Coreference Resolution on Entities and Events for Hospital Discharge Summaries

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
NOV 13 2008
LIBRARIES

by

Tian Ye He

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2007

©2007 Massachusetts Institute of Technology 2007. All rights reserved.

Author_____
Department of Electrical Engineering and Computer Science
August 21, 2007

Certified by_____
Ozlem Uzuner
Assistant Professor, SUNY
Thesis Supervisor

Certified by_____
Peter Szolovits
Professor
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses

ARCHIVES

Coreference Resolution on Entities and Events for Hospital Discharge Summaries

by

Tian He

Submitted to the
Department of Electrical Engineering and Computer Science

August 21, 2007

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

The wealth of medical information contained in electronic medical records (EMRs) and Natural Language Processing (NLP) technologies that can automatically extract information from them have opened the doors to automatic patient-care quality monitoring and medical-assist question answering systems. This thesis studies coreference resolution, an information extraction (IE) subtask that links together specific mentions to each entity. Coreference resolution enables us to find changes in the state of entities and makes it possible to answer questions regarding the information thus obtained.

We perform coreference resolution on a specific type of EMR, the hospital discharge summary. We treat coreference resolution as a binary classification problem. Our approach yields insights into the critical features for coreference resolution for entities that fall into five medical semantic categories that commonly appear in discharge summaries.

Thesis Supervisor: Ozlem Uzuner
Title: Assistant Professor, SUNY
Thesis Supervisor: Peter Szolovits
Title: Professor

Acknowledgements

It has been a long journey since I first started working at the Clinical Decision Making Group two years ago. Countless hours spent working, eating, and learning at the lab, and every step of the way Professor Ozlem Uzuner was there—to guide me when I was lost, and to encourage me when I was down. Ozlem, thank you for your great patience and those long bus rides from Albany to Cambridge. You have been nothing short of an inspiration. These words do not do justice in describing the integral role you played in this thesis and my past two years at MIT.

To Professor Szolovits, thank you for pearls of wisdom, your constant support, and your willingness to help me whenever the need arises. Listening to your thoughts on research and your life stories will be something I sorely miss.

I would also like to thank Sharon Zhang and Silvia Batista for their hard work assisting me with this research. Your positive attitudes and your company made my final months in lab enjoyable and fun.

And finally to my family. This past year has been the most challenging year of my academic and personal life. Without my parents, godparents, and my best friend, I would not be where I am. Mom and dad, you are the wind in my sail and the beat to my drum. Wherever, I am, whatever, I do, I know you will support me 100%. That means the world to me. And to gan-ma, you are the most loving and righteous person I know. What I learn from you about life, I can not learn from the classroom or anyone else. Lu, funny how a 12 hour plane ride turned into an 11 year friendship huh? Thanks for 11 years of laughs, summer trips, and birthday celebrations. Even though our birthday is a day apart, you seem the infinitely wiser to me. You have always given me a reason to believe I can do more and be more.

Twenty years from now you will be more disappointed by the things that you didn't do than by the ones you did do. So throw off the bowlines. Sail away from the safe harbor. Catch the trade winds in your sails. Explore. Dream. Discover. – Mark Twain

To all those mentioned and the many others who have touched my life, you have all given me a reason to explore, dream, and discover.

Table of Contents

1 INTRODUCTION 9
1.1 Motivation 9
1.2 Coreference Resolution 9
1.3 Thesis Structure 10
2 BACKGROUND 11
2.1 Information Extraction Overview 11
2.1.1 Preprocessing Module 11
2.1.2 Main Processing Module (MPM) 11
2.1.3 A Sample IE System: CaRE 12
2.2 Coreference Resolution 13
2.2.1 Terms and Definitions 14
2.2.2 Convention 14
2.2.3 Past Research 14
3 RESOURCES 25
3.1 UMLS 25
3.2 C4.5 Decision Tree 26
3.3 Kappa Statistic 27
3.4 Evaluation Methods 28
3.4.1 Precision, Recall, and F-measure 28
4 COREFERENCE RESOLUTION 32
4.1 Task Definition 32
4.2 Data 33
4.3 Annotation 34
4.3.1 Coreference Resolution 34
4.3.2 Time Stamp 39
4.4 Initial Evaluation 40
Preliminary Evaluation by Markable String-Match Type 40
4.5 Semantic Coreference Resolver 41
4.5.1 Instance Creation: 41
4.5.2 Feature Set: 42
Orthographic 45
Semantic 46
Lexical Features 47
Syntactic and Morphological Features 48
Temporal Features 49
Miscellaneous 49
4.5.3 Machine Learner 50
4.5.4 Clustering Algorithm: 50
4.6 Evaluation 51
4.6.1 Overall System Performance 51
4.6.2 Effect of the Multi-Perspectives Approach 55
4.6.3 Feature Contributions 56
Evaluation Method 57
Direct Effect Analysis 58
Complementary Effect Analysis 60
4.7 Discussion 61
4.7.1 CONS 61
4.7.1.1 False Alarm 62

4.7.1.2 False Negative 62
4.7.2 DIS 63
4.7.2.1 False Alarm 63
4.7.2.2 False Negative 65
4.7.3 MED 66
4.7.3.1 False Alarm 66
4.7.3.2 False Negative 67
4.7.4 SYMP 68
4.7.4.1 False Alarm 68
4.7.4.2 False Negative 68
4.7.5 TEST 69
4.7.5.1 False Alarm 70
4.7.5.2 False Negative 71
5 FUTURE WORKS 72
6 CONCLUSION 75
7 BIBLIOGRAPHY 76
APPENDIX 81
A. Decision Trees for each Semantic Category 81
CONS 81
DIS 81
MED 83
SYMP 85
TEST 86

List of Figures

Figure 1: KIM semantic IE flow diagram [40] 12
Figure 2: Examples of coreference in sentences 13
Figure 3: Algorithm for Metathesaurus mapping [2] 26
Figure 4: Illustrations of B-CUBED evaluation 30
Figure 5: General coreference resolution rules 32
Figure 6: Sample coreference annotations 34
Figure 7: Screenshot of coreference annotation GUI 35

List of Equations

Equation 1: Minimum edit distance 21
Equation 2: TFIDF-weighted cosine similarity score 22
Equation 3: The information gain ratio [62] 26
Equation 4: Kappa statistic 27
Equation 5: Traditional precision, recall, and F-measure evaluation 28
Equation 6: Model theoretic approach to calculating recall and precision 29
Equation 7: B-CUBED approach to calculating precision and recall 30
Equation 8: Wilcoxon signed-rank statistic 31

List of Tables

Table 1: Common NLP tools 11
Table 2: CaRE semantic category 13
Table 3: Features for the McCarthy and Lehnert coreference resolution system 15
Table 4: RESOLVE evaluation 15
Table 5: Performance of DSO on MUC-6 data 17
Table 6: Methods at each step that were combined by Ng's system [38] 19
Table 7: Best performing combined coreference system for each test set 19
Table 8: Features used to determine coreferent anaphor/antecedent pairs 20
Table 9: Relational features used by LMR 22
Table 10: Pair-wise classification performance (F-measure) for three name-matching algorithms [28] 23
Table 11: Annotator agreement confusion matrix 27
Table 12: Sections of a hospital discharge summary 34
Table 13: Coreference Kappa statistics for each semantic category 36
Table 14: Annotation formats for date information 40
Table 15: Distribution of coreferent markable over different string-match types 40
Table 16: Pair-wise F-measure evaluation of classifying all markables of different string-match types as coreferent 41
Table 17: Total number of markable pairs 42
Table 18: Features set 45
Table 19: UMLS semantic type mappings to CaRE semantic category [46] 46
Table 20: all-feature and baseline system MUC and B-CUBED F-measures 53
Table 21: Difference between all-feature and baseline system 53
Table 22: F-measure evaluation of all-feature vs. baseline system by string-match type 54
Table 23: Single perspective system MUC and B-CUBED evaluations 55
Table 24: Difference between all-feature and single-perspective system 56
Table 25: Difference between all-feature system and antecedent-perspective 56
Table 26: Single feature system pair-wise F-measure evaluation 58
Table 27: Leave-one-feature-out system performance 60

1 INTRODUCTION

1.1 Motivation

Healthcare providers around the world are starting to introduce electronic medical record (EMR) technology into their operations to improve medical records' accessibility and quality. In the United Kingdom, a national initiative is under way to deploy a nation-wide system for centralized storage of patient EMRs [25]. In the United States, congress has considered legislations to provide EMR service for all federal employees [5] in the wake of private efforts by large corporations such as Intel, Dell, and Wal-Mart [4].

As adoption of EMRs continue and more hospitals begin to store clinical activities/courses of patients in EMRs, researchers can utilize information extraction (IE) and information retrieval (IR) methods that automatically seek out critical medical information in EMRs. Each EMR holds critical information about how medical practitioners diagnose and treat a patient, the underlying reasoning for the practitioner's decisions and actions, and the effect of the practitioner's recommendations on the patient. Furthermore, an aggregate set of EMRs can yield frequency statistics for patient prescriptions, diseases, epidemics, etc.

The wealth of information contained in EMRs and improving natural language processing (NLP) technologies make computer-assisted patient-care and quality assurance monitoring realistic possibilities. Such systems require accurate information extraction in order to locate relevant facts. Previous research efforts in medical informatics have found IE methods for extracting named entities and entity relations from one type of EMR, the hospital discharge summary [46]. We examine the next step in the IE process, coreference resolution.

1.2 Coreference Resolution

Coreference resolution is the effort to find nouns and pronouns that refer to the same underlying entity. Most coreference resolution systems have been developed for newspaper corpora [30, 36, 51]. In this thesis, we explore the application of coreference resolution techniques to medical corpora.

Medical corpora differ from newspaper corpora in several ways. Newspaper articles are unstructured, grammatical pieces of text written by journalists. In contrast, discharge summaries are usually semi-structured documents scattered with domain-specific linguistic characteristics and incomplete, fragmented utterances. Furthermore, these documents are based on doctor dictations rather than written text. This last fact is particularly important because written and spoken texts are rather different from each other. In addition, the authors of each corpus have different audiences. Newspaper articles must be written so that an average person can understand the material, while discharge summaries are intended for use by medical professionals. Due to this difference, discharge summaries contain extensive amount of domain specific vocabulary, short-hands, and abbreviations that are unlikely to appear in newspaper articles. These differences in structure, syntax, composition style, and vocabulary require adapting coreference resolution to medical records.

Because of their differences from newspaper articles, EMRs contains some of the more challenging problems in coreference resolution. While past research has found fairly accurate methods for resolving pronominal coreference [36, 11], it is generally accepted that noun phrase coreference resolution, particularly indefinite noun phrase coreference resolution, is a harder task [51, 60]. Indefinite nouns are generally more "ambiguous" than named-entities or proper nouns. By ambiguous, we mean that the same noun can refer to several different entities as it appears in different parts of the same text. For

example, a doctor may use the phrase "x-ray" to refer to two separate x-rays taken on two different days. Named-entities and proper nouns are less likely to be ambiguous than definite nouns because two different individuals or two different organizations that appear in the same text rarely have the same name; therefore, traditional methods that are well-suited for named-entity and proper noun resolution (e.g., string-match and edit-distance) have resulted in only limited success (F-measures around .72-.74) when applied to indefinite nouns [51, 60]. In this thesis, we take into consideration the particular characteristics of hospital discharge summaries to create a noun phrase coreference resolution system for these documents. We introduce new features and approaches to increase resolution performance for five different types of commonly found medical entities: diseases, symptoms, tests, treatments, and practitioners.

1.3 Thesis Structure

This thesis details our research efforts on the coreference resolution of five common medical semantic categories: medical practitioners (referred to as CONS in the rest of this document), treatments (referred to as MED), diseases (referred to as DIS), symptoms (referred to as SYMP), and medical tests (referred to as TEST). We next present a broad overview of information extraction and coreference resolution, detail our experimental methods, analyze the results from our experiments and discuss future works and conclusions.

2 BACKGROUND

2.1 Information Extraction Overview

An information extraction system automatically extracts entities, relations, and/or other information from text. Research efforts in IE intensified under a series of competitions called the Message Understanding Conference (MUC) [15]. In all, seven MUC conferences with foci in five different domains (naval operations message, terrorism in Latin America, business joint ventures, management change, satellite launch reports) were held between 1987 and 1998. While the corpus topic/domain changed from competition to competition, the texts were all extracted from newspaper articles. The grammatical, formal nature of newspaper articles provided a perfect starting point for exploring the information extraction task. As a result, the MUC competitions not only gave rise to many modern approaches to IE [24, 21], the competition itself also provided standards, frameworks, and methods for preparing and evaluating various IE tasks [21,55].

Some of the most commonly researched IE tasks to date are named-entity recognition [35, 15], terminology extraction [16, 22], event extraction [59, 58], semantic categorization [46], temporal/causal relation extraction [44], and coreference resolution [51, 36]. Most IE systems perform more than just one IE task [21, 24]. In these systems, each IE task will have its own main processing module (MPM) [21, 24]. In addition to the MPMs, systems employ preprocessing NLP modules to assist the MPMs in their IE task [21, 24].

2.1.1 Preprocessing Module

Preprocessing modules normalize and prepare text for use by other modules in the IE system [9]. These modules are the building blocks that make IE possible. Tokenizers, sentence splitters, part-of-speech taggers, and syntactic parsers are examples of preprocessing modules. Research in the past two decades has yielded powerful and reliable preprocessing tools such as the Brill part-of-speech tagger [10] and the Collins syntactic parser [20]. Table 1 explains the purpose of some of the most common preprocessing tools.

Preprocessing Module	Function
Tokenizer	Separates the text into tokens (words, numbers, punctuation marks)
Sentence Splitter	Assigns each individual sentence or statement block into its own line
Part-of-Speech Tagger	Assigns a part of speech for word tokens
Syntactic Parser	Outputs a parse tree that shows the syntactic structure of the sentence
Stemmer	Maps morphological variations of words to a single root

Table 1: Common NLP tools

2.1.2 Main Processing Module (MPM)

An MPM contains the prediction model for the IE task. Outputs from preprocessing modules are eventually routed to MPMs for processing. Often, MPMs may also take as input the output from other MPMs [21]. For example, a typical IE system would have dependencies as shown in the system in Figure 1 below, where certain MPMs like the named-entity (NE) coreference module depend on the output of the Simple Disambiguation MPM in addition to outside knowledge provided by the Knowledge and Information Management (KIM) ontology & knowledge base.

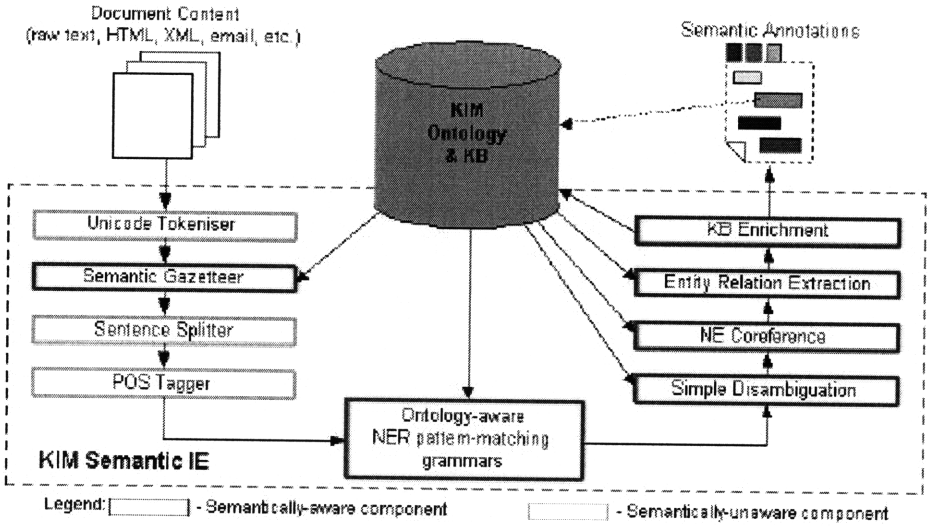

Figure 1: KIM semantic IE flow diagram [40]

2.1.3 A Sample IE System: CaRE

CaRE [46] is a discharge summary IE system developed by the Clinical Decision Making Group at MIT. The system has three main functionalities handled by three separate MPMs. The functionalities are: de-identification, semantic categorization, and semantic relation extraction.

The de-identification module replaces private information (e.g., dates, names, addresses, telephone numbers) with false data. The semantic categorization module identifies entities in eight semantic categories: diseases, symptoms, tests, results, medical treatments, dosage, abused substances, and medical practitioners. In addition, CaRE can also determine the presence-status of disease and symptom entities. Presence-status details whether a disease or symptom exists in the patient, possibly exists in the patient, does not exist in the patient, or exists in someone other than the patient. CaRE's semantic relation extraction module identifies semantic relations that exist between the entities in the eight semantic categories.

Semantic Category	Type	Description	Abbrev
PRACTITIONERS	Physical Entity	Medical teams and practitioners	CONS
DISEASES	Event	Disease, syndrome, or medical problem	DIS
TREATMENTS	Physical Entity / Event	A surgical/operational event or medication employed to treat a patient	MED
SYMPTOMS	Event	Signs or symptoms that describes the patient's feelings and/or physical state	SYMP
TESTS	Event	An event that measures an attribute, condition, or status of the patient	TEST
SUBSTANCES	Physical Entity	Abusive substance used by the patient, (cigarettes, drugs, alcohol)	SUBS
DOSAGES	Attribute	Attributes for TREATMENTS that detail the quantity, frequency, and manner in which the drug is delivered/used by the patient	DOS
RESULT	Attribute	Attributes that detail the outcome of TESTS	RESULT

Table 2: CaRE semantic categories

CaRE employs a suite of preprocessing modules including a stemmer, the Link Grammar Parser [47], and UMLS Norm [53]. The system then uses its MPMs which create predictive models using support vector machines (SVM). The semantic categorizer is of particular importance for this thesis because the coreference resolution system uses the semantic categorizer output.

2.2 Coreference Resolution

Coreference occurs when two strings (referred in MUC convention as markables) in a text refer to the same entity [54]. Two or more markables that refer to the same entity form a coreference chain. Sometimes, the term "equivalence class" can also be used. In general, all entities in a coreference chain agree in number, person, and gender. These entities have an equivalence relationship with each other and satisfy transitive closure. In other words, if markable i corefers to j and j corefers to k, then i must corefer to k.

As Jack was passing by the boutique store, an item in the display piqued his interest.

((Jack)$_4$ and the (store clerk)$_5$)$_1$ discussed the (price of the (item)$_6$)$_2$ at (great length)$_3$. After an (hour)$_3$ of negotiations, (they)$_1$ settled on an (agreeable number)$_2$. The (clerk)$_5$ congratulated (Jack)$_4$ on the (great find)$_6$.

note: markables in the same coreference chain contain the same subscript.

Figure 2: Examples of coreference in sentences

While coreference resolution is often confused with anaphora resolution [54], the two tasks are different. An anaphoric relationship holds when the leading markable in a markable pair (i.e., the antecedent) is required for the interpretation of the second markable (i.e., the anaphor). Because the antecedent is only needed for the contextual interpretation of the anaphor, it does not mean that they have to be equivalent to each other. For example, "every dog" and "its" in the sentence "Every dog has its day" are in an anaphoric relationship but they do not corefer [23].

While the examples above seem straightforward, coreference resolution is not always so clear. In fact, some relationships are open to interpretation depending on the reader's objective and in what context the markables are used. Consider the example of the closely related terms "cancer" and "tumor." A radiologist in his/her function may consider "tumor" and "cancer" to be equivalent terms because x-rays identifies the location of the tumor/cancer. On the other hand, an oncologist who deals much more in detail with cancer is likely to see "tumor" as only an aspect of the patient's cancer. These minute changes are very difficult to detect and interpret. In many cases, there is not a single right answer. We take a fine grain, literal approach to coreference resolution. Only in cases where it is obvious that two markables are coreferent, whether through syntax or context, do we draw coreference links. This interpretation lowers the number of coreference links drawn in a data set, but has the benefit of being less ambiguous and a more clearly-defined task.

2.2.1 Terms and Definitions

In this thesis, we adopt terminology from MUC. The only deviations from MUC convention are the terms antecedent and anaphor, which are normally used in anaphor resolution but are adapted to coreference resolution in this thesis for convenience..

Markable – a noun phrase that can be classified as one of CONS, DIS, MED, SYMP, or TEST as identified by CaRE.

Entity – the object, concept, physical entity, or event that a markable refers to.

Antecedent – in a pair of coreferent markables, the markable that appears earlier in text

Anaphor – in a pair of coreferent markables, the markable that appears later in text.

Coreference Chain – all markables within the scope of a single discharge summary that represent the same entity.

2.2.2 Convention

If two markables are represented by two letters, then the markable represented by the letter earlier in the alphabet is the antecedent, i.e., for a coreferent pair **i** and **j**, **i** is the antecedent and **j** is the anaphor.

2.2.3 Past Research

MUC-6 recognized coreference resolution as an independent IE task. Early coreference resolution modules often used rule-based logic and regular expressions to locate coreference chains [29, 30, 1]. These modules were high in precision and low in recall. Most of them were domain dependent [30].

McCarthy and Lehnert proposed applying a supervised machine learning approach to coreference resolution of nouns when they introduced RESOLVE [30]. They proposed a four step resolution process that became the standard framework used by machine learning approaches to coreference resolution: feature set determination, instance creation, learning, and clustering.

RESOLVE system was designed to handle four types of entities: organizations, facilities, persons, and products-or-services. McCarthy and Lehnert paired all markables with each other, represented each markable pair by a feature vector of eight features (as shown in Table 3), and used the feature vector during training and testing.

Feature	Description	Possible Values
NAME-i	Does reference I contain a name?	YES/NO
JV-CHILD-i	Does reference i reference to a joint venture child, i.e., a company formed as the result of a tie-up among two or more entities?	YES/NO/UNKNOWN
NAME-j	Does reference j contain a name?	YES/NO
JV-CHILD-j	Does reference j reference to a joint venture child, i.e., a company formed as the result of a tie-up among two or more entities?	YES/NO/UNKNOWN
ALIAS	Does one reference contain an alias of the other, i.e., does each reference contain a name and is one name a substring of the other name?	YES/NO
BOTH-JV-CHILD	Do both references, i and j, refer to a joint venture child?	YES, if JV-CHILD-i and j == YES NO, if JV-CHILD-i and j == NO UNKNOWN, otherwise
COMMON-NP	Do the references share a common noun phrase? Some references contain non-simple noun phrases, e.g., appositions and relative clauses.	YES/NO
SAME-SENTENCE	Do the references come from the same sentence?	YES/NO

Table 3: Features for the McCarthy and Lehnert coreference resolution system

McCarthy and Lehnert used the C4.5 decision tree algorithm to train a model for predicting whether pairs of markables coreferred to each other. They chose the C4.5 decision tree [42] "due to its ease of use and its widespread acceptance... ". The authors used 50-fold cross-validation to train and evaluate a data set with 1230 markable pairs (from 472 entities). Because RESOLVE performed only pair-wise coreference prediction, violations of transitive closure occurred. For example, if the system classified i as coreferent to j and j as coreferent to k, it was not necessarily true that it would classify i as coreferent to k. To solve this problem, the authors use the aggressive-merge clustering algorithm to assign markables to coreference chains. In aggressive-merge, any markables linked together explicitly or implicitly are clustered into the same coreference chain.

To evaluate RESOLVE's performance, McCarthy and Lehnert used the model theoretic approach [55] of the MUC-6 and MUC-7 competitions [16, 52]. We detail the model theoretic approach in section 3.4.2

System	Recall	Precision	F-Measure
RESOLVE (unpruned)	.854	.876	.865
RESOLVE (pruned)	.801	.924	.858
MUC-5 rule set	.677	.944	.789

Table 4: RESOLVE evaluation

McCarthy and Lehnert compared RESOLVE to a rule-based system that was used in their MUC-5 IE system. The rule-based system used lexical patterns to obtain information on markables (e.g., is the markable a company name, is it an alias, ...). Then it ran markable pairs through a set of hard-coded rules to determine if the markables were coreferent. McCarthy and Lehnert found the F-measures of the rule-based system to be much lower than those of RESOLVE. Given the relatively small sample size used by McCarthy and Lehnert, however, it is possible that the differences in system performance were not, in

fact, statistically significant. Even so, the results obtained by McCarthy and Lehnert were encouraging for a system that employed only eight features. Overall, RESOLVE showed the great potential for applying machine learning approaches to the coreference resolution problem. Various efforts based on RESOLVE followed [48, 36, 11]. Soon et al. [48] introduced the DSO system that extended the decision tree learning approach to include both noun and pronoun resolution. The authors proposed many new features that did not exist in RESOLVE. These features include sentence distance, number agreement, semantic category agreement, gender agreement, and various features to distinguish different types of references (i.e., appositives, pronouns, definite nouns, demonstrative nouns, and proper names). The markable pairing approach was also markedly different from McCarthy and Lehnert's exhaustive method. Soon et al. chose to limit the training data size so that training time could be cut down. In a chain A-B-C-D, the algorithm only selected neighboring entity pairs A-B, B-C, and C-D, as positive training examples, rather than all possible combinations . If there were non-coreferent markables a, b, and c between A and B (e.g. A-a-b-c-B-C-D), then the negative examples A-a, A-b, and c-B would be generated. As a result of this selectivity, each positive markable in a chain would appear in a maximum of two positive samples and any negative markables that appear between the antecedent and anaphor would only be paired with the closer of the two markables to generate a negative sample.

