. The distribution of the sentences into these four groups is shown in Figure 8.

Sentences in Clinical Notes			
	Informative	NonInformative	
Positive	985	251	1,236
Negative	3,698	24,298	27,996
	4,683	24,549	29,232

Figure 8: Distribution of Sentences

4.3.3 Features

As discussed, previous sentence classification studies used the sentence as a bag of words. I added features relative to the output from the medical condition classifier, such as the semantic type of the detected medical conditions and the count of medical conditions. The semantic types of all named entities in the sentence are used to define relationships between entities, e.g. medical conditions and anatomical locations.

Figure 9 demonstrates an example with two similar sentences which have different semantic type features. Both sentences are informative because of the inclusion of “pain” (a medical condition), but only Sentence 1 is positive for

ADE. Sentence 2 describes a generic evaluation of pain for the patient whereas Sentence 1 describes a specific instance of pain in a body location (back). A sentence which has the medical condition "pain" and the semantic type for an anatomical location like sentence 2 is more likely to be a description of a specific pain than a sentence with only the former.

1) Pain team evaluated *** today, added Neurotonin

2) Back Pain, Grade 3: History of back pain for several months, that previously required frequent narcotic therapy, but has responded well to massage therapy and rehabilitation services.

Figure 9: Informative Sentences with Different Semantic Type Features

Eleven feature types were used in order to develop a sentence profile for classification. These are shown in Table 5. Feature type categories were part of speech (POS) sequences, n-grams, semantic types of terms, UMLS entities, and number of predicted disease, disorder, sign or symptoms. By using semantic features the vector representation includes a higher level meaning abstraction than a simple bag of words. The original words and word structures in the sentences are maintained to avoid the loss of key lexical information.

The POS features represented single tokens as well as sequences of two and three tokens. If a tag or sequence occurred more than once, only one feature was represented in the text. These POS features provide a flat representation of a syntax tree fragments because syntax patterns have been proven important in creating a "fingerprint" of a text. I hypothesize that descriptions of disease and adverse events are different syntactically from other text.

Unigram, bigrams and trigrams were used without removing stop words that could be important connectors to terms (e.g. "history-**of**-present"). In this way, two token and three token collocations are maintained as features which can

function as important information as a semantic unit. The concept of taking these collocations as features has been termed “multi-word” and is motivated by searching for more meaning than single lexical forms. Compound words and phrases convey more than the sum of the individual words.

cTAKES provided the UMLS entity representation (semantic type, CUI and SNOMED-CT code). These are primarily medical entities, though the breadth of the UMLS terminology includes qualitative concepts like “Positive” (umlsCode=10828004).

Finally, counts of predicted medical conditions (disease/disorder, sign/symptom) were represented as feature types (e.g., disease_disorder:2, Table 5). By the definition of an informative sentence, there is at least one such predicted entity in each of these sentences. If there are zero entities, as in the case of a noninformative sentence, the feature is absent.

Table 5: Features

Feature Type Category	Feature Type	Feature example	Explanation
POS Sequences	1POS; 2POS; 3POS	NN; VBN-NN; DT-VBN-NN	
NGram	Unigram; Bigram; Trigram	clotted; clotted-picc; clotted-picc-line	
UMLS Semantic type	UMLS TUI	umlsTUI=T191	Neoplastic Process
UMLS Semantic Entities	UMLS CUI; SNOMED-CT Code (umlsCode)	umlsCUI=C1705690; umlsCode=274518007	Dosing instruction; Illness (finding)
Predicted medical condition	disease_disorder:x; sign_symptom:y	disease_disorder:2	2 disease/disorder entities predicted

4.3.4 Vector Creation

The third step is to create a vector for each sentence. After taking the results of the preprocessing, the offsets of the sentences are matched with the offsets of the ADE gold standard. As previously mentioned, some sentences have more than one ADE. I process only one ADE per sentence. Although there are 26 categories of ADEs (e.g. Neurology, Pain, Gastrointestinal) the presence or absence of one or more ADEs, regardless of category is treated as a positive case, and absence of an ADE is a negative case). The simplicity of collapsing of all ADEs into a single class justifies the loss of information for this situation.

After assigning the 29,232 sentences with labels (1,236 are positive, 27,996 negative), I generate the features beginning with the base input from the preprocessing step and the medical condition classifier. Once the vectors have been generated, an additional label is added to the class label to identify if the sentence is informative or note. By definition the sentence is informative if one or more medical conditions are predicted (either disease/disorder or sign/symptom). If not, the sentence is determined noninformative.

4.3.5 Machine Learning Classification

For development of the classifier, a development set of 10% was randomly selected from the complete set and a test set of 10% was also set aside for final classification results. The remaining 80% was used with the development set in 10-fold cross validation to determine the appropriate feature size. Error analysis was performed on the development set and iterative improvements were made to the rule based algorithm and features were added.

I used the WEKA (Hall et al., 2009) implementation of an SVM classifier to calculate the results with 10-fold cross-validation. My hypothesis is that by using the informativeness of sentences will improve the classification of ADE. This will be tested by creating four separate classifiers which handle informativeness

differently. The building of the training model for classification is depicted in Figure 10. For each of the training sentences, the medical condition classifier is run to determine if it has a disease/disorder or sign/symptom entity. If so, then by rule the sentence is informative. The baseline classifier is binary and includes all sentences from the training set but does not include information about informativeness. Feature selection is done on the entire training data set for the baseline classifier. Informativeness is ignored in this baseline classifier and it serves as the comparison set for the other three classifiers. The three remaining experimental classifiers all directly consider informativeness. For experiment 1, there are two binary experimental classifiers (labeled "informative", Experiment 1a and "noninformative", Experiment 1b in Figure 11) to separate the classification of informative and noninformative sentences into two classifiers. Each classifier has separate training sentences (informative and noninformative respectively) and feature selection is performed distinctly for each. The systems built were tested as shown in Figure 11. For each of the testing sentences, the medical condition classifier was run to determine informativeness. In order to fairly handle all sentences, the performance of the classifiers in experiment 1a and 1b are combined in the final results (Figure 11, right side "Combined results for Experiment 1). For testing in experiment 1, I ran the medical condition classifier on the test set and determined which sentences were informative. For testing in experiment 1a, I only classified sentences from the test set that were informative. For testing in experiment 1b, I only classified sentences that were noninformative.