During testing, the classifier found the closest satisfactory antecedent to each markable in the document, starting from the second markable. As soon as the classifier found an antecedent candidate similar enough to the current markable, the classifier assigned a coreferent link between the antecedent and the markable, the classifier then moved to the next markable in the document. In doing so, the algorithm assumed that the first antecedent to be accepted was probably the best possible antecedent for the current markable.

Soon et al.'s algorithm presented results comparable to other non-machine-learning approaches on the MUC-6 and MUC-7 corpora. The authors compared the performance of their DSO system to that of RESOLVE and a RESOLVE-DSO hybrid (referred to as DSO-TRG). The hybrid used RESOLVE's exhaustive pairing method to train a prediction model on DSO's feature set. The authors, however, do not mention how prediction and clustering were done.

System	Recall	Precision	F-Measure	Description
Complete systems				
DSO	.586	.673	.626	The Soon, Ng, and Lim system
DSO_TRG	.526	.676	.592	DSO system using RESOLVE's method of generating positive and negative examples
RESOLVE	.442	.507	.472	The McCarthy and Lehnert system
Baseline systems: DSO using just one feature				
DIST	0	0	0	Only "distance" feature is used
SEMCLASS	0	0	0	Only "semantic class agreement"
NUMBER	0	0	0	Only "number agreement"
GENDER	0	0	0	Only "gender agreement"
PROPER_NAME	0	0	0	Only "both proper names"
ALIAS	.245	.887	.384	Only "alias"
J_PRONOUN	0	0	0	Only "j-pronoun", answers the question is markable j a pronoun?
DEF_NP	0	0	0	Only "definite noun phrase"
DEM_NP				Only "demonstrative noun phrase"
STR_MATCH	.457	.656	.539	Only "string-match"
APPOSITIVE	.039	.577	.073	Only "appositive"
I_PRONOUN	0	0	0	Only "i-pronoun"
Other baseline systems with DSO				
ALIAS_STR	.515	.664	.580	Only the "alias" and "string-match" features
ALIAS_STR_APPOS	.552	.664	.603	Only the "alias", "string-match", and "appositive" features
ONE_CHAIN	.889	.318	.470	All markables form one chain
ONE_WORD	.554	.336	.441	Markables corefer if there is at least one common word
HD_WORD	.564	.504	.532	Markables corefer if their head words are the same

Table 5: Performance of DSO on MUC-6 data

Out of the three complete systems (see Table 5), DSO performed significantly better than RESOLVE. The primary reason cited by the authors for the difference in performance was that RESOLVE had a very low recall on the MUC data set because it was designed only to handle "persons", "entities", and "organizations", whereas DSO handled resolution for all entity types specified by the MUC standard. DSO_TRG also performed worse than DSO.

Soon et al. also evaluated how each of the features from their 12-feature set individually influenced DSO's performance. Out of the 12 features that were tested, only the single feature ALIAS, APPOSITIVE, and STR_MATCH systems resulted in non-zero F-measures. Soon et al. then evaluated how a system with the previous three feature sets performs against the complete system with the full-feature set. The three feature system (ALIAS_STR_APPOS) did extremely well, performing only 2.3% worse than the 12 feature system. In fact, features such as SEMCLASS, PROPERNAME, DEF_NP, and DEM_NP were not even used in the MUC-6 prediction tree model.

While the DSO approach seems to be better than RESOLVE in both performance and training time, it has a major flaw. Its "closest satisfactory candidate" clustering approach causes a cascading error effect. If the system assigns a markable to the wrong chain then all anaphors that are correctly classified as coreferent to the markable would also be assigned to the wrong coreference chain.

In a series of papers from 2002-2005 [36, 37, 38], Ng and Cardie proposed several improvements over Soon et al.'s DSO algorithm. They explored modifications to Soon et al.'s sample selection process and the effect of an expanded feature set on resolution performance [36]. Rather than Soon et al.'s "closest satisfactory candidate" approach to assigning antecedents to anaphors, Ng and Cardie proposed "best-first clustering". In this approach, the machine finds the highest scoring candidate out of all antecedents (taken as the certainty of the machine's prediction) rather than stopping at the candidate with a satisfactorily high score. This modification eliminated DSO's error propagation problem. Both the training and testing pair selection process were modified to accommodate this change. A further modification was the introduction of different string-match features for pronouns, proper names, and non-pronominal noun phrases (NPs). The authors observed that string-match worked better for certain categories of markables (proper names rather than pronouns, for example), and created separate string-match features for indefinite nouns, definite nouns, and pronouns, thereby giving the system the option of using the appropriate string-match mechanism for each type of markable. These modifications to the original Soon et al. framework did result in a slight system performance gain.

Ng and Cardie also extended the features of Soon et al., creating a set of 53 features. However, they observed mixed results in response to the added features. When including all 53 features, the system's recall increased but its precision fell dramatically, causing the system F-measure to be worse than the baseline. In order to get the best performance (F-measure of 70.4% and 63.4% for MUC-6 and MUC-7 respectively), the authors manually selected a subset of the features. The selected features were ALIAS, SOON_STR_NON_PRO, ANIMACY, PRO_STR, PRO_RESOLVE, APPOSITIVE, GENDER, PRONOUN_1 MAXIMALNP, MNCLASS.

Ng explored other add-ons to the learning-based coreference resolution approach [37, 38]. He [37] used the cost ratio, i.e., cost of misclassifying a positive instance / cost of misclassifying a negative instance, to improve performance of systems implemented with RIPPER [17], a constraint-based rule learning algorithm, and MaxEnt [6, 33], a probabilistic machine learner. The cost ratio can be applied to algorithms during learning to adjust the algorithm's preference to misclassify positive or negative instances. Ng reasoned that changing the cost ratio can fine-tune a system's performance. Ng termed the cost ratio adjustment a "global optimization" because rather than improving system performance by introducing new features, Ng is simply optimizing the entire system by finding the best cost ratio for training. In his follow up work, Ng expanded the global optimization approach one step further [38]. He gathered up a variety of methods for each of the four parts of his coreference resolution system (i.e., the learning algorithm, the instance creation method, the feature set, and the clustering algorithm). He then used a SVM ranking algorithm to pick the combined system that performed best over 3 corpora from the ACE competition.

Coreference Step	Method	Relevant Literature
Learning Algorithm	Decision tree learners (C4.5/C5/CART)	Aone and Bennet (1995), McCarthy and Lehnert (1995), Soon et al. (2001), Strube et al. (2002), Strube and Muller (2003), Yang et al. (2003)
	RIPPER	Ng and Cardie (2002b)
	Maximum entropy	Kehler (1997), Morton (2000), Luo et al. (2004)
Instance creation method	McCarthy and Lehnert	McCarthy and Lehnert (1995), Aone and Bennett (1995)
	Soon et al.	Soon et al. (2001), Strube et al. (2002), Iida et al. (2003)
	Ng and Cardie	Ng and Cardie (2002b)
Feature set	Soon et al.	Soon et al. (2001)
	Ng and Cardie	Ng and Cardie (2002b)
Clustering algorithm	Closest-first	Soon et al. (2001), Strube et al. (2002)
	Best-first	Aone and Bennett (1995), Ng and Cardie (2002b), Lida et al. (2003)
	Aggressive-merge	McCarthy and Lehnert (1995)

Table 6: Methods at each step that were combined by Ng's system [38]

Test Set	Scoring Program	Average Rank	Coreference System			
			Instance Creation Method	Feature Set	Learner	Clustering Algorithm
BNEWS	MUC	7.5249	McCarthy and Lehnert	Ng and Cardie	C4.5	aggressive-merge
	BCUBED	16.902	McCarthy and Lehnert	Ng and Cardie	C4.5	aggressive-merge
NPAPER	MUC	1.4706	McCarthy and Lehnert	Ng and Cardie	C4.5	aggressive-merge
	BCUBED	9.3529	Soon et al.	Soon et al.	RIPPER	closest-first
NWIRE	MUC	7.7241	McCarthy and Lehnert	Ng and Cardie	C4.5	aggressive-merge
	BCUBED	13.1379	Ng and Cardie	Ng and Cardie	MaxEnt	closest-first

Table 7: Best performing combined coreference system for each test set

Ng evaluated systems resulting from various combinations of his coreference resolution system parts using both the MUC (Model-theoretic) and the B-CUBED metrics. The MUC evaluation overwhelmingly favored McCarthy and Lehnert's instance creation method, while the B-CUBED evaluation resulted in inconclusive findings.

While Ng and Soon et al. addressed how to best formulate the coreference resolution problem into a binary classification problem, other researchers aimed to understand what types of features are best suited for coreference resolution in limited domains. Of particular interest to this thesis are two pieces of research performed on biomedical corpora.

Castano et al. [12] presented a sortal and pronominal anaphora resolution system for Medline abstracts

that detail biomolecular relations. Sortal anaphors occur when a quantifier such as "both" is used to refer to two markables. Castano et al. used MetaMap [27] to identify markables and their respective semantic types. They then assigned pronouns and sortal nouns to the identified markables by computing a salience score using the metrics in Table 8:

Feature	Score
Person/Number Agreement	+1
Person/Number Disagreement	No Salience
NP String Similarity (LCS)	+1 to +3
Semantic Type(s) Matching	+2 per match
Coercion Type(s) Matching	+2 per match
No Matching Semantic Types	-1
Bio. Antec. for Pronominal	+2
Non-Bio. Antec. for Pronoun	-2

Table 8: Features used to determine coreferent anaphor/antecedent pairs

Each feature from the table above contributes to the salience score for an antecedent and anaphor pair. Person/Number agreement tests if two markables are both first, second, or third person pronouns and that both pronouns agree in terms of plurality. NP string similarity uses the longest common subsequence (LCS) method to calculate string similarity between two markables. Semantic type matching assigns scores to each pair based on the number of matching semantic types that the two markables have in common. Coercion expands on the semantic type matching feature by assigning additional implied semantic types to each markable. Coercion occurs if a markable is a patient or agent of a biomedical verb. The markable is coerced to bear the implied semantic types that are frequently associated with the agent or patient of the verb. The "Biomedical Antecedent for Pronominal" feature increases the markable pair score if the antecedent is a biomedical term.

Two features from Table 8 are noteworthy: NP string similarity and number of matching semantic types. Rather than traditional exact string-matching, Castano et al. used LCS for the string similarity metric so that morphological variants such as "grafts" and "xenografts" could be identified. Castano et al. also made the observation that markables often receive more than one UMLS semantic type assignment. They believed that the more semantic types two markables have in common the more similar they are. To capture this belief, they assign a +2 salience score for every semantic type that two markables have in common. While both features seem like good ideas, the paper did not evaluate individual feature contributions to the system's performance to test this hypothesis.

Yang et al. [61] presented a noun phrase resolution system for Medline abstracts. They explored various methods of improving coreference resolution. They distinguished antecedents and anaphors into different noun phrase types (definite, demonstrative, indefinite, pronouns, and proper noun). Each markable was also assigned an attribute list that contained the various types of modifiers (number, comparative adjective, superlative adjective, etc.) which described it. Like Castano et al., Yang et al. did not provide a detailed analysis of the contribution of each feature to the system performance. Instead, Yang et al. focused on examining how different string-match methods altered the performance of the system. In particular, they examined the following features:

Contain(S1, S2) – is S1 contained in S2

ContainRatio1(S1, S2) – number of common tokens between S1 and S2, normalized to the number of total tokens in S1. Tokens are assigned equal weights.

ContainRatio2(S1, S2) - number of common tokens between S1 and S2, normalized to the number of total tokens in S1. Tokens are assigned different weights based on frequency statistics.

COS-Similarity(S1,S2) – commonly used information retrieval statistic for calculating similarity between documents and sentences.

Yang et al. found using ContainRatio1 resulted in the highest recall, while COS-Similarity produced the highest precision. In F-measure, ContainRatio1 performed better than the other systems.

String-match is a powerful ally for coreference resolution systems [30, 61, 48], but exact string-match captures limited information. As a result, various authors [51, 18, 19] have proposed using distance metrics to better model similarity between strings.

Strube and Müller [51] used the minimum edit-distance along with features from [48] and [11] to perform coreference resolution. They defined two minimum edit-distance features for each markable pair, one based on the anaphor and the other based on the antecedent. The value for each is evaluated as:

$$\text{Minimum Edit Distance (i, j)} = \frac{\text{Length(i)} - \text{Distance(i, j)}}{\text{Length(i)}}$$

$$\text{Minimum Edit Distance (j, i)} = \frac{\text{Length(j)} - \text{Distance(i, j)}}{\text{Length(j)}}$$

Equation 1: Minimum edit-distance

The value is a word-length-normalized edit-distance for each markable. The added minimum edit-distance features resulted in a .08 F-measure increase over the baseline features. They specifically helped resolve definite nouns and named entities, improving each category's F-measure by .18 and .11 respectively.

Research in Other Related IE Tasks

Research results from other IE tasks like name-matching and name-disambiguation are also of use to coreference resolution. Name-matching systems try to cluster similar names in a document. These similar names may refer to different entities. The name-matching task does not guarantee disambiguation of name markables into different entities. However, it can be used to identify potentially coreferent markables.

Cohen [42] first proposed employing the term frequency, inverse document frequency weighted (TFIDF-weighted) cosine similarity metric to evaluate similarity between named markables. The cosine similarity metric is a vector-based approach to evaluating similarity between markables. A markable is represented by a token vector that contains a value for each unique token in the markable. Tokens that do not exist in the markable have a value of 0. The cosine similarity metric evaluates similarity between two markables by finding the cosine of the angle of separation between two token vectors. TFIDF-weighted cosine

similarity metric simply uses the TFIDF method for determining the appropriate score for a token in the markable.

The TFIDF-weighted cosine similarity score between two markables S and T can be defined as:

$$\text{CosSim}(S,T) = \sum_{w \in S \cap T} \frac{V(w,S)V(w,T)}{\sqrt{\sum_{w'} V(w,S)^2 \sum_{w''} V(w,T)^2}}$$, where w is a token in both S and T.

$$V(w,S) = \log(TF_{w,S} + 1) \times \log(IDF_w)$$

Where $TF_{w,S}$ is the frequency of word w in markable S, IDF_w is the inverse of the fraction of markables in the entire corpus that contain w. V(w, T) can be obtained analogously.

Equation 2: TFIDF-weighted cosine similarity score

In a follow up paper [19], Cohen proposed a variant of TFIDF called SoftTFIDF. For two elements S and T, the SoftTFIDF score evaluated V not only on tokens that appear in both S and T, but also on tokens in S and T that are similar to each other. The authors considered tokens that had a Jaro-Winkler score (for definition see [57]) greater than 0.9 to be "similar". SoftTFIDF proved to be extremely powerful for name-matching, resulting in F-measures of .89 and .85 for two different corpora, while the next best distance metric, TFIDF-weighted cosine similarity, had F-measures of .79 and .84 respectively.

Li, Morie, and Roth [28] compared Cohen [19] and Bilenko's [7] discriminative approaches of disambiguating names to an unsupervised generative approach they called LMR. They proposed a system that "automatically [extracted] active features in a data-driven way for each pair of names." The system contained both relational and structural features. Relational features detail how names are related to each other, while structural features detail the token structure similarities between the two names. We give examples of relational features in Table 9 and then give further examples of structural features later. LMR employed 13 features in all, each representing a condition examined by the system. If a particular condition is satisfied, then the corresponding feature is activated.

Feature	Activation Condition	Activation Example
Honorific Equal	both tokens are honorifics and equal	"Mr." and "Mr."
Honorific Equivalence	honorifics are not equal but equivalent	"Professor", "Prof."
Honorific Mismatch	honorifics are not equal	"Mr." and "Professor"
NickName	If one token is a nickname of the other	"Thomas" and "Tom"
Equality	both names are equal	"Thomas" and "Thomas"
Edit Distance	the tokens have an edit-distance of 1	"Edit distance of 1"
Symbol Map	one token is a symbolic representative of the other	"and" and "&"

Table 9: Relational features used by LMR

Relational features however do not completely disambiguate names. The authors gave the example of two token pairs ("John Kennedy", "John Kennedy") and ("John Kennedy", "John Kennedy Davis"). Both pairs have the same active features; however, the two pairs are clearly different in structure. Li et al.'s structural features captured this information by describing how tokens from two strings map to each other.

The previous examples would respectively have structure features "(1,2)", "(1,2)" and "(1,ø,2)", "(1,3)". Each token's structure value represents the token in the other string that it matches to.

Li et al. evaluated their system on three types of named-entities: people, location, and organization names. For each of these three named-entity types, Li et al.'s generative approach was able to perform better than Bilenko's Marlin [7] and Cohen et al.'s SoftTFIDF [18].

	F-measure of Marlin	F-measure of SoftTFIDF	F-measure of LMR
People	88.3	89.0	90.5
Location	77.3	90.5	92.5
Organization	78.1	87.7	93.0

Table 10: Pair-wise classification performance (F-measure) for three name-matching algorithms [28]

While the aforementioned research primarily explore name-matching by exploiting various string-match distance metrics, Pedersen et al. [39] proposed using a lexical clustering approach to find coreferent markables. In lexical clustering, a machine groups names together based on the similarity of their surrounding context. Specifically, Pedersen et al. identify salient bigrams, consecutive words that appear in the text in a 50-word window around each name. A bigram is salient if the log-likelihood ratio between the word pair is above a threshold. Name occurrences are clustered based on common bigrams that they share when certain stop-words are removed. The lexical clustering approach yielded encouraging results.

Lessons from Literature

Coreference resolution efforts historically have been performed on newspaper articles/corpora where a large proportion of coreferring distinct noun markables can be identified by using string-match and semantic features; but even so named-entity resolution is often considered "more problematic" than resolution of other types of markables [51]. While much progress has been made in the newspaper corpus domain, research has not yet diversified to other domains.

In our literature search, we were only able to find two coreference resolution efforts that focused on the biomedical domain [12, 61]. These efforts used Medline abstracts for their research. While Medline abstracts are more related to the corpora of this thesis than newspaper articles, they are still fairly different from discharge summaries. Though they share vocabulary with medical discharge summaries, Medline abstracts are research abstracts that are written in a grammatical, scientific style, while medical discharge summaries are dictated by doctors in an informal style. Furthermore, Castano et al. and Yang et al. presented relatively little information on how individual features contributed to their systems' performance. Their focus was on problem solving approaches.

It is still unclear what features are most powerful for resolving coreference of markables that appear in medical discharge summaries. We believe the salience of the features (orthographical, grammatical, temporal, etc.) may vary depending on the type of coreferring entity involved. For example, medical events such as surgeries and radiology investigations may rely heavily on semantic clues such as time, because practitioners may request the same type of medical treatment or investigation several times over the course of a patient's hospital stay. In hospital discharge summaries, the dictating doctor may refer to these recurring events using the same noun, increasing the complexity of coreference resolution. We use the C4.5 decision tree algorithm to train prediction models for nominal and named-entity coreference resolution in hospital discharge summaries. Previous works identified string-match distance metrics as extremely important to the performance of a coreference resolution system [30, 48, 61]. Yang et al.

surprisingly found that token frequency statistics adjustments do not improve similarity metrics. We employ a string-match distance metric that is equivalent to Yang et al.'s ContainRatio. Our approach to resolving coreference is to present the system with a large feature set and use a "multi-perspective" approach to improve the salience of each feature. Large feature sets have been known to improve system performance [36, 61]. Unlike previous feature sets that include only two or three types of features [61, 12], our feature set includes features that detail orthographic, semantic, syntactic, lexical, morphological, temporal and domain specific information of our corpora. Many of these features (e.g., date-default-unknown, word-distance, sentence-entity) did not appear in any previous literature we found. Our "multi-perspective" approach (explained in the Feature Set section) also presents a new, yet simple way to present feature values to a system.

3 RESOURCES

3.1 UMLS

The Unified Medical Language System (UMLS) is an effort headed by the National Library of Medicine (NLM) to organize biomedical concepts and knowledge into a single comprehensive digital knowledge source [27]. NLM's goal in developing UMLS is to provide working knowledge sources and tools to assist researchers in medical informatics. UMLS consists of three main knowledge repositories: the Metathesaurus, the Semantic Network, and the SPECIALIST Lexicon.

The Metathesaurus is a database composed of biomedical concepts, concept names, and various relationships between concepts. This information is compiled from various medical thesauri, classification standards, cataloged biomedical research and other source vocabularies such as ICD-9 and SNOMED CT. Biomedical concepts are the atomic, distinct terms that make up the Metathesaurus. Different source vocabularies may refer to the same biomedical concept with a variety of concept names. It is also possible that they may use the same concept name to refer to different biomedical concepts in the Metathesaurus. The Metathesaurus does not attempt to resolve any such conflicts. It simply stores all concept-to-name and name-to-concept mappings.

The Semantic Network stores a set of UMLS semantic types (e.g., organisms, surgical procedures, and medical occupations) and any semantic relations that exist amongst them. The semantic types in the Semantic Network are intended to be used to categorize Metathesaurus concepts, while the semantic relations detail how the semantic types are physically, spatially, temporally, functionally, and conceptually related to each other.

The last knowledge repository is the SPECIALIST Lexicon. The lexicon contains linguistic information on both biomedical vocabulary present in the UMLS Metathesaurus and on common English vocabulary. Specifically, it details the syntactic, morphologic, and orthographic information related to each entry it contains.

NLM provides tools to help researchers extract information from these knowledge repositories. A lexical tool called LVG contains various methods for matching spelling, derivational, or inflectional variants. The tool Norm is a specific set of LVG functions run in sequence to normalize a string. Norm removes case, punctuation, possessive markers, and inflection from words; and as a final step, it sorts tokens in the string in alphabetical order.

One tool that is particularly useful for this thesis is MetaMap. MetaMap [2] is a NLP system designed to map text strings to concepts in the UMLS Metathesaurus. MetaMap uses a five step process to map concepts in UMLS to a given string input. We include an excerpt from [2] that gives an overview of each step of the process.

1. Parse the text into noun phrases and perform the remaining steps for each phrase;

2. Generate [a candidate set of Meta strings which contain] the variants for the noun phrase where a variant essentially consists of one or more noun phrase words together with all of its spelling variants, abbreviations, acronyms, synonyms, inflectional and derivational variants, and meaningful combinations of these;

3. Form the candidate set of all Meta strings containing one of the variants;

4. For each candidate, compute the mapping from the noun phrase and calculate the strength of the mapping using an evaluation function [UMLS Score]. Order the candidates by mapping strength; and

5. Combine candidates involved with disjoint parts of the noun phrase, recompute the match strength based on the combined candidates, and select those having the highest score to form a set of best Meta mappings for the original noun phrase.

Figure 3: Algorithm for Metathesaurus mapping [2]

3.2 C4.5 Decision Tree

The C4.5 [42] decision tree algorithm is a supervised machine-learning algorithm that improves upon the ID3 decision tree [41]. Generally, machine-learning algorithms have training and testing stages. During training, the learner trains a prediction model on an already classified data set. During testing, the trained prediction model is used to classify new data points. Data is presented to the learner in the form of feature vectors. Each feature vector contains a set of n features that encapsulates a data point in a data set.

A decision tree trains its predictive model by repeatedly examining and partitioning data on different features until a stop condition is satisfied. The prediction model records the features that are examined and the chosen split for each step. C4.5 determines which feature to partition by evaluating the information gain or the information gain ratio due to each feature. Information gain, G(E,f), from a feature f is the change in the entropy of the data samples due to f. In other words, information gain measures the amount of new information gained from partitioning on feature f. However information gain is inherently biased towards features with multiple values, therefore, a normalized information gain metric, the information gain ratio is often used.

$$\text{GainRatio}(E, f) = \frac{G(E, f)}{-\sum P(v)\log(P(v))}$$

where:

$$G(E, f) = H(E) - H(E \mid f)$$

$$H(E) = -\sum P(c)\log(P(c))$$

$P(c)$ = probability that an event in E has category c.

Equation 3: The information gain ratio [62]

3.3 Kappa Statistic

The Kappa statistic is the standard method used to evaluate inter-annotator agreement between annotated data. It is a measure of annotator agreement beyond simple chance. The Kappa statistic is often used as a measuring stick for the difficulty of a task. A task that results in a high K (> .6) can be considered doable for machines. A low K implies low inter-annotator agreement. In such a case, the task is either too difficult or too subjective.

K is defined as:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

Where:

P(A) is the observed agreement probability between two annotators
P(E) is the expected agreement by pure chance.

Equation 4: Kappa statistic

To compute Kappa on our data set, we treat each markable pair as a data point for a 2x2 confusion matrix (see below). One axis of the confusion matrix represents how Annotator 1 marked the pair and the other axis represents how Annotator 2 marked the pair. P(A) and P(E) can be calculated from the confusion matrix.

		Annotator 1		
		Coref	Non-Coref	
Annotator 2	Coref	M	N	M+N = P
	Non-Coref	O	L	O+L = Q
		M+O = R	N+L = S	M+N+O+L = T

Table 11: Annotator agreement confusion matrix

P(A), the observed agreement probability between two annotators, is simply (M+L)/T. We calculate the expected agreement due to chance, P(E), using simple probabilities. Assuming the probability that annotator i assigns a pair to a category z, P(i, z), can be calculated by the number of pairs the annotator assigned to z over the total number of pairs, then we simply need to find, for each category, the probability that all annotators assign a pair to that category. We can then sum the probabilities to find the probability of annotator agreement over all categories, $\sum_z \prod_i P(i,z)$. For the two-annotator, two-class case, $P(E) = \frac{P}{T} \times \frac{R}{T} + \frac{S}{T} \times \frac{Q}{T}$.

Due to the nature of the coreference resolution task, the Kappa statistic is only a rough measure of inter-annotator agreement. Coreference resolution involves assigning a markable to a coreference chain; however, because the Kappa score evaluates coreference relationship in a pair-wise fashion, it over-penalizes markables wrongly assigned or left out of long coreference chains. For example, if a markable

is misplaced in a long coreference chain, then it will result in many disagreeing markable pairs. As a result, the confusion matrix count will increase by more than 1. However, a false alarm link that identifies two standalone markables as coreferent will only result in a single false alarm. As a result of this over penalization, inter-annotator agreement may actually be higher than what's indicated by Kappa.

3.4 Evaluation Methods

3.4.1 Precision, Recall, and F-measure

In information extraction, the standard metrics of algorithmic performance are precision, recall, and F-measure. Precision measures the accuracy of the system's predictions, while recall measures the percentage of the gold standard accurately predicted by the system. F-measure is a single measure derived from combining precision and recall of a system.

The most basic coreference resolution evaluation is to treat each markable pair as a data point and to predict whether they corefer or not. A pair that is falsely predicted to be coreferent is called a false alarm, a pair that is falsely predicted to be non-coreferent is called a false negative, and a pair that is correctly predicted to be coreferent is called a hit. Given statistics about hits, false alarms, and false negatives, precision, recall, and F-measure are defined as follows:

Precision

$$P = \frac{\text{Hits for class i}}{\text{Hits for class i} + \text{False alarms for class i}}$$

Recall

$$R = \frac{\text{Hits for class i}}{\text{Hits for class i} + \text{False negatives for class i}}$$

F-measure

$$F = \frac{(1+\beta)(P \times R)}{P + R}$$

Equation 5: Traditional precision, recall, and F-measure evaluation

F-measure uses β to adjust the relative weight on precision and recall. For our purposes, we give equal weight to precision and recall; therefore, we set β to be equal to 1.

For an n-class classification problem, n sets of precision, recall, and F-measures can be reported. Even though coreference resolution is a binary classification problem, researchers only report the statistics for the coreferent class because non-coreferent prediction is almost always near 1.

This approach to measuring coreference resolution system performance is clearly lax. Because this metric is only a pair-wise evaluation, it does not penalize contradicting links that violate transitive closure. That is, if a system outputs **i** as coreferent with **j**, and **j** as coreferent with **k**, but **i** as non-coreferent with **k**, then the result would be two hits and a false alarm. However, the result is somewhat

meaningless, since the system output contains contradictions.

3.4.2 The Model Theoretic Scoring Method

The most widely used evaluation standard is the model theoretic approach, also known as the MUC approach to evaluating coreference resolution [55]. This approach requires that the system output satisfy transitive closure. It then computes the minimal number of links that would need to be added or taken away from the prediction so that it matches the gold standard. The links that need to be added are treated as false negatives, while the links that need to be taken away are false alarms.

Let S_i contain markables in a coreference chain in the gold standard and $P(S_i)$ be S_i partitioned into subsets containing the intersection of S_i with the prediction coreference chains that contain markables in S_i.

For example, if a chain A-B-C-D is classified as A-B, C, D, then:

$S_i = \{A,B,C,D\}$,

$p(S_i) = \{\{A,B\}, \{C\}, \{D\}\}$

$$R = \frac{\sum(|S_i| - |p(S_i)|)}{\sum(|S_i| - 1)}$$

The precision can be found by reversing the role of the gold standard and the prediction.

Equation 6: Model theoretic approach to calculating recall and precision

Bagga et al. [3] argues that model theoretic scoring is useful for information retrieval but inadequate for evaluating coreference resolution because of two major shortcomings. First, model theoretic scoring does not give credit for correctly predicting standalone markables. This problem occurs because the model theoretic approach evaluates the precision and recall of links that exist in the gold standard and the system prediction. It therefore captures correctly assigned coreference links and any falsely assigned coreference links. This method correctly accounts for any coreference chain in the gold standard with more than one markable, but what of the stand alone markables that shouldn't be linked to any other markable? If the system prediction incorrectly assigns the markable to a chain, then the mistake would be recognized, however if the system prediction is correct and leaves the markable by itself, then no credit is given for the correct classification! In addition, the model theoretic approach treats all links equally. However, one can argue that certain link errors should be penalized more than others. For example, Bagga et al. [3] argues that a spurious link that links two coreference chains with 10 items each should be penalized more than a link that connects two stand-alone markables.

3.4.3 B-CUBED

In response to the shortcomings of the model theoretic approach, Bagga et al. introduced the B-CUBED scoring metric. B-CUBED is an entity-based approach to evaluating system performance. The algorithm computes the precision and recall for each entity in the document and then performs a weighted average calculation to find the overall precision and recall.

$$\text{Precision}_i = \frac{\text{number of correct markables in the prediction chain containing Markable}_i}{\text{number of markables in the prediction chain that contains Markable}_i}$$

$$\text{Recall}_i = \frac{\text{number of correct markables in the prediction chain containing Markable}_i}{\text{number of markables in the gold standard chain that contains Markable}_i}$$

$$\text{Precision} = \sum w_i \text{Precision}_i$$

$$\text{Recall} = \sum w_i \text{Recall}_i$$

$w_i = 1/N$, where N is the total markable count

Equation 7: B-CUBED approach to calculating precision and recall

The B-CUBED resolves both shortcomings of model theoretic scoring by calculating precision and recall based on the coreference chains that each markable belongs to. In this manner, even stand-alone markables are taken into account. By evaluating what percent of the gold standard and the system prediction are commonly shared markables, B-CUBED penalizes wrong assignments to long chains by assigning those markable a lower score. The B-CUBED measure is more accurate than the model theoretic approach for ranking whether systems predictions are close to the gold standard. However, the B-CUBED approach is not perfect. While the B-CUBED F-measure is a good indicator for how system performance compare to each other [3], it sometimes yields very unintuitive precision and recall. Consider the examples below,

Truth:	A-B-C D	
Prediction1:	A-B-C-D	Precision = ¼(3 x (3/4) + 1 x (1/4)) = 5/8
		Recall = ¼(3/3 + 1/1) = 1
Prediction2:	A B C D	Precision = ¼(4 x 1/1) = 1
		Recall = ¼(3 x 1/3 + 1/1) = 1/2

Figure 4: Illustrations of B-CUBED evaluation

Both sample evaluations showcase some unintuitive behavior when using the B-CUBED evaluation. In both cases, the B-CUBED evaluation is overly optimistic. In Prediction1, recall should certainly not be one because the gold standard states that there should be two chains (A-B-C and D) in the output, however only one chain (A-B-C-D) is predicted. In Prediction2, the system predicts four chains (A, B, C, D), if each predicted chain can only be assigned to one coreference chain in the gold standard, then the prediction should have a lower score than 1.0. These problems arise because when evaluating a prediction coreference chain, the B-CUBED algorithm considers each markable's precision and recall separately. As a result, multiple gold standard chains that contain the prediction chain's markables will contribute to its precision and recall. However, in reality, based on the definition of coreference resolution, there should be a one-to-one mapping between the gold standard and prediction chains.

3.5 Significance Testing - Wilcoxon signed-rank test

We employ the Wilcoxon signed-rank test [56] to determine whether two scores are statistically significantly different from each other. The Wilcoxon signed-rank test is a statistical test used to determine if there is significant difference between two dependent data sets. It is the nonparametric equivalent of the paired Student's t-test. Unlike a parametric test such as Student's t-test, a nonparametric test does not assume that its data set has a specific distribution, e.g. normal, Gaussian, uniform, etc. The Wilcoxon signed-rank test simply assumes that its data set is symmetric and that data points, Z_i, are independent of each other.

The Wilcoxon signed-rank test has been used extensively in NLP research. Most recently, the NIST 2007 Automatic Content Extraction Evaluation (ACE07), an NLP conference similar to MUC, used the Wilcoxon signed-rank test to evaluate the significance of score differences between different systems. Similar to previous works in NLP [49, 50, 43], we employ the Wilcoxon signed-rank test to determine if the F-measure scores between different configurations of our system result in significant changes in resolution performance.

The Wilcoxon test validates the null hypothesis, H0 : $*\theta = 0$ (there is no difference between the data sets Y_i and X_i) against H1 : $*\theta > 0$ or $*\theta < 0$ (there is a difference between the data sets). The test ranks the difference of each data points, $|z_i|$, then calculates the Wilcoxon signed-rank statistic:

$$T+ = \sum z_i \Psi_i, \text{ where } \Psi_i \text{ is an indicator function}$$

$$\Psi_i = \begin{cases} 1, \text{ if } z_i > 0, \\ -1, \text{ if } z_i < 0 \end{cases}$$

Equation 8: Wilcoxon signed-rank statistic

The calculated T+ value is then matched to a table to find the probability p, that the null hypothesis is false at a given confidence level alpha. We set alpha = 0.05.

We use the Wilcoxon signed-test as implemented by the Harvard Neural Systems Group [26].

4 COREFERENCE RESOLUTION

We apply a machine-learning approach, along with an assortment of features that include lexical, syntactic, and orthographic information, to resolve selected entity markables that are in hospital discharge summaries. We focus on coreference resolution of nouns that belong to five commonly appearing semantic categories in medical discourse. We believe that different feature sets may be appropriate for different semantic categories. The results from our experiment verify this hypothesis. In addition, we perform experiments to find the most salient features for each semantic category.

4.1 Task Definition

Coreference resolution is the identification of markables in text that refer to the same entity. In this thesis, we study coreference resolution in medical discharge summaries, construct a coreference resolution engine for this domain, and identify the most informative features for resolving coreferent markables in five semantic categories extracted by CaRE. These categories are: CONS, DIS, MED, SYMP, and TEST.

CaRE can identify three additional semantic categories which are not used in this thesis. Two of these categories, DOS and RESULT, are usually descriptive prepositional/verbal attributes of MED and TEST, respectively. Therefore, they are closely linked to the resolution of MED and TEST. They are also more relevant to event extraction rather than coreference resolution because most markables in these categories are standalone instances. Instances in the third omitted category, SUBS, occur rarely in the data set and do not show much variation in vocabulary or orthographic form.

As mentioned earlier, CaRE can also identify the presence-status of SYMP and DIS markables. SYMP and DIS markables can either be asserted to be present (pres), absent (abs), possibly present (poss), or present in someone other than the patient (some).

For the five semantic categories studied in this thesis, we consider two markables to corefer to each other if:

CONS – markables explicitly or implicitly refer to the same person, team, or organization

DIS – markables refer to the same occurrence of a disease (if a disease reappears, after it was cured, then the two occurrences are treated as different entities)

MED -
(Medication) markables refer to the same medication or same exact equipment
(Procedure) markables refer to the same occurrence of an invasive procedure or surgery

SYMP – markables refer to the same occurrence of a symptom (if a symptom reappears, after it was cured, then the two occurrences are treated as different entities).

TESTS - markables refer to the same occurrence of a laboratory or diagnostic test

Figure 5: General coreference resolution rules

We also define rules on resolving coreference amongst DIS and SYMP markables that refer to entities that are asserted to be absent or only possibly present. In these cases, we consider markable pairs that have the same presence-status and refer to the same named-entity to corefer to each other. For example,

"headache" in the sentences "the patient does not have a <dis-abs> headache </dis-abs> on 8/12" and "the patient does not have <dis-abs> headache </dis-abs> on 8/14" corefer to each other. However, "some" presence-status is a special case because it is a declaration of presence in someone other than the patient. It, therefore, should follow the conventions of "pres" markable pair classification.

In cases where markables that make up a markable pair have different presence-statuses, context is important in determining whether they corefer; it is not the case that all markables with differing presence-status are non-coreferent to each other. For example, consider the sentences below:

Sample sentences where coreferent markables do not have matching presence-status.
S1. Patient was checked for suspicion of <dis-poss> pneumonia </dis-poss>
S2. The patient was found to have <dis-pres> pneumonia </dis-pres>
S3. The <dis-abs> infectious disease </dis-abs> resolved on 12-10

These sentences contain markables that have different presence-status, however, they all refer to the same instance of "pneumonia".

In general, because diseases and symptoms can recur, it is important to distinguish between phrases announcing newly arisen disease/symptoms versus those announcing a change in status of existing disease/symptoms. External world knowledge on the usual duration of diseases and symptoms may help decipher whether markables are coreferent. For example, mentions of AIDS in "AIDS in 2006" and "AIDS in 2002" are referring to the same disease because once a patient has AIDS it cannot be cured. But in the case of "pneumonia in 2006" and "pneumonia in 2002", the markables are likely referring to different underlying entities because pneumonia lasts much less than a year.

The ambiguous cases from above illustrate that the interpretation and use of presence-status to resolve coreferent markables depends on the entity/event represented by the markable. Uncertainty arises due to varying durations amongst different diseases and symptoms. We take the conservative approach and assert that coreference relationships can only be drawn on markables that refer to the same exact entity/event.

4.2 Data

We perform our experiments over a collection of 47 hospital discharge summaries totaling 4978 lines of text. The records have been previously tokenized, sentence split, and semantically categorized. We annotate these records with coreference chains and time stamps.

As mentioned previously, hospital discharge summaries are semi-structured texts. In general, hospital discharge summaries are divided into sections that contain information vital to a patient's medical care. Not all records have a standard format that contains the same set of sections. Across medical records, doctors may also assign different names to the same section. The main sections that usually appear in all discharge summaries are listed below along with a description of the information that the sections contain.

Section	Description	Example
Past Medical History	Significant surgeries, illness that occurred in previous hospital visits	status post cerebrovascular accident

Allergies	Allergic reactions	peanut allergies
Social History	Family and social habits	The patient has two daughters and smokes a pack a day
History of Present Illness	Any history related to the patient's current hospital visit, e.g. events leading up to the hospital visit, relevant past medical history	The day prior to hospitalization, the patient was walking and slipped on the floor
Physical Examination	Results of the patient's physical examination	Head, eyes, ear, nose, and throat
Laboratory Results	Results of laboratory tests	Platelet count, Chem-7 test
Hospital Course	Actions taken by the Medical Practitioners during this hospital visit along with the results of the actions, patient's status and conditions are also updated	A chest x-ray was taken on 6/17
Discharge Medication	Medications that the patient is discharged with	Tylenol b.i.d.

Table 12: Sections of a hospital discharge summary

Within each section, information may be detailed in list, paragraph, or other structured formats. "Hospital Course" and "History of Present Illness" are almost always in paragraph form. They contain the narrative for the patient record; the other sections are often used mostly as data references.

4.3 Annotation

4.3.1 Coreference Resolution

We modify the MUC SGML coreference annotation scheme [34] for our annotations. Like MUC we enclose all markables with "<COREF>" and "</COREF>" tags. In MUC, the "ID" attribute of the <COREF> tag signifies the unique ID for a markable. While the "REF" attribute is used to denote the markable's antecedent. Instead of this method, we simply use ID to identify the coreference chain the markable belongs to, removing the need for the REF tag. The MUC coreference annotation scheme contains three additional attributes "TYPE", "MIN", and "STATUS"; these attributes are not useful for our purposes and therefore we omit them from our annotations. "TYPE" details the relationship between the antecedent and the anaphor (e.g., identity, possessive, etc.). In coreference resolution only the identity relationship can exist, making this attribute unnecessary. The MUC coreference task requires systems to identify markables in addition to performing coreference resolution over the identified markables. The "MIN" attribute is used to denote the minimal string that needs to be enclosed within a system-predicted markable in order for the markable to be considered the same markable as the one in the answer key. In our data set, markables' boundaries are predetermined, removing any boundary ambiguities and the need for "MIN". The last attribute is "STATUS". It denotes any ambiguous/uncertain coreferent pairs so that during evaluation such cases can be included or excluded. In our annotations, ambiguous cases also occur, however, there aren't enough cases to warrant incorporating the feature.

Sample Annotations:

<COREF ID="1"> Chest x-ray </COREF> was taken on 2/14.

<COREF ID="1"> Chest x-ray </COREF> showed evidence of pneumonia.

Figure 6: Sample coreference annotations

We use a JAVA graphical user interface (GUI) to improve the coreference annotation experience. The GUI shows a block of 20 sentences to the annotator in the main text area. In a separate pane, the GUI displays all identified coreference chains within the current record. All markables that are part of the same chain are displayed in a single, selectable line as "mark1 -> mark2 -> mark3 ...". When a coreference chain is selected, the GUI highlights any markable in the current sentence block that is in the selected chain. The annotator can move to other sentence blocks by clicking the "Forward" or "Back" buttons. Switching sentence blocks while a coreference chain is selected will highlight any markables from the chain that exist in the new sentence block. The annotator can declare two chains coreferent by selecting both chains and clicking the "Link Existing Chains" button. Conversely, a markable can be removed from a chain by selecting the chain, clicking the highlighted markable, and then clicking the "Split Chain" button.

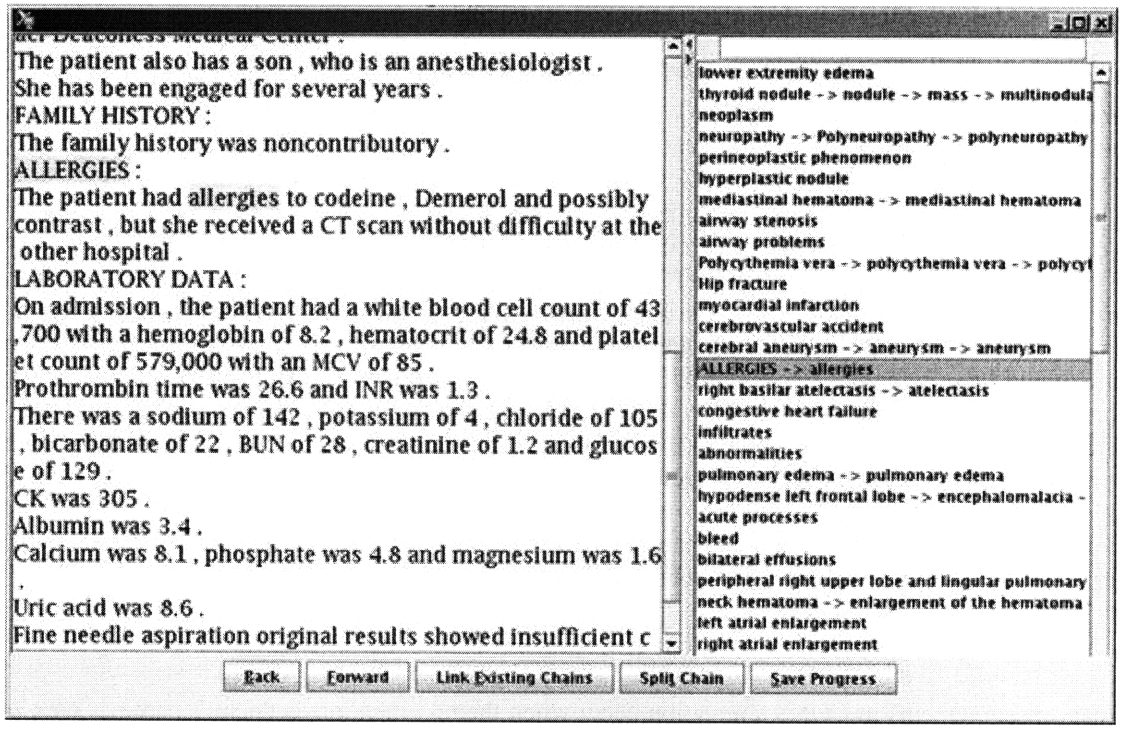

Figure 7: Screenshot of coreference annotation GUI

We also include two features that reduce the likelihood of annotation errors. A search box allows annotators to search for all coreference chains that contain a user inputted string. This feature helps to find chains with similarly spelled markables to ensure that coreference links are not missed. Also, the GUI can display all sentences that contain a markable from the selected coreference chain, allowing for easy context comparisons to ensure that the markables do, in fact, refer to each other. This feature was implemented by Sharon Zhang.

We find that annotating each semantic category separately reduces confusion and shortens the search time when looking for potentially coreferent markables in the text. Annotators, therefore, annotate a file five times, once for each semantic category. To speed up the annotation process, we first run a program that assigns all exact-match markables to the same coreference chain so that the annotator starts with some coreference chains already annotated.

We provide guidelines to two computer science students on what constitutes a coreferent markable. After

some supervision to ensure the clarity of guidelines, the students annotate the data separately. Upon completion, we evaluate the inter-annotator agreement between the students' annotations using the Kappa statistic [45,32].

We evaluate the Kappa statistic for each semantic category individually. Annotator agreement varies greatly depending on the semantic category. The annotators agree the most on CONS annotations and the least on TEST annotations. TEST coreference resolution requires more thorough reading and interpretation of the surrounding text due to there being many non-coreferent, exact-match markables. As a result, annotators are likely to make annotation errors. In general, TEST markables are more ambiguous.

	CONS	DIS	MED	SYMP	TEST	ALL
Kappa	0.9134	0.7829	0.8803	0.8293	0.7099	0.8363

Table 13: Coreference Kappa statistics for each semantic category

Annotator disagreements occur in three predominant forms. A large number of disagreements are due to judgment error made by one of the annotators. Judgment errors usually occur due to annotators overlooking a key piece of information (e.g., time information, co-occurring events, etc.) in the text surrounding the markables. The other major disagreement factor is the confusion caused by some hypernymy relationships. For example, an annotator would reason that an "eye examination" is part of HEENT and therefore coreferent to "HEENT" without thinking if the relationship is really equivalent. The remaining disagreements are due to differing interpretation of what makes two markables coreferent. For example, the annotators disagree on whether two diseases that have the same name and are asserted to be absent should be coreferent. Because MUC rules do not specify how to handle such situations, the annotators eventually set their own standard. Two markables with the same name that are asserted to be absent are coreferent to each other provided that contextual clues do not link one of the markables with another markable. Some other error factors include inherent ambiguity due to lack of context or ignorance of synonymy relationships by the annotators. In the next few paragraphs, we highlight some examples of annotation disagreements for each semantic category. Many of these examples highlight how ambiguities can often arise based on different interpretations and points of view (as mentioned in section 2.2).

CONS

The primary disagreements in CONS annotation occur when the text mentions a doctor's name as well as his/her medical group. From an interpretational standpoint, the two markables may be referencing the same concept, but strictly speaking the doctor is only a single member of the medical group. If there are two members in a medical group, then it would be incorrect to say that the medical group refers to any single member. Therefore, we believe such situations should not result in coreferent markable pairs.

Example:

The patient is to follow up with <cons> Dr. Ra Shuffburle </cons> in <cons> GI Clinic </cons> in one week , <cons> Dr. Telshey Patient </cons> in <cons> Cardiology Clinic </cons> in one week , <cons> Dr. _____ </cons> in Lifairg Louen Likison Hospital Medical Center in three weeks .

DIS

Similar to the previous example, DIS markables with similar meaning or multiple name variations also cause confusion. In the example below, the dictating doctor identifies "stroke" in several different ways. However, one of the annotators was unsure of the synonymous relation and marked the markables as being non-coreferent.

Example:

S1. She was brought to the <cons> emergency room </cons> where she was <test> evaluated </test> and determined to have had a <dis-pres> cerebrovascular accident </dis-pres> .

S2. Our leading theory at this point is that her <dis-pres> cerebellar dysfunction </dis-pres> relates to <dis-pres> perineoplastic syndrome </dis-pres> , relating to her history of <dis-pres> breast Ca </dis-pres> .

S3. <dis-pres> Small stroke </dis-pres> , nearly recovered , likely <dis-pres> embolic from carotid artery </dis-pres> .

A more ambiguous example is when a disease and any resulting medical problems related to it appear near each other. In the example below, annotators disagree on whether "adenocarcinoma" should be part of the coreference chain containing the tumor/mass markables. While it may be tempting to say that the tumor is cancerous and therefore coreferent to adenocarcinoma, this interpretation is incorrect. For example, what if the cancer spreads to other parts of the body? In such cases, there may be multiple tumors. The cancer then would refer to the collection of tumors in the body, rather than any single tumor.

Example:

S1. On 12/23 , she was seen by <cons> Dr. Stonge </cons> at Bri Health where a <dis-pres> 3 x 7 cm left mid clavicular mass </dis-pres> was noted .