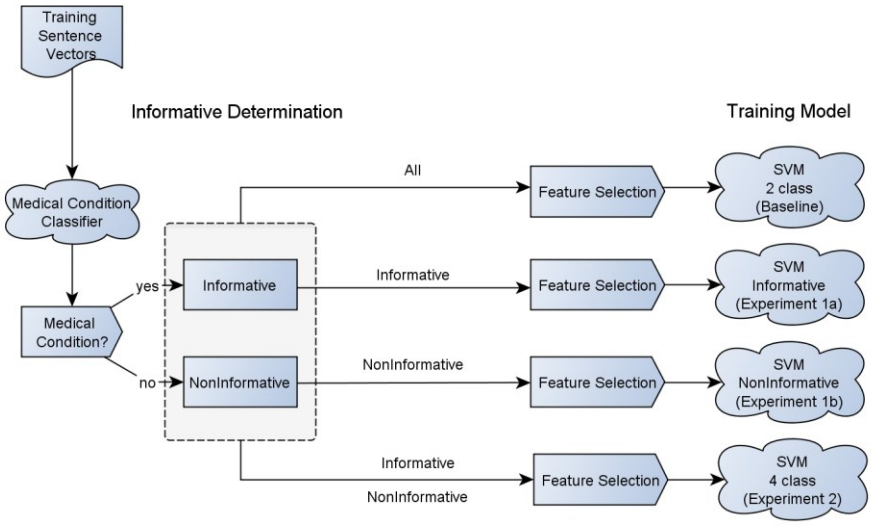

Figure 10: Machine Learning - Training Models

The final classifier (experiment 2) has four class labels: positive-informative, positive-noninformative, negative-informative, negative-noninformative. In this way the informativeness of a sentence provides a split of the positive and negative classes. Feature selection is done on the entire data set, given the four distinct labels. In the final results, positive-informative and positive-noninformative are considered as a positive class and negative-informative and negative-noninformative is considered a negative class.

Given a common test data set, the results are compared between the experiment 1 and the baseline and experiment 2 and the baseline. My hypothesis is that both methods (which use informativeness of a sentence as a factor in classification) will outperform the baseline. I present comparison results in Chapter 6.

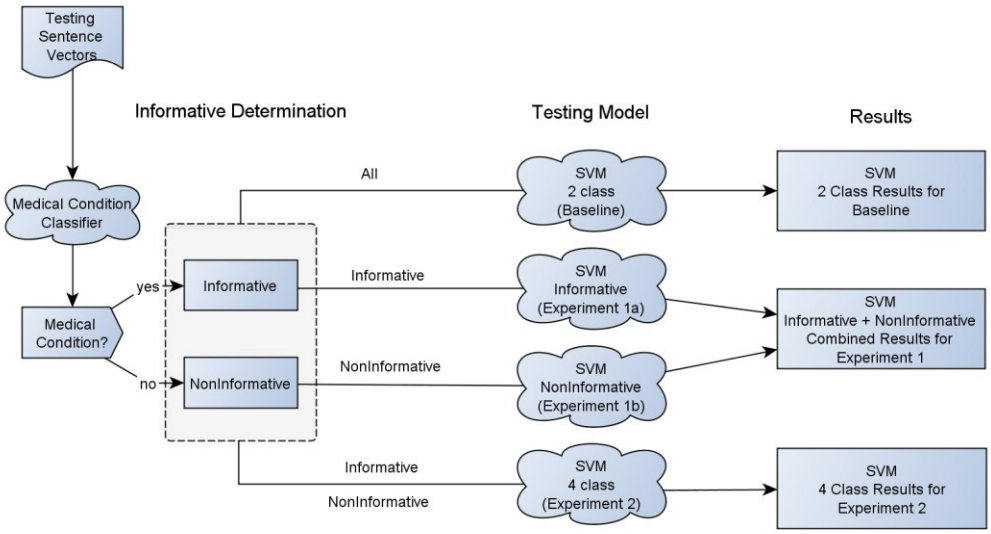

Figure 11: Machine Learning Experiments – Testing Models

In addition to machine learning methods, I wrote a rule based algorithm to classify the sentences. This rule-based classification serves as additional baseline method to compare the experiments and is shown in Figure 12. The results of the rule-based baseline are given in section 6.1.1.

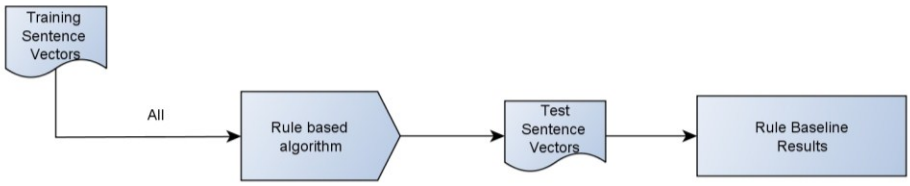

Figure 12: Rule Baseline

4.4 Feature Selection

Even with only 11 feature types, there were more than 255,000 unique features in the baseline experiment set of 29,232 sentences. If there are *n* unique words in a sentence length *k* and n^k unique sequences, the number of unique phrases (e.g. n1_n2_n3) is an order of magnitude higher (Scott et. al., 1999). Certainly not all of these features are useful, possibly providing significant noise to the machine learning algorithm. Rather than develop a logic to reduce them, I decided to experiment with different methods of feature selection. Factoring informativeness into sentence classification did affect the feature space of the sentences but there were still 98,463 and over 240,000 features among the informative and noninformative sentences, respectively.

In order to perform feature selection, I first attempted to convert the sparse real valued features (see Table 5) into a complete binary representation like in Figure 13. I had intended to use included packages in WEKA for feature selection, and the program required a non-sparse vector to do this task. However, with so many features and almost 30,000 sentences, it became intractable to create a complete feature vector before feature selection. I wrote a program to calculate each feature selection method using the sparse feature vector as input and generating an ordered list of top features based on weight. The weight was determined by the individual feature metric: chi-square, information gain, and mutual information. I experimented with these feature selection methods on training data set for all classifiers.

4.4.1 Chi-Square

Chi-square measures the lack of independence between a feature and the class.(Zheng et. al., 2004) If there is complete independence the score will be

zero. The higher scores of this metric indicate which features are most important for determining the class. Given the standard chi-square formula (1) where *O* is the observed value and *E* is the expected value for *i* features and *j* classes, I used formula (2) for feature selection where *t* is the feature for *i* features, *c* is the class for *j* classes.

(1) $$\chi^2 = \sum_{ij} (O_{ij} - E_{ij})^2 / E_{ij}$$

(2) $$\chi^2(t_i, c_j) = \frac{\left(P(t_i, c_j) * P(\bar{t}_i, \bar{c}_j) - P(t_i, \bar{c}_j) * P(\bar{t}_i, c_j)\right)^2 * N}{P(t_i) * P(c_j) * P(\bar{c}_j) * P(\bar{t}_i)}$$

Given that the number of possible classes is 2, the degrees of freedom is 1. The threshold for significance (p-value 0.05) according to the chi-square distribution table is 3.84. I ordered the features by the metric score and removed all features less than 3.84.

4.4.2 Information Gain

Information gain measures the bits of information needed to obtain a correct classification(Yang et. al., 1997) based on the entropy difference when the feature is present versus when it is absent. I ordered the feature set by the metric score.

(3) $$IG(t_i, c_j) = P(t_i, c_j) \log \frac{P(t_i, c_j)}{P(t_i)P(c_j)} + P(\bar{t}_k, c_j) \log \frac{P(\bar{t}_i, c_j)}{P(\bar{t}_j)P(c_j)}$$

4.4.3 Mutual Information

Mutual information measures the amount of information, based on dependence, shared between feature and class. It is 0 when the feature and class are independent. I used the formula (4), ordering the features by the metric score and removed out all zero features.

$$(4)\quad MI(t_i, c_j) = \log \frac{P(t_i, c_j)}{P(t_i)P(c_j)}$$

Given the differences in feature size for each of the baseline and informative vectors, feature selection needed to be performed separately on each (line 17 in Figure 6).