S2. A <test> fine needle aspirate </test> of this <dis-pres> mass </dis-pres> reportedly revealed <dis-pres> giant cell tumor </dis-pres> .

S3. The patient was initially told that this was a <dis-pres> benign tumor </dis-pres> and that it could simply be watched and definitively treated after the delivery of her baby .

S4. The <test> pathology </test> , unfortunately , revealed an <dis-pres> aggressive adenocarcinoma (micropapillary type ; mucin producing) </dis-pres> .

SYMP

Unlike diseases, which have definitive diagnoses, symptoms are more ambiguous. By definition, symptoms can only be felt by the patient and therefore symptoms can change locations and presence-status quickly depending on the patient's report. Resolving SYMP markables, therefore, require careful interpretation and reading of the passage. In the sentences below, an abdominal pain gradually changed into back pain. The pain briefly receded and then came back in the form of "minimal back pain". Due to the close timing of the back pain to the abdominal pain and the abdominal pain having gradually shifted to back pain earlier, the annotators were unsure whether "some minimal back pain" is coreferent to the other markables. We decided the "back pain" should not be corefent because the "back pain" in S4 represents a new event, rather than a continuation of the old event.

Example:

S1. This is a 49 year-old male with a history of a <med> low anterior resection </med> in May of 1998 and a recurrence of <dis-pres> metastasis </dis-pres> with <med> asleeve of section of left colon diverting ileostomy </med> for recurrent <dis-pres> metastasis </dis-pres> later in May of 1999 who presents with <symps-pres> anterior midepigastric abdominal pain </symps-pres> .

S2. The patient states that the <symps-pres> pain </symps-pres> began at 10:30 on the day of admission with <symps-pres> increase </symps-pres> during the day of admission with <symps-pres> positive nausea </symps-pres> and <symps-pres> vomiting </symps-pres> , no <symps-abs> diarrhea </symps-abs> , <symps-pres> decreased output from the ostomy </symps-pres> , and the <symps-pres> pain was radiating to the back </symps-pres> .

S3. The <symps-pres> pain </symps-pres> improved quickly and the patient was started on a <med> clear liquid diet </med> which was advanced as tolerated .

S4. The patient had <symps-pres> some minimal back pain </symps-pres> that occurred after food but with <results> negative </results> <test> urinalysis </test> and negative <symps-abs> fever spikes </symps-abs> over the entire course of this stay .

TEST

The most common annotator disagreement for TEST markables is when one test is a sub-test of another. For example, HEENT incorporates a series of small examinations to check a patient's head, eyes, ears, nose, and throat. Even though each one of the examination procedures is part of the HEENT process, "HEENT" refers to the collection of exams rather than just any single exam. We apply this reasoning to the example below and argue that guaiac is only a part of the rectal test and therefore should not be coreferent to rectal.

Example:

<test> Rectal </test> was <test> guaiac </test> <results> negative </results> .

The example below demonstrates another TEST disagreement. The annotators disagree on whether an exercise test can corefer to a standard/protocol. We believe, in this case, that the protocol is coreferent to the exercise test because it is the name for a specific set of procedures for the test. In this context, the two markables can be used interchangeably.

Example:

S1. On 11-2-93 , the patient underwent a <test> low level treadmill exercise test with Thallium imaging </test> .

S2. The patient was able to <results> exercise for approximately nine minutes </results> on a modified <test> Bruce protocol </test> , however , he <results> did not reach his predicted maximal heart rate </results> while he was on the <med> beta blockade </med> .

MED

MED disagreements arise due to the presence of both events (treatment procedures, treatment names) and entities (medication). In the example below, one annotator marked "COUMADIN" as coreferring to "PROSETHETIC VALVE ANTICOAGULATION." However, Coumadin is the medication used to treat the patient, it does not refer to the process of anticoagulating the patient. In general, we find that events and entities cannot corefer to each other.

Example:

HIS <med> COUMADIN </med> WAS RESTARTED FOR <med> PROSTHETIC VALVE ANTICOAGULATION </med> WITH A GOAL <test> INR </test> OF <results> 1.5-2.0 </results> .

4.3.2 Time Stamp

The discharge summaries were also manually tagged for date information so that the system can include some temporal features. We assign each markable a time stamp based on the temporal reference frame under which it is discussed. In cases where there are dates in the vicinity of the markable that assign a more specific date of occurrence for an event markable, we defer to the more specific date rather than the time frame of the passage. The various forms of time stamps that can be assigned are detailed below:

Type	Annotation Format
Single day	[MM/DD/YYYY]
Multiple dates	[MM/DD/YYYY, MM/DD/YYYY,...]
Date Range	[MM/DD/YYYY-MM/DD/YYYY]
Month and Year	[MM/YYYY]
Year	[YYYY]
Unknown	[?]
Possible date	[?MM/DD/YYYY]
Up to a certain date	[-MM/DD/YYYY]

Starting a certain date	[MM/DD/YYYY-]

Table 14: Annotation formats for date information

To facilitate date annotation, we first run a date recognition program to find explicit dates from the passage. Then the annotators add in additional dates and check to ensure that the automatically tagged dates are correct. The date recognition program is fairly useful because nearly a quarter of the documents have some form of explicit date clues. However, the frequency of dates that appear in a record often depends on the narrator's style. There are many passages that do not have a clear date reference frame. For these cases, we ask the annotators to make a guess. Guessed dates are identified by a "?" symbol near the date. Date annotation is particularly time consuming due to the large amounts of guesswork and the many inferred dates that appear in the text.

4.4 Initial Evaluation

Preliminary Evaluation by Markable String-Match Type

	CONS			DIS			MED			SYMP			TEST		
	Count	% of All Mark.	% of All Coref Mark.	Count	% of All Mark.	% of All Coref Mark.	Count	% of All Mark.	% of All Coref Mark.	Count	% of All Mark.	% of All Coref Mark.	Count	% of All Mark.	% of All Coref Mark.
Partial Match															
Coreferent	38	3.8	49.4	303	1.1	34.6	140	0.3	14.5	90	1.0	29.7	53	0.1	19.7
Non-Coreferent	167	16.9		524	1.9		281	0.5		204	2.2		1127	2.1	
Exact Match															
Coreferent	32	3.2	41.6	440	1.6	50.2	749	1.5	77.5	151	1.6	49.8	198	0.4	73.6
Non-Coreferent	0	0.0		55	0.2		29	0.1		46	0.5		643	1.2	
No Match															
Coreferent	7	0.7	9.1	133	0.5	15.2	77	0.1	8.0	62	0.7	20.5	18	0.0	6.7
Non-Coreferent	746	75.4		25716	94.6		50202	97.5		8719	94.0		52685	96.3	
Total															
All	990	100	100	27171	100	100	51478	100	100	9272	100	100	54724	100	100

Table 15: Distribution of coreferent markable over different string-match types

We evaluate the difficulty of the coreference resolution by examining string-match type and edit-distance of markable pairs. The three types of string-match that are found in pairs of markables are exact-match, partial-match, and no-match (we will examine a fourth type substring-match later). A markable pair is an exact-match if the markables are exact copies of each other, while the pair is a partial-match if the two markables share at least one token in common. All markable pairs that do not share any tokens in common are of no-match type. Each semantic category contains a large proportion of coreferent exact-match pairs. More than 70% of TEST and MED markable pairs are exact-matches, while coreferent CONS, DIS, and SYMP markable pairs are exact-matches between 40-50% of the time. TEST and MED have extremely high proportions of exact-match coreferent pairs because markables belonging to these two semantic categories are usually single token words.

It is worth noting that simply having a large number of coreferent pairs of a string-match type does not mean that the string-match type is automatically a good indicator of coreference. It is also important to examine if a large number of non-coreferent pairs are also of the same string-match type. If so, a coreference system that uses the string-match type to predict coreference would suffer from precision

problems as it would misclassify many non-coreferent pairs as coreferent.

To examine the usefulness of each string-match type for coreference resolution, we use the pair-wise F-measure to evaluate system performance when all exact-match, partial-match, or no-match markables are classified as coreferent.

	CONS	DIS	MED	SYMP	TEST
RECALL					
Partial Match	.494	.346	.145	.297	.197
Exact Match	.416	.502	.775	.498	.736
No Match	.091	.152	.080	.205	.067
PRECISION					
Partial Match	.185	.366	.333	.306	.045
Exact Match	1.000	.889	.963	.766	.235
No Match	.009	.005	.002	.007	.000
F-MEASURE					
Partial Match	.270	.356	.202	.302	.073
Exact Match	.587	.642	.859	.604	.357
No Match	.017	.010	.003	.014	.001

Table 16: Pairwise F-measure evaluation of classifying all markables of different string-match types as coreferent

As expected, exact-match is a much better indicator of coreference than partial-match or no-match. However, its usefulness varies across semantic categories. Exact-match information is extremely useful for MED markables, while it is much less helpful towards coreference resolution in TEST markables. For the remaining categories, exact-match yields F-measures near .60.

From these preliminary evaluations, simply evaluating the string-match types of markables can yield fairly high performance measures for a majority of the categories. The lone exception is TEST, where prevalence of exact-match, non-coreferent markable pairs significantly reduces the power of using exact-match as a predictor of coreference. Though the high precision rates of exact-match CONS, DIS, and SYMP markable pairs suggest that exact-match is a good indicator of coreference, only half of the coreferent markables in these categories are exact-matches. The other half contains mostly partial-match pairs; however, assigning partial-match pairs as coreferent without knowing to what degree they partially match is imprecise. Additional features that examine the percentage token-match overlap between markable pairs may result in significant improvement on TEST, CONS, DIS, and SYMP resolution by eliminating false alarms on exact-match and partial-match pairs. We believe these features are also likely to significantly increase the system performance on no-match coreference resolution.

4.5 Semantic Coreference Resolver

We treat coreference resolution as a binary classification machine-learning problem in the spirit of McCarthy and Lehnert. Below, we detail each of the four steps of our algorithm: instance creation, feature set selection, machine learning algorithm, and output clustering.

4.5.1 Instance Creation:

Our instance creation method is akin to the one used by McCarthy and Lehnert [30]. We pair together all markables of the same semantic category within a record. A record with four disease markables (i, j, k, l) would contain six markable pairs ([i,j] [i,k] [i,l] [j,k] [j,l] [k,l]). A single record with n total markables

would result in $\binom{n-1}{2}$ markable pairs.

We choose the instance creation approach detailed above because it provides full information on the data set. As mentioned in the related works section, instance creation methods used by Ng and Cardie and Soon et al. do not include all possible coreferent markables, which means their method may lose some valuable training instances. Furthermore, there is no evidence that McCarthy and Lehnert's approach is worse than the other two approaches [38]. Ng's evaluation of McCarthy and Lehnert, Ng and Cardie, and Soon et al.'s approaches over three different data sets finds McCarthy and Lehnert's MUC F-measure to be higher than those of the other two approaches. When evaluating the system using B-CUBED F-measure, Ng finds each approach to be superior in one data set.

Another reason for choosing the McCarthy and Lehnert's approach is the small size of our data set. Even after pairing markables exhaustively, we are only able to produce 77 coreferent CONS pairs, 269 TEST pairs, and 303 SYMP pairs. Compared to the MUC competition data set, which contains around 1300 instance pairs, our data sets are relatively small [15]. We, therefore, need as much training data as possible.

	CONS	TEST	SYMP	DIS	MED
Total Markable Pairs	990	54724	9272	27171	51478
Coref	77	269	303	876	966
Non Coref	913	54455	8696	26295	50512

Table 17: Total number of markable pairs

4.5.2 Feature Set:

We devise 57 features for use with a decision tree classifier. These features are divided into seven different feature groups, based on how they attempt to resolve coreference. The groups are orthographic, semantic, grammatical, lexical, morphologic, temporal, and miscellaneous. Many of the features in these groups have different perspectives depending on their frame of reference, for example the token-match feature has four perspectives: token-match-anaph (anaphor-perspective), token-match-antec (antecedent-perspective), token-match-greedy (greedy-perspective), and token-match-stingy (stingy-perspective).

While other coreference resolution systems use a single-perspective approach that represent each feature with a single value (e.g., using the cosine-similarity metric to find how similar two strings are), we propose a multi-perspective approach for representing feature values. In our system, there are many features that evaluate to different values based on whether the evaluation is done by comparing the anaphor to the antecedent or vice versa. For example, when calculating the percentage of overlapping tokens that exist between two markables, should the percentage token-match be with respect to the number of tokens in the anaphor or the antecedent? Both evaluations, or perhaps a greedy-perspective (taking the maximum score out of the anaphor and antecedent-perspectives) or stingy-perspective (taking the minimum score between the anaphor and antecedent-perspectives) may be more informative. Intuitively, a greedy-perspective would increase the system's recall while decreasing its precision by classifying more coreferent pairs, while a stingy-perspective could potentially increase a system's precision and decrease its recall.

If our initial string-match based coreference resolution evaluation is any indication, features will impact different semantic categories in different ways. While the overall system performance will likely increase

with the inclusion of a feature, there might also be some decrease in recall or precision. By including different perspectives for a feature, we allow the system to optimize that feature's contribution to the system. These optimizations probably only slightly change a feature's contribution to the overall system performance, but we hope that over the entire feature set, the slight gains from all the features will compound into a significant gain.

A major drawback to the multi-perspective approach is that it dramatically increases the number of system features. Out of the 57 entries in the table below, only 28 entries are distinct features; the rest are perspectives to distinct features. Multi-perspective features are present in 5 out of the 7 feature groups, with the temporal and miscellaneous feature groups not containing any multi-perspective features.

	Feature And Feature Perspectives	Possible Value	Description
Orthographic Features			
1	token-match-anaph	[0,1]	% of i's tokens that match j's tokens
2	token-match-antec	[0,1]	% of j's tokens that match i's tokens
3	token-match-greedy	[0,1]	MAX(feature 1, feature 2)
4	token-match-stingy	[0,1]	MIN(feature 1, feature 2)
5	normalized-token-match-anaph	[0,1]	After Norm, % of i's tokens that match j's tokens
6	normalized-token-match-antec	[0,1]	After Norm, % of j's tokens that match i's tokens
7	normalized-token-match-greedy	[0,1]	MAX(feature 5, feature 6)
8	normalized-token-match-stingy	[0,1]	MIN(feature 5, feature 6)
9	edit-distance	0,1,2...	Levenstein Distance
10	normalized-edit-distance	0,1,2...	After Norm, Levenstein Distance
Semantic Features (MetaMap)			
11	umls-concept-match-anaph	[0,1]	% of i's assigned UMLS concepts that matches j's UMLS concepts
12	umls-concept-match-antec	[0,1]	% of j's assigned UMLS concepts that matches i's UMLS concepts
13	umls-concept-match-greedy	[0,1]	MAX(feature 12, feature 11)
14	umls-concept-match-stingy	[0,1]	MIN(feature 12, feature 11)
15	umls-concept-token-match-anaph	[0,1]	% of i's assigned UMLS concepts that matches j's UMLS concepts by words
16	umls-concept-token-match-antec	[0,1]	% of j's assigned UMLS concepts that matches i's UMLS concepts by words
17	umls-concept-token-match-greedy	[0,1]	MAX(feature 16, feature 15)
18	umls-concept-token-match-stingy	[0,1]	MIN(feature 16, feature 15)
19	umls-type-match-anaph	[0,1]	% of i's UMLS Semantic Types assignment that matches j's assignment
20	umls-type-match-antec	[0,1]	% of j's UMLS Semantic Types assignment that matches i's assignment
21	umls-type-match-greedy	[0,1]	MAX(feature 19, feature 20)
22	umls-type-match-stingy	[0,1]	MIN(feature 19, feature 20)
23	umls-type-anaph	Selected UMLS Semantic Types	The semantic type for the antecedent

Lexical Features			
24	sentence-token-match-anaph	[0,1] or Unknown	% of tokens in i's sentence that matches tokens in j's sentence
25	sentence-token-match-antec	[0,1] or Unknown	% of tokens in j's sentence that matches tokens in i's sentence
26	sentence-token-match-greedy	[0,1] or Unknown	MAX(feature 24, feature 25)
27	sentence-token-match-stingy	[0,1] or Unknown	MIN(feature 24, feature 25)
28	sentence-stop-words-removed-token-match-anaph	[0,1] or Unknown	After stop-words are removed, % of tokens in i's sentence that matches tokens in j's sentence
29	sentence-stop-words-removed-token-match-antec	[0,1] or Unknown	After stop-words are removed, % of tokens in j's sentence that matches tokens in i's sentence
30	sentence-stop-words-removed-token-match-greedy	[0,1] or Unknown	MAX(feature 30, feature 31)
31	sentence-stop-words-removed-token-match-stingy	[0,1] or Unknown	MIN(feature 30, feature 31)
32	sentence-markable-all-category-match-greedy	[0,1] or Unknown	% of other markables that match each other in i and j's sentence
33	left-markable-all-category-greedy	True / False / Unknown	% of tokens that match in i and j's left neighboring markable
34	left-markable-all-category-stingy	True / False / Unknown	% of tokens that match in i and j's left neighboring markable
35	right-markable-all-category-greedy	True / False / Unknown	% of tokens that match in i and j's right neighboring markable
36	right-markable-all-category-stingy	True / False / Unknown	% of tokens that match in i and j's right neighboring markable
Syntactic Features			
37	noun-match-anaph	[0,1]	% of i's noun tokens that match j's noun tokens
38	noun-match-antec	[0,1]	% of j's noun tokens that match i's noun tokens
39	noun-match-greedy	[0,1]	MAX(feature 39, feature 40)
40	noun-match-stingy	[0,1]	MIN(feature 39, feature 40)
41	plurality-match	True / False / Unknown	Do i and j match in number?
Morphological Features			
42	prefix-match-anaph	[0,1]	what % of i's tokens match j's tokens, when only considering the first 4 letters of each token
43	prefix-match-antec	[0,1]	what % of j's tokens match i's tokens, when only considering the first 4 letters of each token
44	prefix-match-greedy	[0,1]	MAX(feature 42, feature 43)
45	prefix-match-stingy	[0,1]	MIN(feature 42, feature 43)
46	last-name-match	True / False / Unknown	locate a last name, does it match?
Temporal Features			
47	date-default-certain	True / False / Unknown	only match certain dates, all pairs with uncertain dates result in unknown value

48	date-default-unknown	True / False / Unknown	match certain dates with certain dates, uncertain dates with uncertain dates, all other matches are unknown
49	date-default-false	True / False / Unknown	match certain dates with certain dates, uncertain dates with uncertain dates, all other matches are false
50	date-ambig	True / False / Unknown	allow certain dates to be matched with uncertain dates
Miscellaneous Features			
51	word-distance	0,1,2,…	the number of words i and j are away from each other
52	entity-distance-all-category	0,1,2,…	the number of total markables from any sem class i and j are away from each other
53	entity-distance	0,1,2,…	the number of markables of the same category I and j are away from each other
54	sentence-distance	0,1,2,…	the number of sentences i and j are away from each other
55	section-distance	0,1,2,…	the number of sections i and j are away from each other
56	section-type-match	all pair-wise combinations of section types -- past, presenthistory, present, discharge, none	a-b, a represent's i's section type, b represents j's section type.
57	presence	present, absent, possible, some, no-match, unknown	If both markables agree in presence, then they are assigned one of the first four values, if they do not agree then the result is a no-match, all categories other than DIS or SYMP will have unknown value for this feature

Table 18: Features set

Orthographic

While there are ten entries in the orthographic section of the table above, there are only four distinct orthographic features: token-match, normalized-token-match, edit-distance, and normalized-edit-distance. We will examine the token-match feature perspectives, i.e., token-match-anaph, token-match-antec, token-match-greedy, token-match-stingy, in detail. Assume markable i represents the antecedent markable and j represents the anaphor markable. To find the value for each perspective of token-match, we first find t*, the number of tokens that appear in both i and j. Token-match-anaph can be found by dividing t* by the total number of tokens in j. Token-match-antec can be found by dividing t* by the total number of tokens in i. Token-match-greedy is the maximum of token-match-antec and token-match-anaph, while token-match-stingy is the minimum of the two approaches. The same approach is used to find the anaphor-perspective, antecedent-perspective, greedy-perspective, and stingy-perspective for the other multi-perspective features in the feature set.

Token i: Chest X-ray	Token j: Left Chest X-ray
Token-match-anaph: 2/3	Token-match-antec: 2/2
Token-match-stingy: 2/3	Token-match-greedy: 2/2

Of the three remaining orthographic features, edit-distance is the Levenstein edit-distance for the two markables. This calculation is symmetric and therefore the multi-perspective approach isn't used. Normalized-edit-distance and normalized-token-match are the token-match scores of the markables when UMLS norm is first used to normalize the markables (see Resources 3.1 for how UMLS norm normalizes strings).

Semantic

We evaluate the semantic similarity of two markables in four ways: umls-concept-match, umls-concept-token-match, umls-type-anaph, and umls-type-match. These semantic features are derived from the output of MetaMap. As mentioned in Resources 3.1, MetaMap can match a name (in our case, a token or sequence of tokens in a markable) to several UMLS concepts. To reduce the number of matching UMLS concepts, we employ UMLS semantic type constraints and choose the top scoring UMLS concepts that have UMLS semantic types which are compatible with the semantic category of the markable in question (see Table 19). We thus use both the UMLS score as detailed by Aronson [2] and UMLS semantic type constraints to limit the number of UMLS concepts matched to any markable.

Semantic Category	UMLS Semantic Type
Disease	Pathologic Functions, Disease or Syndrome, Mental or Behavioral Dysfunction, Cell or Molecular Dysfunction, Congenital Abnormality, Acquired Abnormality, Injury or Poisoning, Anatomic Abnormality, Neoplastic Process, and Virus/Bacterium.
Treatment	Therapeutic or Preventive Procedure, Medical Device, Steroid, Pharmacologic Substance, Biomedical or Dental Material, Antibiotic, Clinical Drug, and Drug Delivery Device.
Practitioner	Biomedical Occupation or Discipline, and Professional or Occupational Group.
Test	Laboratory Procedure, Diagnostic Procedure, Clinical Attribute, and Organism Attribute.
Symptom	Sign or Symptom.

Table 19: UMLS semantic type mappings to CaRE semantic category [46]

We use two heuristics to measure semantic similarity between two markables: semantic similarity based on the number of matching UMLS concepts (umls-concept-match) and the number of matching tokens between the markables' UMLS concepts (umls-concept-token-match). While matching tokens of concepts may seem aggressive, the example below illustrates why it might be useful.

Example:

Markable: proximal femoral shaft fracture
UMLS concept: Shoulder Fractures; Fracture of shaft of femur (disorder)

Markable: a left femur fracture
UMLS concept: Femoral Fractures

umls-concept-match: 0
umls-concept-token-match-antec: 1/8
umls-concept-token-match-anaph: 1/2

The umls-concept-match value for the two markables above is 0 because the UMLS concepts assigned to each markable are different. However, the umls-concept-token-match value is greater than zero because the concepts assigned to the two markables share a matching token ("Fractures"). umls-concept-token-match-antec is assigned a value of 1/8 because there are 8 tokens for the two concepts assigned to the antecedent markable "proximal femoral shaft fracture" and one of these tokens ("Fractures") appears in the concept assigned to "a left femur fracture". Similar logic yields an umls-concept-token-match-anaph of 1/2. There are clear improvements that can be made to the exact-match approach to finding similarity between concepts. For example, normalizing the MetaMap output would be helpful because "Fracture" and "Fractures" would match to each other. Some other improvements include removing stop-words and normalizing the tokens. The most promising tool for determining interconcept relationships is likely the UMLS Semantic Network (as mentioned in section 3.1). These improvements, however, remain unexplored in this thesis.

We also include an anaphor markable's UMLS semantic type as a feature (umls-semantic-type-anaph) because grouping markables into smaller more homogeneous sets may improve the performance of the decision tree algorithm. In particular, our MED category contains medicines and operational events which may be distinguished by their semantic types.

Lastly, we include the umls-type-match feature to separate markables into even smaller groups. Umls-type-match is equal to 0 when two markables have different UMLS semantic types and 1 when two markables have the exact same UMLS semantic types. We believe this feature may improve the precision of our output by separating markables pairs into even more homogeneous groups than umls-concept-match.

Lexical Features

Up until now, the features that have been introduced are mostly markable-to-markable features that compare the characteristics of the antecedent markable to the characteristics of the anaphor markable. However, both markables' surrounding context also offer clues about their similarities. We include lexical features in the hope of capturing additional contextual clues that markable-to-markable comparisons cannot provide. These features are especially important for deciphering ambiguous nouns that refer to different underlying entities [39,3].

We perform lexical analysis on two window sizes: at the sentence level and at the neighboring markable level. The sentence-token-match feature examines the number of shared tokens that exist between the two sentences that contain i and j. If two markables are in the same sentence, their value for this feature is unknown. Sentence-stop-words-removed-token-match only takes into account overlapping words that do not appear in a stop-list of common prepositions, articles, numbers, and other words. We argue that the

higher the overlap in tokens of sentences, the higher the contextual agreement.

In addition to examining token overlap between two sentences, we also examine markable overlap. Intuitively, two coreferent markables will likely be surrounded by similar markables. Unlike tokens, markables are rarer and therefore overlapping markables between sentences is a more reliable indicator of contextual agreement. We also examine how much markables' left and right neighbor markables match each other (left-markable-all and right-markable-all); matching markables that are in close proximity to the markables in question might be a better indication of coreference due to the smaller window size. The table below reveals the weakness of token-matching sentences. The pitfall for using token-match to find sentence content match is that if two sentences contain many common tokens such as "the" or "a" or other uninformative tokens, then it could still result in a high sentence-token-match or sentence-stop-words-removed-token-match score.

Example:

S1. <test> x-ray </test> taken at Bayside Hospital revealed chance of <dis-poss> upper respiratory infection </dis-poss>

S2. <test> Chest x-ray </test> taken at Bower hospital revealed <dis-pres> upper lobe collapse <dis-pres>

Overlapping tokens: taken, at, hospital, revealed, upper
sentence-token-match-antec: 5/10
sentence-stop-words-removed-token-match-anaph: 4/8
sentence-markable-all-category-greedy: 0/1
right-markable-all-category-greedy: 1/3

Syntactic and Morphological Features

Syntactic and morphological features are introduced into the feature set to complement the orthographic features. Orthographic features that rely on token-match to determine string similarity are not be able to recognize markables that are derivational, inflectional, or morphological variants of one another. While using UMLS norm removes inflectional variants, it is sometimes too restrictive because it only removes suffixes and does not attempt to map roots together. For example "operate" and "operation", would result in different normalized forms ("operate" and "operat", respectively). We introduce features to locate markables that are derivationally or morphologically related to each other by introducing the prefix-match feature. This feature considers a markable pair matching if the markables overlap in their first four letters. There are many caveats and faults to this matching method. In particular, prefixes can cause both false alarms and false negatives. For example, "understand" and "underhand" would match, while "true" and "truth" would not. A better alternative would be to expand UMLS norm's composition of LVG funtions to include derivational variant matching; however, this is to be explored in future research.

We also include a feature to specifically help CONS coreference resolution. Many CONS markables include names of people; we try to extract the last name from markables and examine if the last names from two markables are the same. For this, we use last-name-match. The last names are extracted using a small set of REGEX expressions. We employ this feature as a more precise predictor of CONS coreference than token-match.

Another method for comparing similarity between markables is to only match their noun tokens. Introducing this feature allows "chest pain" and "sharp chest pain" to be recognized as a noun-match.

While traditional methods only match head nouns rather than all nouns, we believe that, for our purposes, matching all nouns is useful. This is because many markables in our corpus include nouns referring to body parts, e.g., "chest" in "chest pain". Markables referring to different parts of the body often are not coreferent, regardless of whether or not the head noun is the same. For example, "chest pain" is not the same as "knee pain."

In general, coreferent markables have to agree in number; therefore, we also use plurality-match to determine if two markables refer to the same entity or to different entities.

Temporal Features

We use temporal features to disambiguate commonly appearing event mentions, e.g., "x-rays". To obtain temporal information from text, annotators manually assign dates to each markable. While the narratives provide explicit dates for some markables, annotators have to guess dates for other markables. The two markables that make up a markable pair can both have explicit dates (type 1 markable pairs); both have guessed dates (type 2 markable pairs); or one markable can have an explicit date while the other can have a guessed date (type 3 markable pairs). All four temporal features in the feature set evaluate temporal similarity by looking for exact-date match; however, they treat the previously mentioned three types of markable pairs in different manners.

Date-default-certain only considers type 1 markable pairs, all other pairs are assigned a value of false. Date-default-unknown and date-default-false both evaluate type 1 and type 2 markable pairs. The difference between date-default-unknown and date-default-false is how they treat type 3 markable pairs during training. Date-default-unknown assigns type 3 markable pairs a value of "unknown", while date-default-unknown assign these markable pairs a value of "false." Lastly, date-default-ambig evaluate all three types of markable pairs. These distinctions are necessary because we are unsure of the quality of the annotators' guessed dates. If the quality of the guesses is bad, then only explicit dates may be useful; however, if the quality of the guesses is good, then they can be used for coreference resolution of markable pairs as well.

Miscellaneous

To complement the above features, we include features that examine the distance between markables and add to this group various corpora based features. The distance between markables is evaluated by counting sentences, words, discharge summary sections, markables that are of the same semantic category as the markable pairs, and markables in all semantic categories. The word-distance feature, for example, would count the number of words between the two markables in the markable pair. The machine learner would then potentially find several word-distance thresholds to classify pairs in different ways. Perhaps markable pairs that have large word-distance values would require higher token-match values than those markable pairs that have smaller word-distance values because coreferent markables that are far apart would need to refer to each other in explicit terms to cue the reader that they corefer. Because for DIS and SYMP markables, CaRE identified whether markables were asserted to be present, possibly present, or absent, we take advantage of this feature by examining whether markables refer to entities with same presence-status. Another piece of information that may improve prediction accuracy is the section that markables belong to. For example, a surgery mentioned in the PAST MEDICAL HISTORY section is more likely to be referred to in the HISTORY OF PRESENT ILLNESS section than the HOSPITAL COURSE section because both PAST MEDICAL HISTORY and HISTORY OF PRESENT ILLNESS contain past information, where as HOSPITAL COURSE describes mainly events that take place during the patient's current visit. Therefore, we use sections to approximate the time frame of each mentioned markable. We group sections into four groups: past, past-present, present, and discharge. These four

groups help us differentiate between the approximate time ranges of markables.

4.5.3 Machine Learner

We use the C4.5 decision tree algorithm to train our prediction model. We select the algorithm for its flexibility, prediction model readability, and proven track record [30,36,48].

Our feature set contains both numeric and categorical features. The C4.5 algorithm is flexible enough to classify both types of features. Furthermore, our system contains a variety features that serve different purposes. Some features (stand-alone features) directly locate coreferent markable pairs, while others (complement features) improve the precision and recall of other features. For each semantic category, different feature variants may perform better. For example, normalized tokens are useful for removing irregularities. Normalized-token-match therefore may help resolve TEST, MED, DIS, and SYMP coreference, but normalized-token-match can degrade CONS coreference because it may distort names that seem to be plural nouns. On the other hand, token-match is likely more appropriate for CONS because it does not normalize each markable's tokens before comparing them.

While the algorithm's run time is fairly slow, the C4.5 decision tree is able to deal with our relatively large and diverse feature set. Furthermore, the C4.5 prediction models are simply a set of "if-then" rules that are easy to interpret. These prediction models can provide an easy explanation for why the system makes classification mistakes during testing. Many past research efforts in coreference resolution have used the decision tree as the learning algorithm of choice because of its flexibility and readability [30,48].

Decisions trees however do have their disadvantages. The generated tree is not always stable, especially when the dataset is small. Our data set is small and dividing it into training and test sets would further decrease the data size. We therefore opted to employ 10 fold cross validation so that all the data can be used for training and testing.

Another problem with the C4.5 algorithm is that it only finds the optimal solution if its features are independent of each other. However, our features are not independent of each other. This problem is common in NLP because of the dependencies that naturally exist in language. However, there is no existing algorithm that can find the optimal solution other than exhaustive search. The only feasible alternative is to use an algorithm such as C4.5 that can often get close to the optimal solution but with no absolute guarantees.

In general, the advantages C4.5 outweigh its disadvantages. We employ this classifier particularly because the results and the nature of the decision tree models should provide us with valuable insights while allowing us to compare our results with those of past research efforts.

For our system, we run C4.5 with a confidence factor to .25 and minimum leaf size of 4 The confidence factor is used during pruning to remove branches that contribute minimally to the accuracy of the model. The minimum leaf size prevents the machine learning algorithm from over-fitting data by constraining the smallest number of instances that must be contained in a partition.

4.5.4 Clustering Algorithm:

We apply a clustering algorithm after classification. Our C4.5 model is trained to predict whether two markables are coreferent based on their feature vectors. Because this algorithm performs only pair-wise predictions, its output isn't guaranteed to satisfy the transitive property and may contain contradictions. For example, if the system predicts i to be coreferent to j, j to be coreferent to k, then it is implied that i is

coreferent to k, but it is possible for C4.5 to predict i to be non-coreferent to k.

Our clustering algorithm tries to eliminate these contradictions by assigning markables into coreference chains. It executes the "aggressive-merge" of McCarthy and Lehnert. This clustering approach assumes implied coreference links to be actual coreference links. Other clustering algorithms such as "closest-first" [51] and "best-first" [36] also exist. These methods, however, have been shown to be inferior to the McCarthy and Lehnert instance creation method [38]. For more detail on each algorithm refer to the related works section.

4.6 Evaluation

We run several experiments to evaluate our system. The experiments do not only evaluate system performance against a baseline, but also examine how individual features affect system performance for different semantic categories. Specifically, we examine how each feature's direct effect (a feature's individual contribution to the system) and complementary effect (a feature's ability to enhance other features) influence overall system performance.

Because the data set contains relatively few positive training instances, we use cross-validation to estimate system's performance over the entire data set. In n-fold cross-validation, the entire data set is randomly divided into n partitions. Each partition is used as the test set once. When one of the partitions is used as the test set, the other n-1 partitions are used as the training set. By the end of n cross-validation runs, the system has a prediction for each data point. These predictions are used as the system's output.

When evaluating overall system performance in terms of B-CUBED and MUC evaluations, we run the clustering algorithm. However, we do not run the clustering algorithm when evaluating the impact of individual features due to system and time constraints. Instead, we run a pair-wise F-measure directly on the output of the machine learner. We will examine the validity of this approach in the "complementary effect" section of this thesis.

4.6.1 Overall System Performance

Evaluation Method

We begin with an evaluation of the C4.5 system used with all of our features, i.e., the all-feature system. We evaluate the all-feature system performance using the MUC and B-CUBED metrics. We also examine how the system's performance differs over markables of different string-match type.

For the overall system evaluation, we run our all-feature system using 10-fold cross-validation. We then perform aggressive-merge clustering to triangulate any inferred coreference links from the system predictions. The system output is evaluated using the MUC and B-CUBED evaluation metrics. We compare the results against the performance of a baseline decision tree system that contains two features previously used in the literature to resolve coreference, i.e., token-match-antec [61] and plurality-match [30]. This baseline system can correctly classify many markable pairs. As indicated by our initial evaluation, many coreference pairs are either exact-match or partial-match pairs. By including the token-match-antec feature, the baseline system can distinguish between each of the three string-match types and perform resolution as is appropriate. With this capability, the baseline system can classify most of the easy string-match related resolutions. The plurality feature strengthens token-match-antec by preventing a match between markable pairs that do not agree in number.

We are unable to perform significance testing on the overall system evaluations because our experimental setup does not allow it. We only get a full sense of our system's performance level if all four steps of our algorithm are run sequentially and then the prediction outputs are evaluated against the gold standard. However because Weka's cross-validation procedure randomly divides the data set into n-folds, it breaks up some coreference chains and destroys transitive closure as not all markable pairs in the same coreference chain will necessarily be placed in the same cross-validation run. We, therefore, cannot use each validation run's performance as an indicator of overall system performance. As a result, we must aggregate the system predictions at the very end, perform clustering, and evaluate the system predictions in one final step.

However, in order to perform significance testing, we need more than a single data point. If we can evaluate the MUC and B-CUBED system performance for each individual cross-validation run, then we would have 10 data points for our Wilcoxon test. However, because both MUC and B-CUBED evaluations are based on evaluating overlaps between prediction chains and gold-standard chains, we would need to create 10 different gold-standards for each cross validation run. The creation of these gold-standards is non-trivial because if we only include the markable pairs that appear in each validation run, then transitive closure is not preserved and MUC and B-CUBED evaluations cannot be performed. Including all markable pairs would be incorrect because we would be trying to evaluate the system's prediction on 1/10 of the data points and comparing it against the wrong answer key that contains many markables that the system was not responsible for predicting. So even if we are willing to accept the system performance for each run as an estimate, we have no means of evaluating the system performance at each run because we are unsure what the gold standard for each run should be. We, therefore, reserve significance testing for when system evaluations are done using pair-wise F-measure because it does not require examining full coreference chains.

We do note that while it is impossible to perform significance testing using the current setup, it is possible to perform significance testing if we manually select the data used for each cross-validation run. Each cross-validation run can test one full hospital discharge summary record. We would therefore, have 47 cross-validation runs, rather than 50. This setup would effectively resolve the transitive closure breakdown because each discharge summary is self contained and the system can make predictions on all possible pairings within the discharge summary. The clustering algorithm can then be run for each validation run and the prediction can be compared with the annotations for that record. We do not perform this experiment in this thesis, though a follow-up experiment is underway.

Data Analysis

	CONS	DIS	MED	SYMP	TEST
Precision					
MUC (all-feature)	.9348	.9076	.9361	.8307	.7297
MUC (baseline)	.9730	.8686	.9397	.8140	.0000
B-CUBED (all-feature)	.9682	.9355	.9605	.9268	.9832
B-CUBED (baseline)	.9920	.9351	.9661	.9370	1.0000
Recall					
MUC (all-feature)	.7818	.8912	.9638	.8533	.3876
MUC (baseline)	.6545	.8095	.8964	.7609	.0000
B-CUBED (all-feature)	.9473	.9486	.9857	.9618	.9293
B-CUBED (baseline)	.9160	.9111	.9557	.9282	.8900
F-measure					
MUC (all-feature)	.8515	.8993	.9498	.8418	.5063

MUC (baseline)	.7826	.8380	.9175	.7865	.0000
B-CUBED (all-feature)	.9576	.9420	.9730	.9440	.9555
B-CUBED (baseline)	.9525	.9229	.9608	.9326	.9418

Table 20: All-feature and baseline system MUC and B-CUBED F-measures

	CONS	DIS	MED	SYMP	TEST
F-measure					
MUC	0.069	0.061	0.032	0.055	0.506
B-CUBED	0.005	0.019	0.012	0.011	0.014

Table 21: Difference between all-feature and baseline system

In general, we have constructed a high precision system. For example, even the extremely tough TEST markables, where 76.5% of the exact-match markable pairs are non-coreferent, achieve a MUC precision of .73 and a B-CUBED precision of .983. The all-feature system performs better than the baseline in all categories regardless of which type of evaluation metric is employed. However, the MUC improvements are larger than those of B-CUBED. This phenomenon occurs because B-CUBED includes the correct prediction of standalone markables as part of its score where as MUC does not. There is a fairly large number of standalone markables in the corpus so the B-CUBED statistics is automatically higher than the MUC evaluations from the start, and therefore there is less room for improvement of B-CUBED compared to MUC. In other words, as long as the system continues to correctly predict standalone markables at a high rate, improvements in how the system predicts coreferent markable pairs would only slightly affect the B-CUBED F-measures.

In terms of precision and recall, the addition of more features to the baseline results in large increases over the baseline's recall, while the effect on the baseline's precision is mixed. MUC and B-CUBED precisions decrease for CONS and MED resolution, while they slightly increase for DIS resolution. In the other two remaining semantic categories, SYMP and TEST, MUC precision is improved and B-CUBED precision is worse. We attribute this contradiction in the MUC and B-CUBED results to the difference in how the algorithm penalizes wrongly assigned links. B-CUBED penalizes markables that are wrongly linked to longer coreference chains; MUC does not. A decrease in B-CUBED score and an increase in MUC score indicates that the percentage of correctly predicted pairs rose from the baseline; however, the incorrectly predicted links are now connecting longer coreference chains, resulting in a heavier penalty to B-CUBED precision.

In both the MUC and B-CUBED evaluations, TEST markable resolution improved more drastically than most other categories. TEST resolution improved the second most in B-CUBED evaluations and the most in MUC evaluations. In categories other than TEST, MUC increased by .04-.06, while B-CUBED increased by .006 - .019. The large increase is a strong indication that the all-feature system goes beyond simple string-match to identify coreferent markables.

As expected, CONS markables experienced the least amount of system improvements (B-CUBED .006 / MUC .069). While coreference resolution of other categories' markables require contextual clues to disambiguate same nouns referring to different entities, CONS markables rarely have this problem. There are relatively few doctor names mentioned in each hospital record; therefore, chances of different doctors having the same name are fairly small, making string-match an extremely strong indicator of coreference in CONS.

Because exact-match markable pairs are easy to resolve, we extend our analysis to examine how our

system performs over three different string-match groups, i.e. no-match, partial-match, and exact-match. In order to evaluate B-CUBED and MUC F-measures to evaluate how well the system does on each string-match type, we need to split our gold standard into three different answer keys. Each answer key would only contain coreference chains consisting of exact-match, no-match, or string-match markables.

	CONS			DIS			MED			SYMP			TEST		
Feature	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F
No-match	.143	.143	.143	.638	.384	.479	.557	.701	.621	.378	.677	.486	.000	.000	.000
No-match (baseline)	1.00	.143	.250	.571	.530	.571	.308	.104	.155	1.00	.161	.317	.000	.000	.000
Partial-match	.675	.711	.692	.674	.901	.772	.564	.921	.699	.524	.856	.650	.727	.302	.427
Partial-match (baseline)	.895	.447	.597	.601	.678	.643	.461	.597	.520	.504	.700	.586	000	.000	.000
Exact-match	1.00	1.00	1.00	.895	.989	.9340	.963	1.00	.981	.851	.987	.914	.741	.404	.523
Exact-match (baseline)	1.00	1.00	1.00	.889	1.00	.941	.963	1.00	.981	.767	1.00	.868	.000	.000	.000
All	.756	.779	.769	.782	.866	.821	.845	.965	.901	.619	.870	.723	.739	.357	.481
All (baseline)	.962	.649	.775	.774	.742	.758	.854	.871	.862	.666	.698	.682	.000	.000	.000

Table 22: F-measure evaluation of all-feature vs. baseline system by string-match type

By breaking down the comparisons into different string-match types, we show that our system improves coreference resolution on partial-match and no-match markable pairs. In DIS, MED, and SYMP markables, the no-match resolution F-measures increased from the single digits to around 50-60%, while in every semantic category other than TEST, partial-match coreference resolution performance also improved more than exact-match resolution. There appears to be no improvements in TEST no-match resolution and a decrease in performance for classifying CONS no-match markables. We believe these are largely due to the lack of training data for these two categories, i.e., there are only 7 and 18 CONS and TEST no-match pairs respectively, while DIS, MED, and SYMP contain 133, 77, and 62 no-match pairs.

The evaluations in this section show that our feature set greatly enhances the coreference resolution performance of the baseline system. Much of the performance improvements come from better resolution of partial-match and/or no-match markables because the token-match baseline system does an excellent job of resolving exact-match resolutions. We also find that the TEST and CONS markables are very different than the markables of the other three categories. The system does not improve TEST and CONS no-match resolution, probably because of the lack of no-match training data.

Having examined overall system performance and confirmed that the feature set improves on the performance of the baseline, we now examine what aspects of our system contribute to this improvement. There are two primary reasons for the improvement over the baseline. First, the multi-perspective approach allows the system to adapt how it uses each feature to the semantic category under evaluation. Second, some features like sentence-markable-all-category can identify coreferent relationships in manners other than string comparisons. We will analyze the multi-perspective approach and the power of features in the feature set in the section that follows.

4.6.2 Effect of the Multi-Perspectives Approach

Evaluation Method

We evaluate the effect of the multi-perspective approach by comparing the all-feature system with systems including four subsets of features. The four sets are: anaph features alone (anaphor-perspective), antec features alone (antecedent-perspective), greedy features alone (greedy-perspective), and stingy features alone (stingy-perspective). For each of these, the multi-perspective feature set is reduced to include only a single perspective.

Data Analysis

	CONS		DIS		MED		SYMP		TEST	
	MUC	B-CUBED	MUC	B-CUBED	MUC	B-CUBED	MUC	B-CUBED	MUC	B-CUBED
Anaphor-perspective	**.8545**	.9508	**.8977**	.9373	.9380	.9695	.8329	**.9415**	.4774	.9534
Antecedent-perspective	.8041	.9550	.8848	.9267	.9444	.9706	**.8462**	.9405	.4459	.9514
Greedy-perspective	.8283	.9612	.8904	.9360	.9463	**.9731**	.8378	.9410	**.5032**	**.9543**
Stingy-perspective	.8333	**.9615**	.8891	**.9393**	.9375	.9694	.8365	.9387	.4778	.9541
All-Feature Set/Multi-perspective	.8515	.9576	**.8993**	**.9420**	**.9498**	.9730	.8418	**.9440**	**.5063**	**.9555**

Table 23: Single perspective system MUC and B-CUBED evaluations
Note: Darker highlight, with bold letters is the best performing feature set for the particular evaluation metric within a semantic category. The lighter highlight is the second best performing feature set.

The all-feature system consistently performs at or above the level of each of the single-perspective systems. The all-feature system contains the best or second best MUC and B-CUBED scores in 9 out of 10 comparisons. In the one exception, CONS B-CUBED, the all-feature system is the third best performing system. The experiment data indicates that, in general, the all-feature system is preferred over the single-perspective systems.

We also explore if any of the four single perspectives is better than the others. We find no perspective that clearly dominates the others by F-measure. However, the greedy-perspective seems most preferable. It is never the worst performing perspective; it performs the best out of all single perspectives for MED and TEST markables; and its MUC and B-CUBED scores in other categories are also in the upper half of the single-perspective systems.

While the multi-perspective approach does improve system performance in most cases, how much of an improvement actually results? To find out, we examine the performance gain of the all-feature system over the average performance of the single-perspective systems [Table 23]. We also examine the largest difference between the all-feature system and any one of the single-perspective systems and specifically compare with the antecedent-perspective. The antecedent-perspective is chosen because Yang et al. employed it in their experiments.

	CONS		DIS		MED		SYMP		TEST	
	MUC	B-CUBED	MUC	B-CUBED	MUC	B-CUBED	MUC	B-CUBED	MUC	B-CUBED
Average Difference Between All-Feature System and Single-Perspective Systems	.0214	.0005	.0088	.0072	.0083	.0024	.0035	.0036	.0302	.0022
Largest Difference Between All-Feature System and Single-Perspective Feature Systems	.0474	.0068	.0145	.0153	.0123	.0036	.0089	.0053	.0604	.0041
Difference Between All-Feature System and Antecedent Perspective	.0474	.0026	.0145	.0153	.0054	.0024	.0044	.0035	.0604	.0041

Table 24: Difference between all-feature and single-perspective system

We are unable to obtain significance figures for the different systems because there is only one evaluation run on the data set. We cannot split the data set into smaller data sets, because it already has a limited number of markables. However, in the table above, we do show differences in system performance for each semantic category. The increase in system performance is modest. TEST and CONS coreference shows fairly large improvements in MUC measures, while DIS shows a large increase in B-CUBED measures.

The differences in performance of the all-feature system from the antecedent-perspective come from two sources: the 28 features feature set and the additional 29 perspectives for the multi-perspective features. We can make a rough estimate of the multi-perspective approach's contribution to the overall system improvements by examining the performance difference between the antecedent-perspective and the all-feature system [Table 25].

	CONS	DIS	MED	SYMP	TEST
F-MEASURE					
MUC	0.069	0.061	0.032	0.055	0.506
B-CUBED	0.005	0.019	0.012	0.011	0.014

Table 25: Difference between All-feature system and antecedent perspective

4.6.3 Feature Contributions

In examining the power of each feature for coreference resolution in a semantic category, we evaluate the feature's direct and complementary effects on the system. The direct effect is how much the feature alone contributes to the system performance. However, our features are not completely independent from each other so a feature's actual effect on the system is also dependent on its interactions with other features. We call the side-effect from a feature's interactions with other features its complementary effect.

Consider the role that date features have in coreference resolution. The system cannot perform coreference resolution with only date information, however if the system knows that two markables have the same name and occurred on the same date, then it is likely the two markables are coreferent. For example, on the same day a "Chest X-Ray" and a "CT Scan" can be done, but without knowing whether these two markables refer to the same concept, the temporal information is too vague to be of any help. However, if a string-match or umls-concept-match feature is also introduced, the temporal information is likely to improve the precision. For example, a TEST such as an "X-ray" often refers to several different "x-rays" in a discharge summary, but a patient gets only a small number of "X-rays" taken on a given day. As a result, temporal features have a weak direct effect but stronger complementary effect on system

performance.

Evaluation Method

Like the previous section, we evaluate our system on different combinations of feature sets. However, due to the large number of features, we only find the features that are statistically significantly different from each other. In order to perform significance testing, for each feature set, we perform 50-fold cross-validation rather than 10-fold cross validation; this provides more data points for the Wilcoxon Ranked Test. Furthermore, instead of the B-CUBED and MUC evaluation methods, we use precision and recall, as described in Section 3.1.1, by examining the correctness of the system's prediction for each markable pair. This approach is the most convenient, least-time-consuming way to achieve an estimation of system performance.

Given a system with the right output features, we could have evaluated the system using the B-CUBED and MUC metrics. Unfortunately, due to system constraints with Weka, preparing the output data for these evaluation metrics would have taken an immense amount of time, therefore this option was abandoned for a faster evaluation method. We decided to use the pair-wise F-measure evaluation.