	Features							Class
Sentence	F_1	F_2	F_3	…	…	…	F_j	
1	1	0	0	0	0	1	0	+
2	0	0	0	0	1	0	0	-
3								…
i	…	…						…

Figure 13: Binary Feature Representation

I compared the three methods of feature selection by using the training and development data for each of the four classifiers (Baseline, Experiment 1a, Experiment 1b, and Experiment 2). PPV performance values are shown in Figure 14. In the figure, chi-square feature selection is represented by lines with circles, mutual information is represented by lines with *x* and information gain is represented by lines with triangles. For each classifier, chi-square outperforms the other feature selection consistently so I chose chi-square as the feature selection method for my thesis.

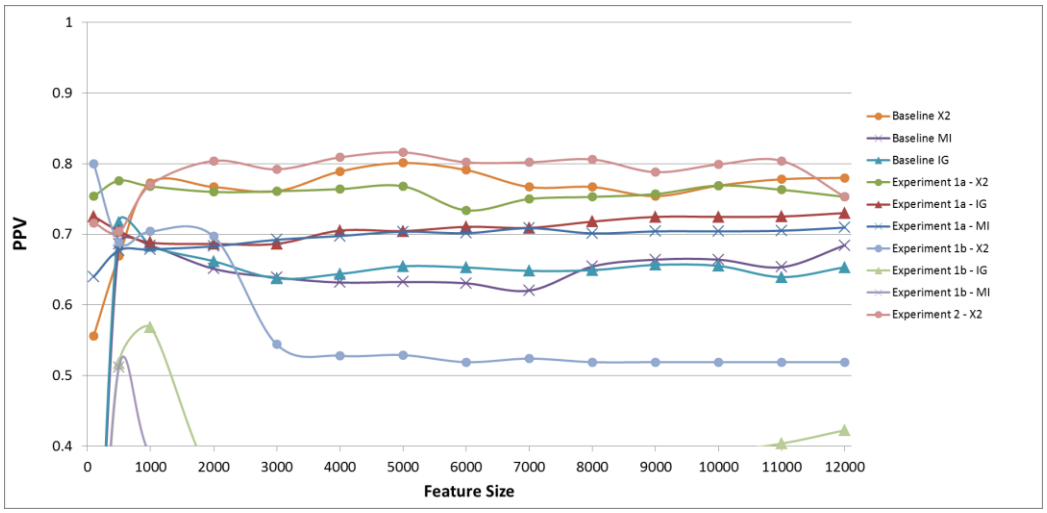

Figure 14: Feature Selection Method Comparison

4.4.4 Statistical Significance of Features

I experimented with 14 different feature sizes (100,500,1000,2000,…,12000) for each of the three feature selection methods. I compared the statistical significance of the performance (true positive, false positive, false negative as variables for the chi-square test) for each feature size (Figure 15) and for each feature selection method. There is no significance in the incremental performance gain after 4,000 features for either of the experiment sets. If the incremental performance gain (or loss) is statistically significant, the graph shows a value below the ($p < 0.05$) black line). For example the performance difference between 500 and 1,000 features is significant for the baseline classifier when using chi-square feature selection so that means that adding 500 features gives a significant increase in performance. Between 0 and 500, between 500 and 1,000 features, between 1,000 and 2,000 features and between 3,000 and 4,000 features most of the performance differences are significant. However, no more significant differences occur between feature size intervals after 4,000 features. At 2,000 features, classifiers listed in Figure 14 the chi-square feature selection methods

are very similar in performance and the performance gain in each is significant from the previous number of features (1,000). Therefore, I decided based on this information learned from the training and development set to fix the number of features for each classifier as the top 2,000 significant features. The features were ranked by the chi-square feature selection method and so I filtered the training set by the respective features that were most significant for each classifier.

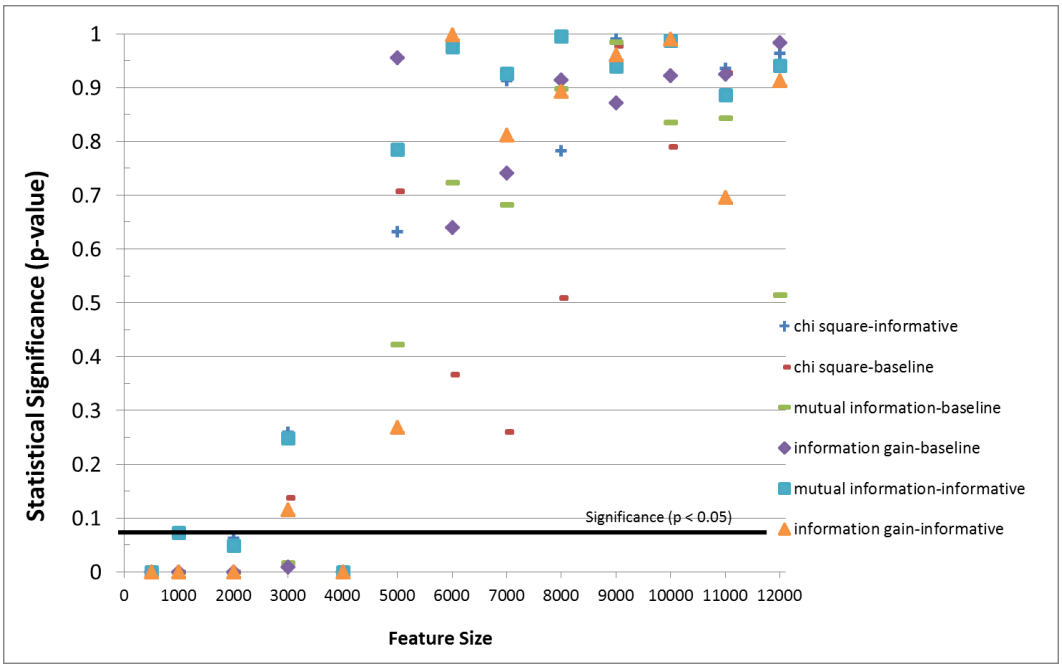

Figure 15: Feature Selection: Statistical Significance for Performance Gain

5 Evaluation Measures

As described in Section 4.3.5, the data set was split into 80% training, 10% development and 10% test sets. For testing, all systems used the common test set, 10% split from the total data set. For training and development, the rule baseline system used both the training and development sets. For the SVM baseline, Experiments 1 and 2, the development set was used to tune the respective parameters of each classifier by training on the training set and testing on the development set. After tuning, the training set was used to create a model to predict the test set.