Even though the pair-wise F-measure is only an approximation for MUC scores, we believe the approximation is highly correlated with actual MUC performance. Inherently, good pair-wise coreference evaluation results implies that the prediction set is very similar to the gold standard. Furthermore, a good pair-wise evaluation should result in good clustering output. There are likely cases where a few misplaced categorizations can result in bad clustering that places many unrelated markables into the same coreference chain. However, bad clustering only occurs if system predictions are not precise. In the previous section, we found evidence that our system is a precision-biased system. Therefore, we believe that the bad clustering effect is likely low.

Direct Effect Analysis

We test the direct effect of features by evaluating the performances of single feature systems and comparing the F-measures of individual single feature systems with the F-measure of the system using only the token-match feature (Table 18: features 1-4).

	CONS		DIS		MED		SYMP		TEST	
	F	Conf	F	Conf	F	Conf	F	Conf	F	Conf
date-match	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
edit-distance	.463	.00	.627	.00	.856	.00	.614	.92	.000	1.00
entity-distance	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
entity-distance-all	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
last-name-match	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
left-markable-all-category, right markable-all-category	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
normalized-edit-distance	.474	.00	.631	.00	.856	.00	.623	.52	.000	1.00
normalized-token-match	.688	.21	.776	0.1	.890	.34	.634	.06	.000	1.00
noun-match	.684	.26	.691	.00	.749	.00	.000	.00	.000	1.00
plurality-match	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
prefix-match	.690	.18	.751	.00	.897	.68	.700	.00	.000	1.00
presence	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
section-distance	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
section-type-match	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
sentence-token-match	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
sentence-distance	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
sentence-markable-all-category-match	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
sentence-stop-words-removed-token-match	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
UMLS-concept	.000	.00	.591	.00	.742	.00	.000	.00	.000	1.00
UMLS-type	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
UMLS-type-match	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
UMLS-concept-token	.000	.00	.639	.00	.742	.00	.000	.00	.000	1.00
word-distance	.000	.00	.000	.00	.000	.00	.000	.00	.000	1.00
token-match	.729	-	.790	-	.892	-	.610	-	.000	-

Table 26: Single feature system pair-wise F-measure evaluation
NOTE: Dark Highlight: F-measures significantly different from the token-match system

There are several striking results from our experiment. First and foremost, we notice that the number of features that directly affect coreference resolution varies over the different semantic categories. MED and DIS semantic categories had three and four features, respectively, that had a direct effect. On the low end, no feature by itself was able to help the system identify TEST coreference pairs. The dearth of useful features for TEST markables is another indication that resolution of TEST markables is more complex than other categories. We believe resolving TEST markables, which often are nominal representations that refer to multiple underlying events, require more context than other categories. No single feature captures enough context to make resolution of TEST markables possible.

In terms of feature analysis, token-match, normalized-token-match, edit-distance, normalized-edit-

distance are all orthographic features. The direct effects of token-match and normalized-token-match on system performance are roughly similar (no differences were statistically significant). The same can be said for edit-distance and normalized-edit-distance. These results indicate that normalizing tokens does not enhance system performance in a significant manner. Overall, the token-match feature highly influenced most semantic categories with the exception of TEST resolution.

We use the UMLS and prefix features to capture morphological variants. Prefix seems the more successful way for locating morphologically related markables. It performs significantly better than all other features for SYMP markable resolution because SYMP markables have more variation than markables of other categories. For example, "hypertension" in its adjective form "hypertensive" is often used to describe a patient's high blood pressure. Markables describing treatment procedures in MED also vary between noun and verb form, which likely explains why prefix performs slightly better than token-match in MED in pair-wise F-measure evaluation. We also observe a very strong correlation between the above table and table 20. When we rank all-feature system performance from highest to lowest by MUC F-measure (Table 20), we find the order to be MED, DIS, CONS, SYMP, TEST. This same order corresponds to how the categories would be ranked, if they are sorted by the number of direct effect features they have! This correlation could be an indication that the number of salient direct effect features is the primary driver influencing resolution performance.

Complementary Effect Analysis

We evaluate the complementary effect by removing a feature from the all-feature system and evaluating the significance of the system performance changes.

	CONS		DIS		MED		SYMP		TEST	
	F	Conf	F	Conf	F	Conf	F	Conf	F	Conf
date-match	.759	.18	.837	.60	.920	.07	.773	.50	.374	.00
edit-distance	.733	.11	.838	.84	.924	.21	.766	.73	.502	.46
entity-distance	.759	.18	.838	.92	.922	.40	.773	.24	.470	.97
entity-distance-all	.759	.18	.837	.68	.920	.23	.769	.58	.492	.94
last-name-match	.767	.79	.838	.80	.919	.05	.764	.98	.498	.64
left-markable-all-category, right markable-all-category	.759	.18	.837	.66	.919	.03	.768	.73	.431	.14
normalized-edit-distance	.759	.18	.835	.39	.921	.09	.763	.75	.498	.67
normalized-token-match	.757	.55	.820	.01	.920	.05	.773	.73	.496	.61
noun-match	.732	.22	.837	.46	.917	.02	.755	.55	.513	.43
plurality-match	.759	.18	.834	.25	.918	.03	.763	.99	.500	.57
prefix-match	.749	.40	.836	.49	.919	.06	.771	.90	.498	.64
presence	.759	.18	.830	.19	.919	.25	.730	.05	.498	.64
section-distance	.759	.18	.839	.86	.919	.05	.766	.88	.470	.36
section-type-match	.750	.11	.831	.13	.918	.07	.765	.85	.506	.42
sentence-token-match	.767	.79	.837	.75	.919	.05	.774	.51	.454	.05
sentence-distance	.755	.11	.837	.70	.921	.08	.769	.46	.501	.48
sentence-markable-all-category-match	.759	.18	.836	.52	.919	.05	.764	.98	.475	.55
sentence-stop-words-removed-token-match	.759	.18	.838	.79	.919	.03	.779	.31	.475	.30
UMLS-concept	.759	.18	.838	.74	.921	.07	.751	.23	.500	.57
UMLS-type	.759	.18	.837	.54	.921	.08	.760	.86	.476	.41
UMLS-type-match	.759	.18	.835	.30	.920	.05	.753	.60	.502	.61
UMLS-concept-token	.759	.18	.835	.37	.921	.11	.765	.95	.497	.70
word-distance	.759	.18	.822	.01	.919	.05	.762	.95	.443	.04
Baseline (All Features)	.770		.838		.930		.763		.492	

Table 27: Leave-one-feature-out system performance
NOTE: Dark Highlight: System performances significantly affected by feature removal.
Light Highlight: System performances strongly affected by feature removal. Significantly Affected: > .95 confidence Strongly Affected: > .90 confidence

Two semantic categories (CONS and SYMP) are not significantly affected by the removal of any single feature from the all-feature system, while system performance on MED markables is extremely sensitive to changes in the feature set. It is easiest to distinguish influential features for DIS and TEST resolution. For DIS, finding the word-distance between markables and normalizing the markables are crucial to system performance. For TEST markables, withholding date and word distance information significantly reduced the performance of the system.

Other semantic categories that were significantly effected by the removal of features from the all-feature set are DIS and TEST. For DIS, finding the word distance between markables and normalizing the markables are crucial to system performance. For TEST markables, withholding date and word-distance

information significantly reduces the performance of the system.

The two semantic categories that were not significantly affected by the removal of any feature showed different reactions to feature removal. In CONS, all-feature removals cause a negative effect on system performance, while in SYMP, the removal of features in a majority of cases (15/23) actually caused a slight increase in system performance.

4.7 Discussion

Our evaluations indicate that both the multi-perspective approach and the diversity of the very large feature set contribute to the superiority of the all-features system over the baseline.

For all semantic categories, the multi-perspective approach contributes to only a fraction of the system's improvement from the baseline. There are two exceptions, however. A majority of the increase in CONS resolution performance over the baseline seems to be due to the multi-perspective approach rather than to a larger feature set than the baseline (Table 26 and 27). This result indicates that additional features did not contribute to the increase in the system's ability to resolve CONS coreference, once again showing that the token-match feature is likely the most powerful feature for locating coreferent CONS markables. Evaluation data [Table 26 and 27] indicates that the large increase in B-CUBED score for DIS markables is also due to the multi-perspective approach Further investigation will need to be conducted to find the disparity between the measures.

When examining system performance changes for each type of string-match markables, we find that the all-features system improves partial-match and no-match markables resolution by a wide margin. In fact, it is the system features that are included to better no-match and partial-match resolutions (e.g., edit-distance, normalized-edit-distance, normalized-token-match, noun-match features, prefix-match, umls-concept-match, and umls-concept-token-match) that significantly and directly affect system performance.

In the complementary effect evaluation, we find that each semantic category has its own distinct reaction to feature removal. SYMP resolution improves slightly when features are removed, while CONS resolution suffers slightly. The feature removal tests find that only DIS, MED, and TEST coreference resolution have any significant changes. Out of the three semantic categories, MED resolution is affected by the removal of many features, while DIS and TEST resolutions are affected by the removal of two features each

Only two features consistently have a positive (significant or non-significant) contribution to system performance. In particular, removing word-distance from the all-feature system causes a decrease in performance for resolving coreference in all semantic categories, though only in two of the semantic categories (TEST and DIS) is the decrease significant. The removal of the other feature, umls-type, causes only a slight decrease in coreference resolution on all 5 categories.

Our evaluations have found contributory features and the strengths of our system. However, where are the weaknesses to our approach? In what follows, we will analyze the system prediction errors for each semantic category. For each category, we will give examples of the prevalent false alarm and false negative errors, why they occur, and any potential solutions to the problem.

4.7.1 CONS

Errors in CONS markable resolution mainly stem from the limitations of orthographic features. The decision tree generated during the CONS resolution process (Appendix.A) reveals that features selected

by C4.5 are primarily those that use spelling to evaluate similarity between two markables: token-match-antec, normalized-token-match-anaph-base, normalized-distance, prefix-match-antec, prefix-match-greedy, noun-match-stingy, last-name-match. The system relies heavily on spelling to discover coreference because CONS markables are all named-entities.

4.7.1.1 False Alarm

There are only a total of six CONS false alarms. The six false alarms fall into two groups. In 3 out of the 6 cases, the markable pairs match in many uninformative tokens, e.g., honorifics, punctuations, etc. These token-matches result in a misleadingly high entity-token-match value, causing the system to believe that the markables corefer. Example 1 demonstrates this problem. In each markable, 3 out of 5 tokens match; however, comma, which is considered a token, is a punctuation mark and "M.D." is an extremely common honorific. These tokens should not be used as part of token-match evaluation because overlaps of these tokens do not increase the likelihood of markable pairs being coreferent.

Example 1:
"[Jack M. Brown] , M.D." and "[Stanley M. Red] , M.D." result in token-match-antec value of .6

The remaining false alarms are due to ambiguous cases. For example, our annotations indicate that "Hematology/Oncology Consultation" and "Hematology" are not coreferent. However, depending on the interpretation of the "/", the doctor could have mean that the Hematology and Oncology doctor are the same person. There was disagreement during annotation about such assignments and we believe during evaluation such ambiguous cases should be evaluated separately from the markable pairs that are more straightforward.

4.7.1.2 False Negative

There are more CONS false negatives than false alarms; in 9 out of the 17 cases, elimination of common irrelevant tokens can correct the false negative errors. Example 2 and 3 showcase false negative pairs that are results of noisy token-match evaluation. These examples are essentially the same problems as mentioned in the false alarm section, however, instead of misleadingly high token-match values, unmatched irrelevant tokens causes misleadingly low token-match values.

Examples 2 and 3:

<cons> primary care physician is Dr. [John] [Brown] </cons> vs. <cons>Dr. [John] [A.] [Brown] , M.D. </cons>

<cons> Hematology/Oncology team </cons> vs. <cons> Hematology/Oncology consultation </cons>

Another type of false negative error occurs when the system fails to recognize two completely unmatched markables as being coreferent. For example, in one of the records, "primary care physician" and "Dr. [Brown]" corefer. Usually in these cases there is a unifying markable that clarifies the coreference relationship (see Example 2); however, because it is so long, comparing "primary care physician" or "Dr. Brown" to it will evaluate to a low token-match score.

If a program can identify the subject and object in an "is-a" sentence or the two parts of an apposition, then it can divide the unifying markable into two "sub-markables." Candidates that corefer to either sub-markable can be accepted as being coreferent to the unifying markable. This approach is valid because appositive relationships are equivalence relationships. Even though "is-a" sentences are indications of hypernym/hyponym relationships rather than equivalence relationships, we observe that in our corpus, hypernyms/hyponyms are often used interchangeably to refer to a single entity. As a result, "is-a" sentences are often indicative of coreference relationships. If the system concludes that a markable is coreferent to one of the sub-markables, then it should follow that the markable is also coreferent to the other sub-markable. With the low token-match score problem solved, unmatched markables can be coreferent to the same unifying markable. Aggressive-merge clustering then will cluster the markables as being coreferent.

CONS markables are different from the other semantic categories that we examine. CONS markables represent people or organizations, while markables in the other categories represent events. This difference means that exact-match CONS pairs are much more likely to be coreferent than exact-match markable pairs in any of the other categories. CONS markables, however, are also longer than DIS, SYMP, TEST, and MED markables because CONS markables include honorifics and appositions. There is, therefore, more noise in CONS markables than other semantic categories. As a result, high token-match scores for partial-match markables can be somewhat misleading.

4.7.2 DIS

4.7.2.1 False Alarm

We find that DIS false alarms mainly fall into two categories of string-match markables. 46 of false alarm pairs are from exact-matches, while another 39 are substring-matches. Substring-match is a more specific kind of partial-match where one markable in a markable pair is entirely made up of tokens from the other markable in the pair. For example, "chest pain" is a substring of "left chest pain". Therefore, the two markables are substring-match pairs. We now discuss string-match errors in more detail.

Exact-match errors occur because it is difficult for the system to distinguish between coreferent and non-coreferent exact-match pairs. Example 4 demonstrates the problem of using high token-match value as an indication for coreference. Both disease markables ("right upper lobe nodule" and "right upper lobe nodules") are referring to the same type of disease; however, they are clearly not referring to the same

occurrence. How can these two markables be distinguished? We currently use temporal information. The temporal information we mark is based on when there is an update on the disease status. Performing date-match based on this type of temporal information allows the system to identify disease markables mentioned on the same date; it does not help link disease updates across multiple dates.

Example 4:

S1. In September of 1999 , a <test> CT of the chest </test> show a <dis-pres> right upper lobe nodule </dis-pres>

S2. In March of 2000 , two more <dis-pres> right upper lobe nodules </dis-pres> were found.

The key to distinguishing these cases is in the article or modifier that describes each markable. Definite and indefinite articles cue readers on whether markables refer to a previously mentioned entity or a newly introduced one. Words such as "the", "those", "these" are usually used to refer to previously declared markables, while indefinite articles like "a" are often used to bring a previously unknown item into focus.

Example 5 and 6:

"... , which showed <dis-pres> three vessel disease </dis-pres> 50 to 80 <dis-pres> stenosis </dis-pres> in the left anterior descending and 95 <dis-pres> stenosis </dis-pres> in the D1 , 50 <dis-pres> stenosis </dis-pres> at the circumflex ..."

"it was believed that the cause of the <dis-pres> anemia </dis-pres> was likely multifactorial including <dis-pres> iron deficiency anemia </dis-pres> , <dis-pres> anemia of chronic disease </dis-pres> , and <dis-pres> anemia </dis-pres> due to <dis-pres> renal failure/diabetes </dis-pres>

Example 5 and 6 show other cases where simple temporal reasoning information combined with token-match is not informative enough. In both cases, the markables listed are mentioned in the same time frame, but none refer to each other. In fact, it is the syntax of the sentence, more specifically listing of markables, that implies these markables are distinct from each other.

Additionally, the context in which the markables are used helps coreference resolution. Errors often occur when the system concludes that past occurrences of diseases are coreferent to the current occurrence. The system can use both the location of each markable within and the markable's surrounding sentential context to locate temporal clues. Past cases of a disease are often placed under the PAST MEDICAL HISTORY section and referred to with keywords such as "history of" or "recurrent". The current case of the disease almost never appears in the PAST MEDICAL HISTORY section, unless it is a chronic disease. From these trends, we believe the identification of temporal keywords surrounding a markable can help a system locate other coreferent markables. Example 7 is an example in which the system classifies an old bout of pseudomonas with the current case.

Example 7:

S1. History of <dis-pres> MRSA </dis-pres> , <dis-pres> pneumonia </dis-pres> and <dis-pres> pseudomonas pneumonia </dis-pres> that is resistant to ....

S2. Her <test> sputum </test> was sent for <test> culture </test> and came back positive for <dis-pres> pseudomonas </dis-pres>.

From our error analysis, we also find that even though markable pairs' presence-statuses do not match, the system will still classify a pair of markables as coreferent (Example 8). In Example 8, the best indication of non-coreference is the presence feature mismatch. While presence is supposed to identify coreferent and non-coreferent markables by examining if markables have matching presence-status, we believe the quality of the value of the presence feature can improve from how it is derived now. Currently, any markables pairs that have different presence-status, e.g., possible (poss) vs. possessed by someone else (some), possible vs. absent (abs), absent vs. present (pres), etc., is assigned a presence value of 0. While markable pairs that match in presence-status are assigned distinct presence values of 1-4 depending on the type of presence-status match (e.g. both are "poss", both are "abs", both are "pres", both are "some"). These values are purely categorical representations of the different combinations of presence-statuses. They themselves do not in any way represent likelihood for coreference. We believe markable pairs possessing markables with different combinations of non-matching presence-status should also be assigned different values. For example, if one markable is "poss" and another is "some", then the presence value should be 5; if one markable is "poss" and another is "abs" then the presence value should be 6, and so on for other combinations. The reason is that the coreference likelihood varies depending on what type of presence-status mismatch the markable pair contains. For example, disease possessed by someone else is never matched with diseases of any other presence-status, but a disease that is asserted to be possibly present can be coreferent to another mention that is asserted to be absent or present. By assigning all of these a value of 0, we ignore the information presented to us when markables have different presence-statuses.

Example 8:

S1. The patient is an elderly woman with an extensive cardiac history including <dis-pres> coronary artery disease </dis-pres> status post <med> coronary artery bypass grafting </med> ...

S2. Her brother has <symps-some> hypertension </symps-some> , and there is <dis-some> coronary artery disease </dis-some> in her brother and sister .

4.7.2.2 False Negative

Unlike the false alarm pairs where most errors stem from substring-match or exact-match markables, most false negatives come from no-match pairs. 69 of false alarms come from no-match markables and another 17 from partial-match markables that do not count as substring-matches.

Errors from no-match usually come from the system's lack of medical knowledge related to generalizations. For example, "pneumonia" is often referred to as an "infectious process". However there is no way for the system to know that the two are related. The lone hope for solving this problem is in UMLS MetaMap; however, the term "infectious process" is too broad even for UMLS to categorize as any specific concept. Another common error is that the system doesn't recognize the connection between "blood" and "bleed". These problems indicate the importance of having knowledge sources that recognize broader concepts that each markable can refer to. Perhaps the use of the UMLS semantic network can help.

Besides medical knowledge shortage, the system also does not recognize abbreviations. As a result terms like "MI" and "myocardial infarction" or "PE" and "pleural effusion" are not recognized as being related. Medical acronym lists exist. We believe these lists, when coupled with a distance metric, may be able to resolve most abbreviation errors. However, overlapping acronyms for different terms can cause ambiguity, for example, "breath sounds" and "bowel sounds" are both abbreviated BS. This ambiguity only causes a problem if two overlapping terms both appear in the same discharge summary. If we know how frequently co-occurrences of overlapping terms occur, we would have a better idea of the usefulness of acronym lists.

The above cases mostly stem from no-match markables, in cases where there is some overlap between markables, the false negatives occur due to the lack of context. For example, in the example below, recognizing that both ulcers are of the duodenum would have been enough for coreference resolution. However, the word "duodenum" does not appear in the first markable, rather it appears two words before the markable is mentioned. We believe the key to recognizing these contexts for disease markables is in finding location/body-parts words around or in markables. Because many diseases affect isolated parts of the body, markables that match in body-part context and disease name are likely to be coreferent. Of course, this method is not perfect. There will certainly be cases like Example 4 where multiple occurrences of the same disease occur.

Example 9:

S1. ... ; duodenum , multiple <dis-pres> cratered nonbleeding ulcers </dis-pres> ranging in size from 2 to 5 mm were found in the duodenal bulb .

S2. <dis-pres> Gastrointestinal bleed </dis-pres> , likely from <dis-pres> duodenal ulcers </dis-pres>

It is also worth noting that both from examining the decision tree and our evaluation of feature contribution to system performance, we find word-distance (the distance between two markables as measured by the number of words in between them) to be of importance. The reason for the importance of word-distance is likely due to the prevalence of substring-match markable pairs. In these markables, the substring markable is often missing one key piece of information (such as location) that is preventing the system from identifying whether it corefers to an antecedent markable. By using word-distance, our system treats substring markables similar to how pronouns are treated by other systems, assigning them to markables that are closest to the substring markable.

4.7.3 MED

While disease markable resolution places an emphasis on location and general temporal information, MED markable resolution can be done by examining the delivery method of medication and the time of medical procedures.

4.7.3.1 False Alarm

Like some of the other semantic categories, false alarms in MED markables are also dominated by exact-match (33) and substring-match (44) markables. As mentioned previously, MED entities consist of medical procedures and medications. Errors are split into these two categories fairly evenly, 48 versus 52 respectively.

There are two primary causes of errors in MED markables. The first type of error only applies to

medications. In our definition, different delivery forms of the same medication are non-coreferent. For example, "Nitroglycerin sublingual" is different from "Nitroglycerin drip". We make this distinction because a patient may be treated with two different forms of the same drug at the same time, and other references may refer to only one type of medication.

The other type of error is due to the lack of more contextual clues to help the system identify different instances. Because many medical procedures are likely to be repeated several times, there are often related words surrounding the markables that identify whether they are identical cases to nearby markables. In Example 10, the use of "Other" before "ablation" in the second sentence is an indicator of different references. The solution to locating the coreference hint is once again to examine an n-word window around the markable.

Example 10:

S1. First <med> ablation </med> was <med> right bundle branch re-entry </med> vs. <med> nodofascicular BPT </med> .

S2. Other <med> ablation </med> was an <med> AV NRT ablation </med>

Furthermore, recognition of "is-a" syntactic structure would also help resolve coreferences for the markables within each sentence. The first "is-a" sentence in Example 10 is a simple equivalence relationship, but the second sentence is non-trivial to resolve due to the comparison relationship.

In general, false alarms occur because the prediction model is overly reliant on orthographic features as an indicator for coreference. A look at the decision tree for MED shows that any markable pairs with edit-distance <= 1 will automatically be classified as coreferent. This assignment causes 37 out of the 90 false alarms. While we have introduced measures that can potentially help improve coreference, it is possible they will not be used if the C4.5 algorithm deems the measures to be only slightly useful or if the pruning process groups together some paths. The partition we just mentioned has an extremely good information gain ratio because it results in 796 correct predictions and 37 wrong predictions. Because there is an overwhelming number of MED entities, correcting the 37 wrong predictions will make little difference in information gain, and the system will likely ignore that edit-distance <= 1 branch for further partitioning. It is also possible that the system did find other features; however, the pruning process merged the features together.

The problem with the above example is that the system gives equal credit for coreferent and non-coreferent pairs. Because, there are so many more non-coreferent pairs in the data set than coreferent ones, partitions that isolate a large number of non-coreferent pairs and some coreferent ones will dominate. We believe training the system with a cost-matrix that assigns a heavier penalty for wrong coreferent classifications and gives less credit for correct non-coreferent pairs than coreferent ones could help offset the system bias towards non-coreferent pairs.

4.7.3.2 False Negative

False negatives in MED category seem to result primarily due to a lack of synonymy and hypernymy knowledge. For example, the system did not realize that "plasma exchange" is just another term for "plasmapheresis" or that "resection" is a type of "surgery".

Other false negative errors may be resolvable by features we included such as using prefix-match to

resolve "diuresis" and "diurese." However due to the prediction model that is formed by important features to decipher coreference are not chosen. We believe the primary problem is in how we have constructed these features. In particular, token-match causes dependencies that will limit the ability of the system to decipher when it is appropriate to use a particular feature. In the case of prefix-match, any strings that exactly match will have a prefix-match value of 1. However, we intended to use prefix-match to locate markables that have similar spelling. By including exact-match markables from this group, we remove the ability of using prefix-match as an indicator of coreference for only no-match markable pairs. To remove the dependence between token-match and prefix-match, we believe markable pairs that match exactly should have a prefix-match value of "?". Other features are likely to have dependencies that we accidentally introduced, and should also be modified accordingly for value assignment.

4.7.4 SYMP

4.7.4.1 False Alarm

Unlike the other categories, the percent of false alarm SYMP markables that are exact-match pairs is relatively low at only 26. The false alarm pairs that are substring-match or prefix-matches also each make up 26 of all false alarms. Even though SYMP errors stem from more diverse types of markables, there is much less variety in terms of values assigned to features. What we mean is most feature values are either going to be 0 or 1. This is because most terms are single token words, so as a result many exact-match and partial-match markable pairs will have prefix-match or either token-match-anaph or token-match-antec value of 1.

Instead, the prediction model uses normalized-edit-distance as an indicator of coreference. Normalized-edit-distance <= 2 is a good metric because there are quite a few cases of morphologically variant markable pairs. Symptoms are often referred to in the noun and adjective forms, because they are as much concepts as they are descriptors of patient states. For example, "hypertensive" appears just as often as "hypertension". However, locating morphological variants only identifies markables as being similar concepts. The markables may still be referring to two separate occurrences of a symptom. Therefore, we still need to use other contextual methods to determine coreference.

4.7.4.2 False Negative

There are 35 false negative pairs. 9 out of the 35 false negative pairs require world knowledge. Of these, 6 are able to attain the knowledge from UMLS MetaMap. We define markables as requiring world knowledge if simply by examining two markables we cannot tell they are related. Some examples are synonyms like "dyspnea" and "shortness of breath" or "fever" and "febrile". While the umls-concept-match feature recognized these terms as the same, they were still classified as non-coreferent. The decision tree path for these markables is shown below:

```
normalized-edit-distance > 2
|  normalized-token-match-stingy-base <= 0.166667
|  |  prefix-match-greedy <= 0.666667: False (8714.0/18.0)
```

Unfortunately, nowhere in this path does the decision tree use the umls-concept-match feature. We believe this problem is the exact same problem as the one we described in the MED false alarms section; it is related to pruning and the prevalence of non-coreferent pairs. Had the program decided to continue building the tree, 6 of the 18 errors would have been correctly predicted. But because prefix-match-greedy is able to correctly predict non-coreferent markables at such a high level, with relatively low error, no further actions were taken by the algorithm to partition the data further. We believe the pruning

process merges too many leaf nodes.

Part of the problem may also be that there are too many token-match related features. Feature values for token-match, prefix-match, umls-concept-match, normalized-token-match, edit-distance, and normalized-edit-distance are all correlated because if token-match is 1 then the other features will also be either 0 (edit-distance features) or 1 (token/concept-match features). We need to eliminate some of these features and assign values more selectively. For example, umls-concept-match is useful only on exact-match markables because otherwise it has the same value as token-match. So to make umls-concept-match and token-match more independent of each other, umls-concept-match should be assigned an "?" value for exact-match markable pairs.

Example 11:

S1. ...who presents with a two to four week history of increasing <symps-pres> shortness of breath </symps-pres>

S2. Approximately two to four week ago , the patient begin developing <symps-pres> dyspnea </symps-pres> on exertion.

Context is also important for removing SYMP false alarms. In the example below, we know "shortness of breath" and "dyspnea" are coreferent because the two terms are synonyms and both occurred "two to four week ago." Recognizing similar context can be done in a couple of ways. The system may convert "two to four weeks ago" into an actual time and then compare the time stamps for each markable, similar to our date features. Another way is to compare surrounding text in a way similar to [39] As mentioned in the related works section, Pederson et al. used bi-gram matches around a 50 word window to identify coreferent named-entities. We also believe that finding the length of the longest common subsequence of the sentences that contain each markable may be useful. The key observation is that consecutive matching tokens in surrounding text are more likely to indicate coreference of two markables than multiple disconnected token-matches.

The previous example demonstrates that sometimes more than one feature is needed to make coreference resolution possible. We consider these to be difficult, multi-step cases for coreference resolution. Consider the example below, the system needs to first understand that "GI" stands for "gastrointestinal", and then make the assessment that the "gastrointestinal area" is around the "abdomen".

Example 12:

S1. Her <symps-pres> non-cardiac abdominal pain </symps-pres> was distinct form her prior <symps-pres> angina </symps-pres> which lowered the suspicion for an <dis-poss> acute coronary syndrome </dis-poss>

S2. She was also started on <med> Protonix </med> for her <symps-pres> GI symptoms</test>

4.7.5 TEST

Out of markables in all semantic categories, the system performed worst on TEST markables. In all other categories, token-match is an extremely useful feature by itself, however, this is not the case for TEST which contains more non-coreferent exact-match pairs than coreferent ones. In TEST, the combination of token-match, word-distance, date, and other contextual features results in improved resolution. The

prediction model formed by these features has a relatively low recall, signifying that false negatives are a common problem. False alarms, though not nearly as prevalent, also exist.

4.7.5.1 False Alarm

81 of TEST false alarms are exact-match pairs. Through our error analysis we have found many ways to recognize non-coreference amongst exact-match pairs. All such cases are related to contextual and lexical clues because non-coreferent exact-match markables cannot be identified through features that are derived directly from markables. In the following error analysis we present some common problems, most of which involve exact-match pairs.

Example 13:

S1. Recent <test> echocardiogram </test> revealed an <test> ejection fraction </test> of <results> 55 </results> .

S2. Other studies of note were a recent <test> echocardiogram </test> from August 9, 2001 which revealed an <test> ejection fraction </test> of <results> 60 </results> .

In many cases, if TEST markables have RESULT markables nearby, comparing the RESULT markables can help determine TEST coreference. In the current feature set, left-markable-all-category and right-markable-all-category may perform a similar function if a RESULT markable immediately precedes or follows the TEST markable; however, no feature specifically compares RESULT markables. Also our neighbor matching strategy is somewhat faulty because certain markable pairs are likely to appear together. For example, "echocardiogram" is used to find the "ejection fraction". This means, "echocardiogram" and "ejection fraction" will have a right-markable-all-category and a left-markable-all-category value of 1. These values falsely indicate coreference and contextual similarity. In these cases, more precise neighbor matching can improve coreference. In addition, frequency analysis using TFIDF or other methods can also discount overlaps between commonly appearing token pairs.

As is with the case with many other indicators, RESULT matching is not a perfect indicator of coreference. It is entirely possible for two different tests to return the same result. Consider the example below:

Example 14:

S1. <test> Iron studies </test> revealed <test> TBC </test> <results> 228 </results> , <test> haptoglobin </test> <results> 451 </results> , ... <test> total bilirubin </test> <results> 0.3 </results> , ...

S2. Was <results> positive </results> for <test> glucose </test> <results> of greater than 1000 </results> , ... <test> total bilirubin </test> <results> 0.3 </results>

Sentence 1 refers to results from iron studies, while sentence 2 refers to results from a urinalysis. The system classified the "total bilirubin" markables as coreferent due to exact-match, a right-markable-all-category value of 1, and umls-concept-match of 1. However, the TEST markables are not coreferent to each other because they do not originate from the same test event. In addition, the surrounding TEST markables for the markables in question do not match, yield a further clue that the TEST markables are not coreferent.

While it is possible to find heuristics and patterns that suggest coreference or non-coreference in most cases, certain ones require the system to have deep understanding of natural language and a degree of reasoning. In Example 15, to resolve that "esophaogastroduodensocopy" in sentence 1 and sentence 2 are not coreferent to each other requires the understanding that in sentence 1 the operation had been done earlier in time because it appeared in the HISTORY OF PRESENT ILLNESS section. The markable in sentence 2 is mentioned in the context of the current HOSPITAL COURSE. In particular, the operation is being considered while the operation in sentence 1 has already been performed. Therefore, to successfully resolve coreference of the markable pairs, the system needs to understand the contexts in which the markables are used and their temporal order.

Example 15:

S1. An <test> esophagogastroduodenoscopy </test> was done , and revealed <dis-pres> fresh bleeding </dis-pres> with large <dis-pres> blood </dis-pres> in the fundus of the stomach .

S2. The patient understood that this eliminated the possibility of gaining control of <dis-pres> bleed </dis-pres> via <test> esophagogastroduodenoscopy </test> .

4.7.5.2 False Negative

TEST false negatives occur in similar fashions as false negatives in other semantic categories. The small data set and the large number of non-coreferent markable pairs compared to coreferent pairs cause the system to be biased towards classifying markables as non-coreferent. The following example demonstrates how these two factors limit the effectiveness of our system:

Example 16:

S1. Prior to the <med> thoracentesis </med> she had a <test> chest CT </test> , which showed ....

S2. A <test> CT </test> had been performed prior to her <med> thoracentesis </med> .

The "CT" in sentence 2 and the "Chest CT" in sentence 1 obviously corefer. However, the decision tree model indicates that any token-match value of half or less will automatically be non-coreferent. Because test tokens are usually only one or two tokens, paired markables like "CT" and "Chest CT" are not processed. Such shallow categorizations are bound to cause errors. Modifications to the decision tree algorithm to ensure a certain level of depth to each leaf in the tree maybe useful. Another alternative is to increase the data set size and use a cost sensitive matrix so that the system is not biased to assigning pairs to be non-coreferent because non-coreferent pairs are more prominent.

5 FUTURE WORKS

This research has only scratched the surface of coreference resolution on medical discharge summaries. Future research that aims to optimize each step of the resolution process (pairing markables, feature evaluation and incorporation, training algorithm, and clustering) may improve system performance in dramatic ways.

In this research, we used McCarthy and Lehnert's method to pair markable pairs. We found both advantages and disadvantages to this method. Namely, this approach gave a large training set for the machine learner, but it also created predominantly non-coreferent pairs. Depending on the type of a markable or markable pair being classified (indefinite nouns, proper nouns, exact-match, and substring-match), the system can choose different pairing methods. For example, indefinite noun coreferences are likely to span great distances, so a complete pairing approach might be better. A noun that is a substring of a nearby markable, however, often acts more like a pronoun, and a pairing approach akin to Ng and Cardie's may be more appropriate. Further research into how to select and pair training data may improve prediction model output and system performance.

This research is primarily aimed to discover insights on salient features for coreference resolution of different semantic categories that appear in hospital discharge summaries. Through our error analysis, we have a better understanding of the limitations of the current approach and how it affects each semantic category. Markables in different semantic categories have different characteristics and will benefit from different approaches.

CONS often contain preposition and punctuation tokens. These non-descript tokens should not be considered part of the markable. To enhance token-match, a stop-list or frequency analysis can be used to disregard common, uninformative tokens. In addition, from our analysis, the power of last-name-match feature has not been fully exploited. Our system must be able to better recognize last names, and assign "?", rather than "False" to markable pairs that contain markables with no last names.

In addition, all semantic categories can benefit from adjustments to syntactic, temporal, and lexical approaches to resolving coreference. Better syntactic analysis can improve system precision. Definite and indefinite articles can be useful for resolving coreference of close distance markables. Certain sentence structures, such as is-a and appositive sentences, are prone to containing coreferent markables. Other sentences with list structures can be strong indicators for non-coreference.

Further research also needs to be done on tokens surrounding markables. Some special cases are:

- Unmatched markable tokens that appear in the surrounding text of the other markables,
- Definite articles that indicate the current markable is referring to a previously introduced entity, and
- Temporal tokens such as "again", "repeat", and "previous".

Separately, even though UMLS imparted some medical knowledge into our system, our attempt to use UMLS MetaMap for locating semantically similar markables is very basic. Future efforts should leverage the UMLS Semantic Network to find related UMLS concepts.

As we have learned, string similarity is only an indication of coreference. In the case of TEST, a majority of markable pairs that are similar to each other are not coreferent. When analyzing MED markables, we also find markable pairs that evaluate to high token-match values, but are non-coreferent. These markables often differ in one token, however the tokens that differ are extremely important to the interpretation of the markables. These tokens often detail how a medication is delivered. Our token-match evaluation method assigns each token equal weight; however, in the case of MED markables, certain tokens that are particularly important should be weighted more. Frequency analysis of token differences and similarities between coreferent and non-coreferent markables can help determine weighting.

Another problem found during MED markable resolution is the system's inability to recognize hypernyms. Hypernym relationships require world knowledge, however, the UMLS Metathesaurus only maps equivalence relationships. Determining how hypernym relationships can be located and using them to determine coreference will result in a more complete resolution system.

As mentioned in the related works section, Pederson et al. used bi-gram matches around a 50 word window to identify coreferent same-name entities. Similarly, finding the length of the longest common subsequence of the sentences that contain each markable may be useful because matching consecutive tokens are more likely to indicate coreference than matching non-consecutive tokens. In this experiment, the system evaluated markable similarity by considering token-match percentage without giving extra consideration for consecutive tokens.

Simply examining the a priori coreference probabilities for different string-match types suggests that different string-match types require different resolution methods. To incorporate consecutive token-match in token-match evaluations, we believe partial-match markables should be further divided into substring-match and other-partial-match markables. Separately resolving each type of string-match markables (exact-match, substring-match, other-partial-match, and no-match) will allow the system to optimize resolution by finding features that are useful for resolving each type of markable pairs.

Many of our suggestions for improvement are modular approaches that improve on the current system. Any realizable gains from these improvements are contingent upon the system being trained on a larger data set. Our data set contains too few coreferent pairs. It is also important that any future methods remove the prominence of non-coreferent pairs to coreferent pairs in our data set. This can either be done by removing non-coreferent data points from the data set or cost-sensitizing our approach.

From our research, we have also found that C4.5 cannot find the optimal prediction model due to dependencies between features, the small data set, and dependencies as a result of natural language. Future research should first use larger data sets to train the prediction model so that the true limitations of C4.5 can be found. Alteration to the decision tree algorithm or substitution of it for another algorithm that better suits the characteristics of the task is the next logical step. An alteration that may improve system performance is hard-coding precedence of features used by the algorithm to reflect the dependencies of the feature set. For example, in the current feature set, we know that certain features should be used before others, e.g., umls-type-anaph should be used before the umls-concept-match features. During training, if both the umls-type-anaph and umls-concept-match features are used, then umls-type-anaph should be a parent node of umls-concept-match. Users should also be allowed to tell the algorithm which features must be included in the prediction model.

Our current greedy clustering approach to grouping predicted markables seems to work well. However, a probabilistic approach to clustering may yield better results. This would require machine learning approaches that assign probabilities to predictions. Using such an approach not only improves clustering

but can yield new approaches to coreference resolution. Current methods use a one iteration approach to resolve coreferences where the system examines all the text data once and attempts to classify coreference in one run through. However, human readers often reread a difficult passage several times to resolve coreference, each time incorporating some new knowledge or assumptions from other parts of the passage to improve their guesses. A probabilistic learning approach can imitate such human behavior by first locating high probability coreferent markable pairs, clustering and extracting information from these markables pairs, and then iteratively improving its guess about coreference.

6 CONCLUSION

We have presented a coreference resolution system modeled after the McCarthy and Lehnert approach with significant expansions in feature set. We apply the system to each of five commonly appearing semantic categories in medical discharge summaries. Initial evaluation on the data set finds that token-match as a standalone feature can be extremely helpful in deciphering coreference in four of the five semantic categories. Due to this evaluation, we use a token-match baseline to measure system performance on easily found coreferent markables. MED coreference is easily identifiable, while TEST coreference is extremely difficult.

Evaluation of the all-feature system finds the system to be fairly precise, with the lowest MUC precision being .730 and lowest B-CUBED precision of .983. With the exception of TEST (recall = .388), the system also performs well in recall, with the lowest MUC recall being .782 and the lowest B-CUBED recall being .947. TEST markables have the biggest MUC F-measure difference (.506) between baseline and all-feature system. MED experiences the smallest difference (.032). The other three semantic categories also have gains from the baseline. The gains primarily come from improved coreference recall from the extra features.

Our analysis of the system feature set finds the multi-perspective approach to be a useful addition that positively contributed to the performance of our coreference system. Analysis of individual features' impact shows edit-distance to significantly improve CONS, DIS, and MED coreference over the token-match and plurality system baseline; noun-match and UMLS features to significantly increase DIS and MED coreference; and prefix-match to significantly increase DIS and SYMP coreferences. The complexity of TEST resolution is demonstrated by the fact that no feature by itself significantly increases TEST resolution performance from the token-match and plurality system baseline. Improved TEST resolution performance is caused by multiple features working together.

To evaluate multiple feature interactions, we perform a leave-one-feature-out test on our all-feature system. No significant system performance increases are detected from feature removal, however, there are some notable performance degradations due to feature removals. System performance significantly decreased when left-markable-all-category, normalized-token-match, noun-match, plurality, sentence-token-match, date, or word-distance is removed from the all-feature set. Only the last two features affect resolution performance on non MED-markables.

Our experiments indicate that token-match is an important feature for coreference resolution of discharge summaries. However, incorporating contextual features into the feature set is important for further system performance gains. In particular, the context of the sentence that contains the markable, the temporal time frame of the markable, and the distance between markables are all important. None of these features, however, apply across all semantic categories. The relevant features for coreference resolution do vary between different semantic categories, We believe a reliable semantic categorizer, therefore, is an important precursor for any high performance EMR coreference resolution systems,

7 BIBLIOGRAPHY

[1] Aberdeen, J., J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain. 1995. MITRE: Description of the alembic system used for MUC-6. Proceedings of the Sixth Message Understanding Conference (MUC-6), pages 141–155. Columbia, Maryland: Morgan Kaufmann Publishers, Inc.

[2] Aronson, A. 2001. Effective mapping of biomedical text to the UMLS metathesaurus: the metamap program. AMIA.

[3] Bagga, A. and B. Baldwin. 1998. Algorithms for scoring coreference chains. In Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation (LREC'98), pages 563--566.

[4] Baker, M. L. Large employers to provide online personal health records [web page]. http://www.eweek.com/article2/0,1895,2069942,00.asp. Accessed August 12, 2007.

[5] Barr S. Bill to Promote Electronic Health Records. The Washington Post. March 2, 2006: D04.

[6] Berger, A. L., S. A. D. Pietra, and V. J. D. Pietra. 1996. A maximum entropy approach to natural language processing. Computational Linguistics, 1(22):39–71.

[7] Bilenko, M. and R. Mooney. 2002. Learning to combine trained distance metrics for duplicate detection in databases. Technical Report AI 02-296, Artificial Intelligence Lab, University of Texas at Austin. Available from http://www.cs.utexas.edu/users/ml/papers/marlin-tr-02.pdf.

[8] Bodenreider, O. 2004. The Unified Medical Language System(UMLS): integrating biomedical terminology. Nucleic Acids Res. 2004;32(Database issue):D267–D270.

[9] Bontcheva, K., M. Dimitrov, D. Maynard, V. Tablan, and H. Cunningham. 2002. Shallow methods for named entity coreference resolution. Chaînes de références et résolveurs d'anaphores workshop TALN 2002, Nancy, France,.

[10] Brill, E. 1992. A simple rule-based part of speech tagger. In Proceedings of the Third Conference on Applied Natural Language Processing, pages 152-155.

[11] Cardie, C. and K. Wagstaff. 1999. Noun phrase coreference as clustering. In Proceedings of the 1999 SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, Md. pages 82-89.

[12] Castano, J., J. Zhang, and J. Pustejovsky. 2002. Anaphora resolution in biomedical literature. In International Symposium on Reference Resolution, Alicante, Spain.

[13] Chieu, H. L. and H. T. Ng. 2003. Named entity recognition with a maximum entropy approach. In Proceedings of CoNLL-2003, Edmonton, Canada, pages 160-163.

[14] Chinchor, N. 1995. Statistical significance of MUC-6 results. Proceedings of the Sixth Message Understanding Conference (MUC-6), pages 39-43. Columbia, Maryland: Morgan Kaufmann Publishers, Inc.

[15] Chincor, N. 1996. Overview of MUC-7. [web page]. http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/overview.html. Accessed Aug 12, 2007.

[16] Chinchor, N. 1998. MUC-7 Named entity task definition dry run version, version 3.5 17 September 1997. Proceedings of the Seventh Message Understanding Conference (MUC-7) (to appear). Fairfax, Virginia: Morgan Kaufmann Publishers, Inc. URL: ftp://online.muc.saic.com/NE/training/guidelines/NE.task.def.3.5.ps.

[17] Cohen, W. 1995. Fast effective rule induction. In Proceedings of the Twelfth International Conference on Machine Learning.

[18] Cohen, W. 2000. Data integration using similarity joins and a word-based information representation language. ACM Transactions on Information Systems 18(3):288–321.

[19] Cohen, W., P. Ravikumar, and S.E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03).

[20] Collins, M. 1997. Three generative, lexicalised models for statistical parsing. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, pages 16–23.

[21] Cunningham, H., D. Maynard, K. Bontcheva, V. Tablan, C. Ursu, M. Dimitrov, M. Dowman, N. Aswani, I. Roberts, Y. Li, and A. Sahfirin. Developing language processing components with gate version 4 (a user guide). [web page]. http://gate.ac.uk/sale/tao/index.html#annie. Last Accessed Aug 12, 2007.

[22] Daille, B. 1996. Study and implementation of combined techniques for automatic extraction of terminology. In Judith L. Klavans and Philip Resnik, editors, The Balancing Act: Combining Symbolic and Statistical Approaches to Language. MIT Press, Cambridge, MA.

[23] Elango, P. Coreference resolution: a survey. University of Wisconsin, Madison.

[24] Fisher, D., S. Soderland, J. McCarthy, F. Feng and W. Lehnert. Description of the umass systems as used for muc-6. Proceedings of the 6th Message Understanding Conference, 1996. Columbia, MD.

[25] Fleming, D. 2006. The potential of electronic medical records for health service management.

[26] Harvard Neural System Group. Python modules. [web page] http://www.nmr.mgh.harvard.edu/Neural_Systems_Group/gary/python.html. Accessed 20 June, 2007

[27] Humphreys B.L., D.A. Lindberg, H. M. Schoolman, and G.O. Barnett. 1998. The Unified Medical Language System: an informatics research collaboration. J. Am. Medi. Inform. Assoc., 5, 1-11.

[28] Li, X., P. Morie, and D. Roth. 2004. Identification and tracing of ambiguous names: discriminative and generative approaches. In Proceedings of the Nineteenth National Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press.

[29] Luo, X. Q. 2005. On coreference resolution performance metrics. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. pages 25-32.

[30] McCarthy, J. and W. Lehnert. 1995. Using decision trees for coreference resolution. In Proceedings of the Fourteenth International Conference on Artificial Intelligence, pages 1050–1055.

[31] McCray, A., A. Aronson, A. Browne, T. Rindflesch, A. Razi, and S. Srinivasan. 1993. UMLS knowledge for biomedical language processing. Bull. Med. Libr. Assoc. 1993; 81, 184-194.

[32] Mitkov, R. 2002. Anaphora Resolution. Longman.

[33] Morton, T. 1997. Coreference for NLP applications. In Proceedings ACL.

[34] MUC. MUC-7 Coreference Task Definition. [web page] http://www-nlpir.nist.gov/related_projects/muc/proceedings/co_task.html. Accessed 12 August, 2007

[35] Narayanaswamy, M. and K. E. Ravikumar. 2003. A biological named entity recognizer.

[36] Ng, V. and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In Proceedings of the ACL, pages 104-111.

[37] Ng, V. 2004. Learning noun phrase anaphoricity to improve conference resolution: issues in representation and optimization. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Barcelona, Spain.

[38] Ng, V. 2005. Machine learning for coreference resolution: from local classification to global ranking. In Proceedings of the 43rd Meeting of the Association for Computational Linguistics, pages 157–164, Ann Arbor,MI.

[39] Pedersen, T., A. Purandare, and A. Kulkarni. 2005. Name discrimination by clustering similar contexts. In Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics, pages 220–231, Mexico City.

[40] Popov, B., A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. 2003. Towards semantic web information extraction. Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003), Florida, USA.

[41] Quinlan, J.R. 1986. Induction of decision trees. Machine Learning, pages 81-106. Morgan Kaufmann.

[42] Quinlan, J. R. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

[43] J. Rennie and T. Jaakkola. 2006. Using term informativeness for named entity detection. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[44] Schilder, F. and C. Habel. 2001. From temporal expressions to temporal information: Semantic tagging of news messages. In Proceedings of ACL Workshop on Temporal and Spatial Information Processing. Toulouse, pages 65–72.

[45] Di Eugenio, B. and M. Glass. 2004. The kappa statistic: a second look. Computational Linguistics, 30(1): pages 95–101.

[46] Sibanda, T. 2006. Was the Patient Cured? Understanding semantic categories and their relationships in patient records. Master's thesis, Massachusetts Institute of Technology.

[47] Sleater, D. and D. Temperley. 1991. Parsing english with a link grammar. Technical Report CMU-CS-91-196, Carnegie Mellon University.

[48] Soon, W. M., H. T. Ng, and D. C. Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. Computational Linguistics, 27(4):521–544.

[49] Steven, M. and M. Greenwood. 2006. Learning information extraction patterns using WordNet.

[50] Stoyanov, V. and C. Cardie. 2006. Partially supervised coreference resolution for opinion summarization through structure rule learning. In Proceedings of EMNLP.

[51] Strube, M., S. Rapp, and C. Müller. 2002. The influence of minimum edit distance on reference resolution. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, pages 312-319.

[52] Sundheim, B. and N. Chinchor. 1995. Named entity task definition (version 2.1). In Proceedings of the Sixth Message Understanding Conference (MUC-6), pages 319- 332. Columbia, Maryland: Morgan Kauffman Publishers, Inc.