For evaluation, the Precision (PPV) will be compared between the baseline and the two experiments, experiment 1 and experiment 2. PPV is defined by the number of true positives (TP) divided by the total number of positives as noted in equation 5.. Precision is equated with PPV here because only the positive case (sentences with ADE) is considered. PPV is an important measure of evaluation for two reasons. First, it provides a common method of comparing other ADE and ADR classification systems which favor reporting in PPV. Secondly, the stated goal of ADE classification is to reduce the cost of manual evaluation of clinical trial notes. The cost of monitoring ADEs includes a manual reviewer who receives an alert from the automated system and determines whether harm to the patient has already happened or can be prevented. An increase of false positives (FP) relative to the true positives results in decrease in PPV, and a corresponding increase in costs of the manual review. (Jha et. al, 2008)

For completeness, recall and F-score will also be reported because they are an important measure of the overall performance of each classifier. (Hripcsak et. al., 2005) Recall is defined as the total number of TP divided by the total number of TP plus the number of false negatives (FN) (equation 6). F-score is the harmonic mean between precision and recall and is defined with the equation (equation 7).

$$Precision\ (PPV) = \frac{\mathrm{TP}}{TP + FP} \tag{5}$$

$$Recall = \frac{\mathrm{TP}}{TP + FN} \tag{6}$$

$$\text{F-score} = \frac{2 * (\text{Precision} * \text{Recall})}{Precision + Recall} \tag{7}$$

For experiment 1, the model built from the informative classifier (1a) will be tested with the informative sentences from the common test set. The model built from the noninformative classifier (1b) will be tested with the noninformative sentences from the test set. The combined results will be compared against the baseline results.

For experiment 2, the labels of the test set will be transformed from two classes (positive and negative) to four class labels. The results of the experiment 2 classifier will transformed to a two class results (positive-informative and positive-noninformative class labels will be considered to be a positive label) since the baseline classifier is binary.

6 Experimental Results

For each of the classification tasks listed in the results (rule baseline, SVM baseline, experiments 1 and 2 SVM) a common test set was used for evaluation. The test set was 10% of the total data set, or 2,923 sentences. This set consists of 123 positive sentences and 2,800 negative sentences. There are 481 informative sentences (99 positive) and 2,442 noninformative sentences (24 positive).

6.1 Baseline

6.1.1 Rule Baseline

Many of the systems in the literature use a variety of rules to predict ADEs. I wrote a rule based system with 46 rules to identify terms and phrases, semantic types, CUIs related to ADEs and predict true sentences. The term and phrase matching was done with regular expressions. Seven of these rules were negative rules, designed to remove FPs from the predicted true sentences. An example of a positive rule: *If the sentence contains mention of a grade (1-5 or I-IV) and contains a mention of one or more of the terms* tired *or* fatigue*. An example of a negative rule: *If the sentence is classified as true and it contains the* CUI C0687695*, and the term* improve**, remove it from the predicted true sentences.* The CUI C0687695 indicates a severity of "grade one" (the least severe) and the sentence is not considered positive for ADE if the patient improved from that condition.

Table 6 shows the results of training and testing with all rules and with only positive rules. There is a 41% increase in false positives in the training performance and 65% in the false positives in testing when negative rules are not used.

Table 6: Rule Based Classification

	Rules	PPV[1]	Number of Sentences	TP[2]	FP[3]
Training (no negative rules)	39	0.257	26,309	392	1,134
Training	46	0.306	26,309	353	802
Testing (no negative rules)	39	0.313	2,923	45	99
Testing	46	0.381	2,923	37	60

[1] Positive Predictive Value [2] True Positives [3] False Positives

6.1.2 SVM Baseline

The SVM baseline system is trained on all 23,885 sentences in the training data set. Feature selection and tuning parameters were determined by testing on the development set. There are just two classes, positive and negative for ADE (984 and 22,041 negative sentences, respectively). After chi-square feature selection, the baseline was trained with 2000 features to match the other classifiers' feature size. The top ten ranked features for the baseline SVM classifier are listed in Table 7. The tokens have been normalized and "-" indicates the presence of a space in bigram and trigram features. Discussion about these features is in Chapter 4.

Among the top features of the four different SVM classifier, all of the 11 feature types (Table 5, Section 4.3.3) are represented. The top token features include unigram (*intermittent*, experiment 1), bigram (*exam-performance*, experiment 1), trigram (*have-grade-i*, experiment 1b) . POS tag features are also present in single (*VBZ*, baseline), double (*VP-ADJP*, baseline) and triple (*NNS-PRP-IN*, experiment 1b) POS tag sequences. UMLS semantic types (TUI) are present in the baseline and experiment 1a classifier top features, SNOMED-CT

codes are present in experiments 1a, 1b, and 2, and a CUI is present in experiment 1b. The descriptions of the UMLS entities listed in Table 7 are located in Table 8.

Table 7: Top Features (chi-square) for SVM Classification

	Baseline	Experiment 1a	Experiment 1b	Experiment 2
1	umlsTUI=T184	sign_symptom:0	umlsCode=263933003	disease_disorder:1
2	sign_symptom:1	umlsTUI=T191	grade-i	sign_symptom:1
3	VBZ	exam-performance-scores	umlsCUI=C0205615	daily-.
4	umlsTUI=T029	exam-performance	her-tongue	:-CD-(
5	JJ	scores-:	)-describes-it	call
6	RB	scores	have-grade-i	disease_disorder:2
7	VP-ADJP	performance-scores-:	NNS-PRP-IN	umlsCode=67834006
8	NP-VP-ADJP	performance-scores	have-grade	,-NN-,
9	ADJP	intermittent	i-)-describes	CD-(-NN
10	,-grade	umlsCode=64572001	to-have-grade	:-JJ-)

Table 8: UMLS Entity Feature Descriptions

UMLS Entity	Classifier	Description
TUI T184	Baseline	sign or symptom
TUI T029	Baseline	Body Location or Region
TUI T191	Experiment 1a	Neoplastic Process
SNOMED-CT 64572001	Experiment 1a	Disease
SNOMED-CT 263933003	Experiment 1b	Well differentiated
CUI C0205615	Experiment 1b	Well differentiated
SNOMED-CT 67834006	Experiment 2	Structure of deciduous mandibular right central incisor tooth

The classifier was trained on the training set and tested on the common testing set described in Chapter 6. The results are given in Table 9. There were 57 TP and 66 FP sentences, resulting in a PPV of 46.3%

Table 9: Results of Baseline SVM

	Precision (PPV)	Recall	F-score	TP	FP
Baseline	0.463	0.750	0.573	57	66

6.2 Experimental SVM

The experimental SVM classifiers use informativeness as described in Section 4.3.5. These are to demonstrate the validity of the hypothesis when compared with the results in Section 6.1.

6.2.1 Experiment 1

In experiment 1 there are two classifiers, one for informative sentences and a second for noninformative sentences. I combined the results of experiment 1, the informative classifier and the noninformative classifier (1a, 1b, respectively) in order to compare against the baseline. Since each of the classifiers only handles a portion of the test data (1a tests the informative sentences and 1b test the noninformative), combining the results is necessary. The classifier in experiment 2 handles the whole data set, so the results are sufficient for this experiment to compare against the baseline.

The training data in experiment 1a consists of 785 positive and 2,951 negative sentences. The training data in experiment 1b consists of 199 positive and 19,450 negative sentences. The testing results of the experiment 1a and 1b classifiers on the common test set are listed separately and combined in Table 10.

Table 10: Results of Baseline and Experiments SVM Classifier

	Precision (PPV)	Recall	F-score	TP	FP
Experiment 1a	0.918	0.795	0.852	627	56
Experiment 1b	0.921	0.704	0.798	140	12
SUM EXP1[1]	**0.919**	**0.776**	**0.841**	**767**	**68**

[1]Experiment 1

6.2.2 Experiment 2

The SVM classifier in experiment 2 used informative sentences in order to modify the class label, or create two subclasses for each positive and negative label. In the training data the positive sentences are divided into informative (785, "positive-informative") and noninformative (199, "positive-noninformative"). The negative sentences consist of informative (2,951, "negative-informative") and noninformative (19,450, "negative-noninformative"). The results for the testing on the common test set are listed in Table 11. Because there are two positive labels, the results of each positive label are listed separately and then collocated as a single positive label. As shown in Table 11, the results of the positive-informative sentences (564 TP, 50 FP), were added to the results of the positive-noninformative sentences (64 TP, 50 FP) to produce a combined result of 91.3% PPV.

Table 11: Results of Experiment 2 SVM Classifier

	Precision (PPV)	Recall	F-scrore	TP	FP
Experiment 2 (informative)	0.919	0.718	0.806	564	50
Experiment 2 (noninformative)	0.865	0.322	0.469	64	10
SUM EXP2[1]	**0.913**	**0.638**	**0.751**	**628**	**60**

[1]Experiment 2

6.3 Results Comparison

The four machine learning classifiers (baseline, experiment 1a, experiment 1b, and experiment 2) and the rule based baseline are presented in Table 12 for comparison. The machine learning baseline outperformed the rule based classification in PPV (46.3% vs. 38.1%) but the results were not significant (p-value 0.222). Both experiments which first classify the informativeness of a sentence had significantly better results to the machine learning baseline (Table 13). Experiment 1 had the best performance when the two component classifiers (informative and noninformative) were combined. The results for experiment 1 PPV were slightly less (70.4% vs. 80.3%, PPV) and this difference is statistically significant (p-value 0.0046). However, the classifier in experiment 1 performed better in terms of F-score (69.1% vs. 61.3%, p-value 2.1 x 10^{-03}).

Table 12: ADE Classification Results Comparison

	Precision (PPV)	Recall	F-score	TP	FP
Baseline Rule	0.381	0.301	0.336	37	60
Baseline SVM	0.463	0.750	0.573	57	66
Experiment 1	**0.704**	0.679	0.691	76	32
Experiment 2	0.803	0.496	0.613	61	15

Table 13: Significance of PPV Results (p-value)

	Baseline SVM	Baseline Rule	Experiment 1	Experiment 2
Baseline SVM				
Baseline Rule	**0.222**			
Experiment 1	1.48 x10^{-12}	1.10 x 10^{-10}		
Experiment 2	5.37 x 10^{-08}	3.22 x10^{-08}	0.0046	

6.4 Effect of Feature Selection

After chi-square feature selection, described in section 4.4, I selected the top 2,000 features by their chi-square value. Feature selection is important not only for computational tractability (there are 255,743 features in the original data) but also for strength of signal for the machine learning algorithm. Only 162,420 of the features occur more than once and only 13,244 occur more than 50 times. In order to see the effect of feature selection on the corpus, I selected the feature which occurred more than 50 times and trained and tested SVM classifiers (baseline, experiment 1 and experiment 2) as described in sections 6.1.2-6.2.

All three classifiers showed lower results (Table 14) with a naïve feature selection. The results for experiment 1 were significantly lower (66.1% vs. 70.4 % PPV, p-value 7.14 x 10^{-12}). For experiment 2 the results were similarly lower (57.9% vs. 80.3 % PPV, p-value 3.39 x 10^{-04}). For naïve feature selection method, there was no statistical difference between the PPV for experiment 1 and 2 (66.1% vs. 57.9%, p-value 0.22), but there was for F-score (64.7% vs. 50.5%, p-value 2.20 x 10^{-02}). Without a sophisticated method of feature selection, there is no statistical difference between the baseline and experiment 2 (51.9% vs. 57.9% PPV, p-value 0.388).

Table 14: ADE Classification with Naïve Feature Selection

	Precision (PPV)	Recall	F-score	TP	FP
Baseline SVM	0.519	0.455	0.485	56	52
Experiment 1	**0.661**	0.633	0.647	76	39
Experiment 2	0.579	0.447	0.505	55	40

7 Conclusions

In this thesis I have proposed the hypothesis that, by using the informativeness of a sentence, the performance of an ADE classification will be improved. I developed a pipeline which identifies informativeness of a sentence, defined by containing one or more medical conditions, and creates a feature vector for SVM classification. The baseline SVM classification ignores the informativeness of a sentence and only classifies positive or negative. Experiment 1 uses two separate classifiers, one for informative sentences and one for positive. Experiment 2 uses the informativeness of a sentence to subdivide the positive and negative classes, resulting in a multiclass (four classes) SVM.

As shown in Figure 8, the distribution of the sentences into informative and noninformative heavily favors positive sentences as informative. Over 21% (985/4683) of positive sentences are informative, while only 1% (251/ 24,549) of negative sentences were informative. However, because 20% (251/1,236) of the positive sentences were still classified as noninformative, simply classifying only informative sentences left out a significant portion of sentences with ADEs. As a result I concluded that it was necessary to classify the noninformative sentences separately with a unique classifier. This was done in experiment 1b. Comparing the result of experiment 1 and 2 confirms the conclusions of Hsu et. al. (2002), who discovered that binary ("one vs. one") SVM classifiers perform better than multiclass SVM (which employ a "one vs. all" method) when considering F-score. The experiment 2 classifier performed better than experiment 1 for PPV (80.3% vs. 70.4%, respectively) but experiment 1 had a better F-score than experiment 2 for F-score (69.1% vs. 61.3%, respectively). Both of the experiments that treat the informativeness of the sentence perform better than the baseline, as shown in Table 12. The null hypothesis (from Chapter 1) is: Classifying sentences into informative and non-informative in the first step of

ADE detection will not improve the performance of the ADE classifier. The null hypothesis is rejected.

7.1 Feature Selection

Chi-square feature selection allowed me the ability to select the top *n* features that were significant, relative to the positive or negative class. When I performed chi-square feature selection on the experiment 2 training data set, it was among the four classes. It was the best performing method among the three feature selection methods of chi-square, mutual information and information gain. In examination of the top features selected as significant for each classifier set, I noticed an interesting fact. Of the top 100 features for the classifiers in experiment 1 (informative and noninformative) there is only one common feature. So each of the two classifiers in experiment 1 have 99 unique significant features for classification. Of the top 2,000, there are only 103 (5.15%) common features between the two classifiers. Of the top 2,000 features among the baseline and experiment 2 classifiers, there are only 56 (2.8%) common features. I conclude from this information that the feature selection method performed quite well in distinguishing the features that were characteristic of each set (informative, noninformative, etc.) because of the limited overlap.

It is also important to note that the informative features (e.g. disease_disorder:1, sign_symptom:2) are not present in the top features for the experiment 1b classifier. It is important because every vector the training set for the classifier is non-informative and has among the features both disease_disorder:0 and sign_symptom:0. Informative features are significantly present in the top features of the classifiers which contain informative sentences in the training examples (baseline, experiment 1a, and experiment 2, Table 7). In the case of experiment 1a the feature sign_symptom:0 is an important feature for the negative class.

Using a sophisticated method of feature selection provided a positive impact in the results of each classifier. Furthermore, with only a naïve method of feature selection such as using features which occur more than *x* times, there is no significant improvement in PPV between the baseline SVM and experiment 2. There is still a statistically significant improvement between the baseline SVM and experiment 1, however demonstrating that both informativeness classification and feature selection are required as prior steps to improving the classification of ADEs.

7.2 Correlation of Informativeness

A sentence is defined as informative if it contains one or more disease/disorder or sign/symptom entities. The prediction of these entities is made by the machine learning classifier. Since the hypothesis is that by using the informativeness of a sentence, I can increase the performance of an ADE classifier, I decided to look at the correlation of these predictions to both the positive class and the informative label of a sentence. Each of these predictions by the classifier was used as a feature. There are 45 medical condition features including sign_symptom:0, disease_disorder:2, disease_disorder:9, sign_symptom:5, where the number following the colon indicates the number of predicted entities in that sentence. The correlation method was Pearson's *r* correlation. *R* values are given in the following discussion.

First, because of the informativeness rule definition, it should follow that most of medical condition features are highly correlated to the informative label. The exceptions are possibly sign_symptom:0 and disease_disorder:0, because the presence of both of these in a vector equates to a noninformative sentence. All of the medical condition features were perfectly correlated ($r = 1.0$) with the expected exceptions (sign_symptom:0, r=-0.0057; disease_disorder:0 = -0.00053)

Secondly, the correlation of the medical condition features to the positive class shows a positive association with the method of selecting informative sentences for classification. Of the 46 medical condition features, only 16 were positively correlated. (> 0) and of these 7 were strongly correlated (> 0.5). Thirty of the features are negatively correlated and 17 of these strongly so (< -0.5). However, for the strongly negative correlated features, a pattern can be linked to an error in the sentence detection algorithm. The average count of the medical conditions is 27 for those features that have a Pearson's *r* score of -1.0 (e.g., disease_disorder:43; sign_symptom:27). In examining these sentences, many of them seemed to be blocks of bullets that were not detected as separate sentences (e.g. one sentence per bullet), but detected en masse as a single sentence. Formatting issues in the CNs caused bulleted lists and other sentences which lacked traditional ending punctuation to be collapsed and not split into separate lines.

The impact of improving the sentence detection module is undetermined, but perhaps it would serve to create more negative sentences. Due to the large number of predicted medical conditions in these bulleted blocks, it is likely that the number of informative sentences would also increase.

The performance of the medical condition classifier on the CNs used in my thesis (Table 4) is very good at 85.8% (overall F-score) and is only slightly worse than the original result on a different corpus (FDA drug labels, 89% F-score). A high performance is necessary because of the downstream impact on identifying sentences as informative. More CNs to the training set will be added in future work in order to improve the performance.

7.3 Limitations

Not all toxicities are highlighted in the notes by the CRCs, usually only grade 2/3 or higher, sometimes baseline toxicities are not highlighted because they are

assumed. The study protocol details the circumstances of toxicity reporting and the context of the course, progression of disease and baseline toxicities are considered when the CRCs report ADEs. In annotation no effort was made to supplement the decision of the CRCs, so it is possible a number of sentences were left negative, when a more complete reporting of toxicities would have identified an ADE. The result of the rule based system identified approximately 802 false positives (out of all 29,232 instances, both informative and noninformative) according to the gold standard. A random sampling of 100 of the false positives were manually reviewed by both annotators and after adjudication 29 were determined to be adverse events, when considered in isolation. Extrapolating these "true" false positives would have raised the training PPV to 48.6% and the testing PPV to 54.6%. Additionally, a CRC was given the same data and determined 30 of the 100 sentences to be true ADEs. The inter-annotator agreement between the consensus (adjudicated between the two annotators) and the CRC was 85% (F-score). The expected increase in PPV for the rule based system (with 30% less FP and 30% more TP) is shown in Table 15. This demonstrates validity in the method of converting the toxicity grading of trained and experienced CRCs to electronic data by annotators who have not had the same training. The fact that both of the annotators and the CRC agreed on the high number of "true" false positives was also seen in other studies (Tinoco et al., 2011; Kilbridge et al., 2009; Raschke et al., 1998) where the automated system of ADE detection captured ADEs that were missed by manual or voluntary reporting.

Table 15: Adjusted Rule Based ADE Classification for Erroneous False Positives

Rule	Test PPV	Training PPV
actual	0.381	0.306
adjusted	0.567	0.620

Only ten patients were used for the thesis work. Adding more patients for both ADE gold standard and medical condition gold standard training sets will improve the classification results and the determination of informativeness. The IAA for the annotation is not ideal. The ten patients represented were among the first patients annotated, and so many were used for training the annotators in the task of converting the annotation. Two additional factors hamper the IAA. First, some protocols for certain clinical trials do not require the reporting of grade 1 or grade 2 ADEs and so there was disagreement between the annotators as to what should be annotated as an ADE. Secondly, many sentences in the printed notes were highlighted but did not contain ADEs because protocols require highlighting of disease progression and other treatment which was confusing to the annotators.

7.4 Future Work

Using the informativeness of sentences is a useful method to improve classification of the positive cases. Classifying ADEs as binary is an initial step toward a complete ADE classification system. Future work on a complete system will include classifying all the categories of ADE (e.g., Pain, Gastrointestinal, Blood/Bone Marrow), identifying severity and attribution of the ADE to the medication.

REFERENCES

Albright, D., Lanfranchi, A., Fredriksen, A., Styler, W. F., Warner, C., Hwang, J. D., Martin, J. (2013). Towards comprehensive syntactic and semantic annotations of the clinical narrative. *Journal of the American Medical Informatics Association*.

Aramaki, E., Y. Miura, et al. (2010). "Extraction of adverse drug effects from clinical records." *Studies In Health Technology And Informatics* 160(Pt 1): 739-743.

Bates, D. W., Evans, R. S., Murff, H., Stetson, P. D., Pizziferri, L., & Hripcsak, G. (2003). Detecting adverse events using information technology. *Journal of the American Medical Informatics Association, 10*(2), 115-128.

Belknap, S., Georgopoulos, C., West, D., Yarnold, P., & Kelly, W. (2010). Quality of methods for assessing and reporting serious adverse events in clinical trials of cancer drugs. *Clinical Pharmacology & Therapeutics, 88*(2), 231-236.

Bhavani, S., Nagargadde, A., Thawani, A., Sridhar, V., & Chandra, N. (2006). Substructure-based support vector machine classifiers for prediction of adverse effects in diverse classes of drugs. *Journal of chemical information and modeling, 46*(6), 2478-2486.

Cairns, B. L., Nielsen, R. D., Masanz, J. J., Martin, J. H., Palmer, M. S., Ward, W. H., & Savova, G. K. (2011). *The MiPACQ clinical question answering system.* Paper presented at the AMIA Annual Symposium Proceedings.

Cami, A., Arnold, A., Manzi, S., & Reis, B. (2011). Predicting adverse drug events using pharmacological network models. *Sci Transl Med, 3*(114), 114ra127.

Cao, H., Stetson, P., & Hripcsak, G. (2003). *Assessing explicit error reporting in the narrative electronic medical record using keyword searching.* Paper presented at the AMIA Annual Symposium Proceedings.

Classen, D. C., Pestotnik, S. L., Evans, R. S., & Burke, J. P. (1991). Computerized surveillance of adverse drug events in hospital patients. *Journal of the American Medical Association, 266*(20), 2847-2851.

Cohen, R., Elhadad, M., & Elhadad, N. (2013). Redundancy in electronic health record corpora: analysis, impact on text mining performance and mitigation strategies. *BMC bioinformatics, 14*(1), 10.

DiMasi, J. A., Hansen, R. W., & Grabowski, H. G. (2003). The price of innovation: new estimates of drug development costs. *Journal of health economics, 22*(2), 151-186.

Ferranti, J., Horvath, M. M., Cozart, H., Whitehurst, J., & Eckstrand, J. (2008). Reevaluating the safety profile of pediatrics: a comparison of computerized adverse drug event surveillance and voluntary reporting in the pediatric environment. *Pediatrics, 121*(5), e1201-e1207.

Forster, A. J., Jennings, A., Chow, C., Leeder, C., & van Walraven, C. (2012). A systematic review to evaluate the accuracy of electronic adverse drug event detection. *Journal of the American Medical Informatics Association, 19*(1), 31-38.

Friedman, C., Shagina, L., Socratous, S. A., & Zeng, X. (1996). *A WEB-based version of MedLEE: A medical language extraction and encoding system.* Paper presented at the Proceedings of the AMIA Annual Fall Symposium.

Gurulingappa, H., A. M. Rajput, et al. (2012). "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports." *Journal of Biomedical Informatics.*

Gurwitz, J. H., Field, T. S., Harrold, L. R., Rothschild, J., Debellis, K., Seger, A. C., Kelleher, M. (2003). Incidence and preventability of adverse drug events among older persons in the ambulatory setting. *JAMA: the journal of the American Medical Association, 289*(9), 1107-1116.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter, 11*(1), 10-18.

He, Y., Chong, F. H. T., Lim, J., Lee, R. J. T., & Yap, C. W. (2013). Determination of the Potential of Drug Candidates to Cause Severe Skin Disorders Using Computational Modeling. *Molecular Informatics*.

Honigman, B., Lee, J., Rothschild, J., Light, P., Pulling, R. M., Yu, T., & Bates, D. W. (2001a). Using computerized data to identify adverse drug events in outpatients. *Journal of the American Medical Informatics Association, 8*(3), 254-266.

Honigman, B., Light, P., Pulling, R. M., & Bates, D. W. (2001b). A computerized method for identifying incidents associated with adverse drug events in outpatients. *International Journal of Medical Informatics, 61*(1), 21-32.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on, 13*(2), 415-425.

Huang, C., Noirot, L. A., Reichley, R. M., Bouselli, D. A., Dunagan, C., & Bailey, T. C. (2005). *Automatic Detection of Spironolactone–Related Adverse Drug Events.* Paper presented at the AMIA Annual Symposium Proceedings.

Hwang, S.-H., Lee, S., Koo, H.-K., & Kim, Y. (2008). Evaluation of a computer-based adverse-drug-event monitor. *Am J Health Syst Pharm, 65*(23), 2265-2272.

Imai, K., & Takaoka, A. (2006). Comparing antibody and small-molecule therapies for cancer. *Nature Reviews Cancer, 6*(9), 714-727.

Jha, A. K., Kuperman, G. J., Teich, J. M., Leape, L., Shea, B., Rittenberg, E., Bates, D. W. (1998). Identifying adverse drug events development of a computer-based monitor and comparison with chart review and stimulated voluntary report. *Journal of the American Medical Informatics Association, 5*(3), 305-314.

Jha, A. K., Laguette, J., Seger, A., & Bates, D. W. (2008). Can surveillance systems identify and avert adverse drug events? A prospective evaluation of a commercial application. *Journal of the American Medical Informatics Association, 15*(5), 647-653.

Kilbridge, P. M., Noirot, L. A., Reichley, R. M., Berchelmann, K. M., Schneider, C., Heard, K. M., Bailey, T. C. (2009). Computerized surveillance for adverse drug events in a pediatric hospital. *Journal of the American Medical Informatics Association, 16*(5), 607-612.

Ko, Y., Park, J., & Seo, J. (2004). Improving text categorization using the importance of sentences. *Information Processing & Management, 40*(1), 65-79.

Li, Q., Deleger, L., Lingren, T., Zhai, H., Kaiser, M., Stoutenborough, L., Solti, I. (2013). Mining FDA drug labels for medical conditions. *BMC Medical Informatics and Decision Making, 13*(1), 53.

Liu, M., Wu, Y., Chen, Y., Sun, J., Zhao, Z., Chen, X.-w., Xu, H. (2012). Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs. *Journal of the American Medical Informatics Association, 19*(e1), e28-e35.

Luo, Z., M. Yetisgen-Yildiz, et al. (2011). "Dynamic categorization of clinical research eligibility criteria by hierarchical clustering." *Journal of Biomedical Informatics* 44(6): 927-935.

McKnight, L., & Srinivasan, P. (2003). *Categorization of sentence types in medical abstracts.* Paper presented at the AMIA Annual Symposium Proceedings.

Melton, G. B., & Hripcsak, G. (2005). Automated detection of adverse events using natural language processing of discharge summaries. *Journal of the American Medical Informatics Association, 12*(4), 448-457.

Mitchell, A., Divoli, A., Kim, J.-H., Hilario, M., Selimas, I., & Attwood, T. (2005). METIS: multiple extraction techniques for informative sentences. *Bioinformatics, 21*(22), 4196-4197.

Morimoto, T., Gandhi, T., Seger, A., Hsieh, T., & Bates, D. (2004). Adverse drug events and medication errors: detection and classification methods. *Quality and Safety in Health Care, 13*(4), 306-314.

Murff, H. J., Forster, A. J., Peterson, J. F., Fiskio, J. M., Heiman, H. L., & Bates, D. W. (2003). Electronically screening discharge summaries for adverse medical events. *Journal of the American Medical Informatics Association, 10*(4), 339-350.

Naranjo, C. A., Busto, U., Sellers, E. M., Sandor, P., Ruiz, I., Roberts, E., Greenblatt, D. (1981). A method for estimating the probability of adverse drug reactions. *Clinical Pharmacology & Therapeutics, 30*(2), 239-245.

Naughton, M., Stokes, N., & Carthy, J. (2008). *Investigating statistical techniques for sentence-level event classification.* Paper presented at the Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1.

Nebeker, J. R., Barach, P., & Samore, M. H. (2004). Clarifying adverse drug events: a clinician's guide to terminology, documentation, and reporting. *Annals of Internal Medicine, 140*(10), 795-801.

Niland, J. C., Stiller, T., Neat, J., Londrc, A., Johnson, D., & Pannoni, S. (2012). Improving patient safety via automated laboratory-based adverse event grading. *Journal of the American Medical Informatics Association, 19*(1), 111-115.

Ogren, P. (2006). *Knowtator: a protégé plug-in for annotated corpus construction*. Paper presented at the Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume: demonstrations, New York, New York.

Raschke, R. A., Gollihare, B., Wunderlich, T. A., Guidry, J. R., Leibowitz, A. I., Peirce, J. C., Susong, C. (1998). A computer alert system to prevent injury from adverse drug events. *The Journal of the American Medical Association, 280*(15), 1317-1320.

Resar, R., Rozich, J., & Classen, D. (2003). Methodology and rationale for the measurement of harm with trigger tools. *Quality and Safety in Health Care, 12*(suppl 2), ii39-ii45.

Savova, G. K., J. J. Masanz, et al. (2010). "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications." *Journal of the American Medical Informatics Association* **17**(5): 507-513.

Scott, S., & Matwin, S. (1999). *Feature engineering for text classification.* Proceedings of ICML-99, 16th International Conference on Machine Learning.

Szekendi, M., Sullivan, C., Bobb, A., Feinglass, J., Rooney, D., Barnard, C., & Noskin, G. (2006). Active surveillance using electronic triggers to detect adverse events in hospitalized patients *Quality and Safety in Health Care, 15*(3), 184-190.

Tatonetti, N. P., Fernald, G. H., & Altman, R. B. (2012). A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports. *Journal of the American Medical Informatics Association, 19*(1), 79-85.

Tinoco, A., Evans, R. S., Staes, C. J., Lloyd, J. F., Rothschild, J. M., & Haug, P. J. (2011). Comparison of computerized surveillance and manual chart review for adverse events. *Journal of the American Medical Informatics Association, 18*(4), 491-497.

Trotti, A., A. D. Colevas, et al. (2003). CTCAE v3. 0: development of a comprehensive grading system for the adverse effects of cancer treatment. Seminars in radiation oncology, Elsevier.

Xu, R., Supekar, K., Huang, Y., Das, A., & Garber, A. (2006). *Combining text classification and hidden Markov modeling techniques for structuring randomized clinical trial abstracts.* Paper presented at the AMIA Annual Symposium Proceedings.

Yang, Y., & Pedersen, J. O. (1997). *A comparative study on feature selection in text categorization. Proceedings of ICML-99, 16th International Conference on Machine Learning.*

Zhang, W., Yoshida, T., & Tang, X. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems, 21*(8), 879-886.

Zheng, Z., Wu, X., & Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletter, 6*(1), 80-89.

APPENDIX A:
Adverse Drug Event Annotation Guidelines

1. Purpose

   - Take the highlighted text from the dated clinical note from the patient binder, match it to the corresponding Toxicity report data with the same date, and convert it to knowtator

2. Set up

   - Open two NX windows
   - One server for Protégé/the other for Toxicity Report
   - /data/knowtator/annotators/ADR – root directory of projects and text folders
   - Control scroll up to increase the font/zoom in

3. How to start

   - Find date in toxicity report and corresponding date in binder
   - Look for Dr name of clinical note to match knowtator text
   - Do not look at labs or patient self-reports (i.e. hand written forms)
   - Do not look at radiographic reports even though highlighted

4. Look at date in toxicity report

   - Look at all area contents marked and find in the progress notes the same phrase.
   - Decide how to annotate it
   - Then give it AE Grade 0-5 (already marked in tox report) :
     0 is resolved (look for date started and date resolved), unless date of

occurrence in toxicity report has grade greater than 0. Then use that current grade.
On toxicity report, it's the cell that corresponds with the date and AE subcategory

- AE subcategory- will be type of content more specific; listed under AE category on tox report; ex: pain can have multiple pull down
- The Attribution Severity-this is the adverse drug reaction possibility; this is already coded in the toxicity report in DARK BLUE column (1-5 scale, leave blank if blank)
  if more than one attribution column, use first one which is the study drug (Do not use "Research" or "Disease" attribution)
- Example: His throat was mildly sore a few days ago and he had a mild headache last night, but both have resolved.
  - Highlight everything but "his" if sore throat and headache have the same grade AND severity attribution – if different, split up
  - Adverse Event (Content) is Pain
  - AE subcategory would be throat pain and headache
  - Then give the number that corresponds to grade (on the toxicity report)
  - Then the attribution severity code would plug in if there is one (blue column on toxicity report)

5. Examples how to annotate
   - Lungs: substernal pain in deep inspiration = highlight this whole phrase
6. Some odd examples
   - Slightly tired-Constitutional Symptom - AE/Content subcategory is fatigue
   - Edema-lymphatic - AE/Content Subcategory is limb edema
   - Allergy immunology - AE/content subcategory is Allergic Rhinitis

- Feeling down-neurology – AE/Content subcategory is mood alteration
- Performance score - AE/content (Karnofsky) and input the # also that is the score
- Pain is better - this is a negative if not in the toxicity report on that date
- Not had any discernible side effects - negative
- Eating well – Negative unless there is a grade under anorexia on toxicity report
- Pain is better, not needing any pain meds - negative

7. Items to ignore
   - Drug dosages "patient received 500mg lasik for two weeks"
   - Treatment, except "given platelets", this is Blood/Bone Marrow – Platelets
   - Signs/symptoms of disease (eg: tumor size; see exceptions below)
8. Other cases
   - If Toxicity report does not contain the grade or data from the highlighting, AND the statement is a negative case like "no discernible side effects" create an "Negative" annotation
   - If the Toxicity report does not contain create an "Adverse Event" annotation (i.e. without the subcategory).

Things to include in annotation (only if highlighted) *Do not add things that are not highlighted*

- All verbs (is, had, has…)
- Continues, reports, complains, noted…
- Include header only if it is highlighted.
- Include dates/time frames (overnight, yesterday) if they relate to a particular adverse event/problem.
- Include the medication if it is mentioned in the toxicity report and is highlighted with the ADR

Things not to include:

- Do not include anything that refers to their disease (plexiform, café au lait spots, neurofibroma, new growth or spots)
- Measurements of wounds or tumor sizes
- Do not include treatments unless the treatment causes, has resulted in, or explains the ADR further.
- Do not take "history of…" unless referring to a recent history (ex: two day history of abdominal pain)

Toxicity Report

- If there is no grade for the ADR on the date that you are examining,
  - Is the date inside a highlighted span which indicates a continuing toxicity?
    Then use most recent grade
  - If not, then refer back at most two days to that grade
- When the grade is 1(0), use the highest (e.g. 1).

When the type of pain is not specified in the note (ex: increased pain today) and the context around the pain suggests different locations or types of pain that match the toxicity report for that date, annotate with no grade or attribution or subcategory.