[53] Unified Medical Language System. 2006. [web page] http://www.nlm.nih.gov. Accessed 20 June, 2007

[54] van Deemter, K. and Rodger Kibble. 2000. On coreferring: coreference in MUC and related annotation schemes. Computational Linguistics, 26(4):629–637.

[55] Vilain, M., J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In Proceedings of the Sixth Message Understanding Conference (MUC-6), pages 45–52, San Francisco, CA. Morgan Kaufmann.

[56] Wilcoxon, F. 1945. Individual comparisons by ranking methods. Biometrics, 1, pages 80-83.

[57] Winkler, W. E. 1999. The state of record linkage and current research problems. statistics of income division, internal revenue service publication r99/04. Available from http://www.census.gov/srd/www/byname.html.

[58] Wu, S., T. H. Tsai, and W. L. Hsu. 2003. Domain event extraction and representation with domain ontology. In proceedings of the IJCAI-03 Workshop on Information Integration on the Web, Acapulco, Mexico, 2003, pages.33-38.

[59] Yakushiji, A., Y. Tateisi Y., Y. Miyao, and Tsujii J. 2001. Event extraction from biomedical papers using a full parser. Pacif. Symp. Biocomp., 6, pages 408-419

[60] Yang, X., G. Zhou, J. Su, and C. Tan. 2004a. An NP-cluster based approach to coreference resolution. In Proceedings of COLING 2004, Geneva, Switzerland.

[61] Yang, X., G. Zhou, J. Su, and C. Tan. 2004b. Improving noun phrase coreference resolution by matching strings. In Proceedings of the 1st International Joint Conference on Natural Language Processing, Hainan.

[62] Loper, E. 2002. Decision trees. [web page] www.cis.upenn.edu/~edloper/presentations/cse391_dtree.pdf

APPENDIX

A. Decision Trees for each Semantic Category

CONS

```
token-match-antec-base <= 0.6
|  prefix-match-greedy <= 0.5
|  |  noun-match-stingy <= 0.2: False (791.0/5.0)
|  |  noun-match-stingy > 0.2
|  |  |  normalized-token-match-anaph-base <= 0.2
|  |  |  |  normalized-edit-distance <= 18: True (4.0)
|  |  |  |  normalized-edit-distance > 18: False (4.0/1.0)
|  |  |  normalized-token-match-anaph-base > 0.2: False (112.0/6.0)
|  prefix-match-greedy > 0.5
|  |  prefix-match-antec <= 0.6
|  |  |  last-name-match = True: True (11.0)
|  |  |  last-name-match = False: False (7.0/1.0)
|  |  prefix-match-antec > 0.6: False (12.0)
token-match-antec-base > 0.6: True (49.0)

Number of Leaves  :     8
Size of the tree :      15
```

DIS

```
token-match-greedy-base <= 0.833333
|  normalized-token-match-anaph-base <= 0.285714
|  |  prefix-match-anaph <= 0.5
|  |  |  umls-concept-token-match-antec-base <= 0.5
|  |  |  |  sentence-match-stingy-base <= 0.52: False (25383.78/69.98)
|  |  |  |  sentence-match-stingy-base > 0.52
|  |  |  |  |  normalized-edit-distance <= 9
|  |  |  |  |  |  edit-distance <= 9: False (6.08/.0)
|  |  |  |  |  |  edit-distance > 9: True (4.01/.01)
|  |  |  |  |  normalized-edit-distance > 9: False (53.12/.01)
|  |  |  umls-concept-token-match-antec-base > 0.5
|  |  |  |  normalized-edit-distance <= 33
|  |  |  |  |  umls-type-match-anaph-base <= 0.75
|  |  |  |  |  |  umls-concept-token-match-anaph-base <= 0.333333: False (23.0)
|  |  |  |  |  |  umls-concept-token-match-anaph-base > 0.333333: True (6.0/1.0)
|  |  |  |  |  umls-type-match-anaph-base > 0.75: False (286.0/3.0)
|  |  |  |  normalized-edit-distance > 33: True (24.0/2.0)
|  |  prefix-match-anaph > 0.5
|  |  |  prefix-match-anaph <= 0.833333: True (8.0)
|  |  |  prefix-match-anaph > 0.833333: False (49.0/8.0)
|  normalized-token-match-anaph-base > 0.285714
|  |  umls-concept-match-greedy-base <= 0.333333
|  |  |  prefix-match-anaph <= 0.6: False (261.0/13.0)
|  |  |  prefix-match-anaph > 0.6
|  |  |  |  umls-type-match-antec-base <= 0.5
|  |  |  |  |  noun-match-anaph <= 0.25: True (5.0/2.0)
|  |  |  |  |  noun-match-anaph > 0.25: False (1.0)
|  |  |  |  umls-type-match-antec-base > 0.5: True (5.0/1.0)
|  |  umls-concept-match-greedy-base > 0.333333
|  |  |  umls-type-match-greedy-base <= 0.666667: False (18.0)
|  |  |  umls-type-match-greedy-base > 0.666667
|  |  |  |  noun-match-stingy <= 0.5
|  |  |  |  |  prefix-match-antec <= 0.666667
|  |  |  |  |  |  umls-concept-token-match-stingy-base <= 0.6
```

```
|  |  |  |  |  |  |  normalized-token-match-stingy-base <= 0.6
|  |  |  |  |  |  |  |  prefix-match-greedy <= 0.4: False (28.0/1.0)
|  |  |  |  |  |  |  |  prefix-match-greedy > 0.4
|  |  |  |  |  |  |  |  |  sentence-stop-words-removed-match-greedy-base <= 0.285714
|  |  |  |  |  |  |  |  |  |  plurality-match = True
|  |  |  |  |  |  |  |  |  |  |  token-match-anaph-base <= 0.4: True (9.93)
|  |  |  |  |  |  |  |  |  |  |  token-match-anaph-base > 0.4
|  |  |  |  |  |  |  |  |  |  |  |  noun-match-greedy <= 0.666667: True (9.0/3.0)
|  |  |  |  |  |  |  |  |  |  |  |  noun-match-greedy > 0.666667: False (22.0/2.0)
|  |  |  |  |  |  |  |  |  |  plurality-match = False: False (17.0/1.0)
|  |  |  |  |  |  |  |  |  sentence-stop-words-removed-match-greedy-base > 0.285714: True (4.07)
|  |  |  |  |  |  |  |  normalized-token-match-stingy-base > 0.6: True (6.0)
|  |  |  |  |  |  |  umls-concept-token-match-stingy-base > 0.6
|  |  |  |  |  |  |  |  word-distance <= 34: False (6.0/2.0)
|  |  |  |  |  |  |  |  word-distance > 34: True (24.0/1.0)
|  |  |  |  |  |  prefix-match-antec > 0.666667: False (24.0/2.0)
|  |  |  |  |  noun-match-stingy > 0.5: False (57.0/5.0)
token-match-greedy-base > 0.833333
|  word-distance <= 7
|  |  normalized-token-match-stingy-base <= 0.75: False (26.0/2.0)
|  |  normalized-token-match-stingy-base > 0.75
|  |  |  section-type-match = past-past: True (2.0)
|  |  |  section-type-match = present-present: True (4.0)
|  |  |  section-type-match = presenthistory-presenthistory: False (6.0)
|  |  |  section-type-match = discharge-discharge: True (1.0)
|  |  |  section-type-match = none-none: True (.0)
|  |  |  section-type-match = presenthistory-present: True (.0)
|  |  |  section-type-match = presenthistory-discharge: True (.0)
|  |  |  section-type-match = presenthistory-past: True (.0)
|  |  |  section-type-match = presenthistory-none: True (.0)
|  |  |  section-type-match = present-presenthistory: True (.0)
|  |  |  section-type-match = present-past: True (.0)
|  |  |  section-type-match = present-discharge: True (1.0)
|  |  |  section-type-match = past-presenthistory: True (.0)
|  |  |  section-type-match = past-present: True (.0)
|  |  |  section-type-match = past-discharge: True (.0)
|  |  |  section-type-match = past-none: True (.0)
|  |  |  section-type-match = discharge-present: True (.0)
|  |  |  section-type-match = discharge-presenthistory: True (.0)
|  |  |  section-type-match = discharge-past: True (.0)
|  |  |  section-type-match = discharge-none: True (.0)
|  |  |  section-type-match = none-present: True (.0)
|  |  |  section-type-match = none-discharge: True (.0)
|  |  |  section-type-match = none-presenthistory: True (.0)
|  |  |  section-type-match = none-past: True (.0)
|  word-distance > 7
|  |  umls-concept-token-match-greedy-base <= 0.25
|  |  |  edit-distance <= 15: False (18.0/2.0)
|  |  |  edit-distance > 15: True (1.0/2.0)
|  |  umls-concept-token-match-greedy-base > 0.25
|  |  |  normalized-token-match-anaph-base <= 0.5
|  |  |  |  presence <= 0
|  |  |  |  |  umls-type-match-anaph-base <= 0.75: True (8.0/1.0)
|  |  |  |  |  umls-type-match-anaph-base > 0.75
|  |  |  |  |  |  umls-type-base = patf: True (6.05/1.05)
|  |  |  |  |  |  umls-type-base = dsyn: False (13.32/4.0)
|  |  |  |  |  |  umls-type-base = mobd: False (.0)
|  |  |  |  |  |  umls-type-base = comd: False (.0)
|  |  |  |  |  |  umls-type-base = cgab: False (.0)
|  |  |  |  |  |  umls-type-base = acab: False (.0)
|  |  |  |  |  |  umls-type-base = inpo: False (3.63)
|  |  |  |  |  |  umls-type-base = anab: False (.0)
|  |  |  |  |  |  umls-type-base = virs: False (.0)
|  |  |  |  |  |  umls-type-base = bact: False (.0)
|  |  |  |  |  |  umls-type-base = neop: False (.0)
|  |  |  |  |  |  umls-type-base = topp: False (.0)
```

```
|  |  |  |  |  |  umls-type-base = medd: False (.0)
|  |  |  |  |  |  umls-type-base = strd: False (.0)
|  |  |  |  |  |  umls-type-base = phsu: False (.0)
|  |  |  |  |  |  umls-type-base = bodm: False (.0)
|  |  |  |  |  |  umls-type-base = antb: False (.0)
|  |  |  |  |  |  umls-type-base = lbpr: False (.0)
|  |  |  |  |  |  umls-type-base = diap: False (.0)
|  |  |  |  |  |  umls-type-base = clna: False (.0)
|  |  |  |  |  |  umls-type-base = orga: False (.0)
|  |  |  |  |  |  umls-type-base = lbtr: False (.0)
|  |  |  |  |  |  umls-type-base = fndg: False (.0)
|  |  |  |  |  |  umls-type-base = sosy: False (.0)
|  |  |  |  |  |  umls-type-base = bmod: False (.0)
|  |  |  |  |  |  umls-type-base = prog: False (.0)
|  |  |  |  |  |  umls-type-base = hops: False (.0)
|  |  |  |  |  presence > 0: True (62.0/13.0)
|  |  |  |  normalized-token-match-anaph-base > 0.5
|  |  |  |  |  normalized-token-match-anaph-base <= 0.75: True (4.0)
|  |  |  |  |  normalized-token-match-anaph-base > 0.75
|  |  |  |  |  |  plurality-match = True
|  |  |  |  |  |  |  word-distance <= 27
|  |  |  |  |  |  |  |  entity-distance <= 1: True (1.0)
|  |  |  |  |  |  |  |  entity-distance > 1
|  |  |  |  |  |  |  |  |  section-type-match = past-past: True (1.0)
|  |  |  |  |  |  |  |  |  section-type-match = present-present
|  |  |  |  |  |  |  |  |  |  sentence-entity-allcategory-greedy <= 0.571429: True (5.0)
|  |  |  |  |  |  |  |  |  |  sentence-entity-allcategory-greedy > 0.571429: False (5.0/2.0)
|  |  |  |  |  |  |  |  |  section-type-match = presenthistory-presenthistory: False (12.0/1.0)
|  |  |  |  |  |  |  |  |  section-type-match = discharge-discharge: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = none-none: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = presenthistory-present: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = presenthistory-discharge: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = presenthistory-past: True (2.0)
|  |  |  |  |  |  |  |  |  section-type-match = presenthistory-none: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = present-presenthistory: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = present-past: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = present-discharge: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = past-presenthistory: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = past-present: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = past-discharge: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = past-none: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = discharge-present: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = discharge-presenthistory: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = discharge-past: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = discharge-none: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = none-present: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = none-discharge: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = none-presenthistory: False (.0)
|  |  |  |  |  |  |  |  |  section-type-match = none-past: False (.0)
|  |  |  |  |  |  |  word-distance > 27: True (567.0/51.0)
|  |  |  |  |  |  plurality-match = False
|  |  |  |  |  |  |  umls-type-match-stingy-base <= 0.75: False (9.0/2.0)
|  |  |  |  |  |  |  umls-type-match-stingy-base > 0.75: True (1.0)

Number of Leaves  :     112
Size of the tree :      154
```

MED

```
edit-distance <= 1: True (796.0/37.0)
edit-distance > 1
|  token-match-greedy-base <= 0.5
|  |  prefix-match-antec <= 0: False (50083.0/13.0)
```

```
|   |  prefix-match-antec > 0
|   |  |  edit-distance <= 5
|   |  |  |  edit-distance <= 2: False (19.0/3.0)
|   |  |  |  edit-distance > 2: True (5.0/9.0)
|   |  |  edit-distance > 5
|   |  |  |  normalized-edit-distance <= 22
|   |  |  |  |  prefix-match-greedy <= 0.8: False (216.0/3.0)
|   |  |  |  |  prefix-match-greedy > 0.8
|   |  |  |  |  |  edit-distance <= 15
|   |  |  |  |  |  |  normalized-edit-distance <= 10: False (42.0/4.0)
|   |  |  |  |  |  |  normalized-edit-distance > 10
|   |  |  |  |  |  |  |  umls-type-match-stingy-base <= 0.75
|   |  |  |  |  |  |  |  |  date-defaultfalse = True: True (5.33/2.07)
|   |  |  |  |  |  |  |  |  date-defaultfalse = False: False (14.67/1.73)
|   |  |  |  |  |  |  |  umls-type-match-stingy-base > 0.75: True (9.0/2.0)
|   |  |  |  |  |  edit-distance > 15: False (16.0)
|   |  |  |  normalized-edit-distance > 22
|   |  |  |  |  umls-concept-match-antec-base <= 0: False (14.0/2.0)
|   |  |  |  |  umls-concept-match-antec-base > 0: True (7.0)
|   token-match-greedy-base > 0.5
|   |  noun-match-greedy <= 0.666667
|   |  |  umls-type-match-stingy-base <= 0.75: True (4.0/1.0)
|   |  |  umls-type-match-stingy-base > 0.75: False (13.0/1.0)
|   |  noun-match-greedy > 0.666667
|   |  |  section-type-match = past-past: False (1.0)
|   |  |  section-type-match = present-present
|   |  |  |  normalized-edit-distance <= 9
|   |  |  |  |  date-defaultfalse = True: False (11.57/4.71)
|   |  |  |  |  date-defaultfalse = False: True (42.43/9.14)
|   |  |  |  normalized-edit-distance > 9
|   |  |  |  |  noun-match-stingy <= 0.666667
|   |  |  |  |  |  sentence-distance <= 6: True (4.0/1.0)
|   |  |  |  |  |  sentence-distance > 6: False (26.0/1.0)
|   |  |  |  |  noun-match-stingy > 0.666667: True (11.0/3.0)
|   |  |  section-type-match = presenthistory-presenthistory: True (3.0/1.0)
|   |  |  section-type-match = discharge-discharge: True (4.0/1.0)
|   |  |  section-type-match = none-none: True (.0)
|   |  |  section-type-match = presenthistory-present
|   |  |  |  entity-distance <= 14: True (15.0)
|   |  |  |  entity-distance > 14: False (18.0/5.0)
|   |  |  section-type-match = presenthistory-discharge: True (1.0)
|   |  |  section-type-match = presenthistory-past: True (2.0)
|   |  |  section-type-match = presenthistory-none: True (.0)
|   |  |  section-type-match = present-presenthistory: True (.0)
|   |  |  section-type-match = present-past: True (.0)
|   |  |  section-type-match = present-discharge
|   |  |  |  sentence-match-greedy-base <= 0.426966
|   |  |  |  |  noun-match-stingy <= 0.5
|   |  |  |  |  |  umls-type-match-stingy-base <= 0.25: True (1.0/2.0)
|   |  |  |  |  |  umls-type-match-stingy-base > 0.25: False (8.0)
|   |  |  |  |  noun-match-stingy > 0.5: True (4.0)
|   |  |  |  sentence-match-greedy-base > 0.426966: True (19.0)
|   |  |  section-type-match = past-presenthistory: True (.0)
|   |  |  section-type-match = past-present: True (5.0)
|   |  |  section-type-match = past-discharge: True (1.0)
|   |  |  section-type-match = past-none: True (.0)
|   |  |  section-type-match = discharge-present: True (.0)
|   |  |  section-type-match = discharge-presenthistory: True (.0)
|   |  |  section-type-match = discharge-past: True (.0)
|   |  |  section-type-match = discharge-none: True (.0)
|   |  |  section-type-match = none-present: True (3.0)
|   |  |  section-type-match = none-discharge: True (.0)
|   |  |  section-type-match = none-presenthistory: True (.0)
|   |  |  section-type-match = none-past: True (.0)
```

Number of Leaves : 46
Size of the tree : 69

SYMP

```
normalized-edit-distance <= 2
|  presence <= 0: False (37.0/6.0)
|  presence > 0
|  |  umls-concept-match-anaph-base <= 0: False (23.0/5.0)
|  |  umls-concept-match-anaph-base > 0
|  |  |  entity-distance <= 8: True (123.0/6.0)
|  |  |  entity-distance > 8
|  |  |  |  sentence-match-antec-base <= 0.196429
|  |  |  |  |  sentence-match-antec-base <= .074074: True (16.0/2.0)
|  |  |  |  |  sentence-match-antec-base > .074074: False (15.0/2.0)
|  |  |  |  sentence-match-antec-base > 0.196429: True (41.0/4.0)
normalized-edit-distance > 2
|  normalized-token-match-stingy-base <= 0.166667
|  |  prefix-match-greedy <= 0.666667: False (8714.0/18.0)
|  |  prefix-match-greedy > 0.666667
|  |  |  umls-type-match-anaph-base <= 0.666667: False (12.0)
|  |  |  umls-type-match-anaph-base > 0.666667: True (16.0/5.0)
|  normalized-token-match-stingy-base > 0.166667
|  |  normalized-token-match-stingy-base <= 0.5
|  |  |  section-distance <= 11
|  |  |  |  prefix-match-anaph <= 0.5
|  |  |  |  |  presence <= 1
|  |  |  |  |  |  token-match-antec-base <= 0.2: True (9.0/2.0)
|  |  |  |  |  |  token-match-antec-base > 0.2
|  |  |  |  |  |  |  umls-concept-match-greedy-base <= 0
|  |  |  |  |  |  |  |  presence <= 0: False (9.0)
|  |  |  |  |  |  |  |  presence > 0
|  |  |  |  |  |  |  |  |  entity-distance <= 1: False (5.0)
|  |  |  |  |  |  |  |  |  entity-distance > 1
|  |  |  |  |  |  |  |  |  |  date-defaultunknown = True: True (22.39/2.44)
|  |  |  |  |  |  |  |  |  |  date-defaultunknown = False
|  |  |  |  |  |  |  |  |  |  |  sentence-stop-words-removed-match-anaph-base <= 0.142857: True (4.06/1.28)
|  |  |  |  |  |  |  |  |  |  |  sentence-stop-words-removed-match-anaph-base > 0.142857: False (4.56/0.28)
|  |  |  |  |  |  |  umls-concept-match-greedy-base > 0
|  |  |  |  |  |  |  |  entity-distance-allcategory <= 172
|  |  |  |  |  |  |  |  |  umls-concept-match-greedy-base <= 0.5: False (4.0)
|  |  |  |  |  |  |  |  |  umls-concept-match-greedy-base > 0.5
|  |  |  |  |  |  |  |  |  |  plurality-match = True
|  |  |  |  |  |  |  |  |  |  |  edit-distance <= 10: False (37.0/4.0)
|  |  |  |  |  |  |  |  |  |  |  edit-distance > 10
|  |  |  |  |  |  |  |  |  |  |  |  sentence-match-anaph-base <= 0.484848: True (8.27/1.0)
|  |  |  |  |  |  |  |  |  |  |  |  sentence-match-anaph-base > 0.484848: False (4.73/1.73)
|  |  |  |  |  |  |  |  |  |  plurality-match = False: False (16.0)
|  |  |  |  |  |  |  |  entity-distance-allcategory > 172: True (1.0/2.0)
|  |  |  |  |  |  presence > 1: False (17.0)
|  |  |  |  |  prefix-match-anaph > 0.5
|  |  |  |  |  |  sentence-match-stingy-base <= 0.291667
|  |  |  |  |  |  |  presence <= 0: False (8.0/2.0)
|  |  |  |  |  |  |  presence > 0
|  |  |  |  |  |  |  |  edit-distance <= 9
|  |  |  |  |  |  |  |  |  date-defaultunknown = True
|  |  |  |  |  |  |  |  |  |  noun-match-anaph <= 0.666667: True (4.0/1.0)
|  |  |  |  |  |  |  |  |  |  noun-match-anaph > 0.666667: False (9.59/1.0)
|  |  |  |  |  |  |  |  |  date-defaultunknown = False
|  |  |  |  |  |  |  |  |  |  token-match-anaph-base <= 0.666667: False (4.0)
|  |  |  |  |  |  |  |  |  |  token-match-anaph-base > 0.666667: True (5.41/0.41)
|  |  |  |  |  |  |  |  edit-distance > 9: True (11.0/1.0)
|  |  |  |  |  sentence-match-stingy-base > 0.291667: True (9.0)
```

```
|   |   |   section-distance > 11: False (22.0)
|   |   normalized-token-match-stingy-base > 0.5
|   |   |   umls-concept-match-anaph-base <= 0.5: False (4.0/1.0)
|   |   |   umls-concept-match-anaph-base > 0.5: True (11.0)


Number of Leaves  :     32
Size of the tree :      63
```

TEST

```
token-match-anaph-base <= 0.5: False (53703.0/4.0)
token-match-anaph-base > 0.5
|   right-entity-allcategory-greedy <= 0.888889
|   |   word-distance <= 6
|   |   |   token-match-greedy-base <= 0.75: False (9.0)
|   |   |   token-match-greedy-base > 0.75: True (36.22/9.69)
|   |   word-distance > 6
|   |   |   date-defaultunknown = True
|   |   |   |   umls-type-base = patf: False (.0)
|   |   |   |   umls-type-base = dsyn: False (.0)
|   |   |   |   umls-type-base = mobd: False (.0)
|   |   |   |   umls-type-base = comd: False (.0)
|   |   |   |   umls-type-base = cgab: False (.0)
|   |   |   |   umls-type-base = acab: False (.0)
|   |   |   |   umls-type-base = inpo: False (.0)
|   |   |   |   umls-type-base = anab: False (.0)
|   |   |   |   umls-type-base = virs: False (.0)
|   |   |   |   umls-type-base = bact: False (.0)
|   |   |   |   umls-type-base = neop: False (.0)
|   |   |   |   umls-type-base = topp: False (.0)
|   |   |   |   umls-type-base = medd: False (.0)
|   |   |   |   umls-type-base = strd: False (.0)
|   |   |   |   umls-type-base = phsu: False (.0)
|   |   |   |   umls-type-base = bodm: False (.0)
|   |   |   |   umls-type-base = antb: False (.0)
|   |   |   |   umls-type-base = lbpr: False (123.81/27.03)
|   |   |   |   umls-type-base = diap
|   |   |   |   |   normalized-token-match-anaph-base <= 0.666667: False (8.53/0.7)
|   |   |   |   |   normalized-token-match-anaph-base > 0.666667
|   |   |   |   |   |   noun-match-stingy <= 0.666667: True (15.13/2.74)
|   |   |   |   |   |   noun-match-stingy > 0.666667
|   |   |   |   |   |   |   umls-concept-match-anaph-base <= 0.5: True (7.37/0.9)
|   |   |   |   |   |   |   umls-concept-match-anaph-base > 0.5
|   |   |   |   |   |   |   |   sentence-stop-words-removed-match-anaph-base <= .05: False (70.4/22.84)
|   |   |   |   |   |   |   |   sentence-stop-words-removed-match-anaph-base > .05: True (17.62/6.35)
|   |   |   |   umls-type-base = clna: False (67.43/9.95)
|   |   |   |   umls-type-base = orga: False (12.71/1.85)
|   |   |   |   umls-type-base = lbtr: False (.0)
|   |   |   |   umls-type-base = fndg: False (.0)
|   |   |   |   umls-type-base = sosy: False (.0)
|   |   |   |   umls-type-base = bmod: False (.0)
|   |   |   |   umls-type-base = prog: False (.0)
|   |   |   |   umls-type-base = hops: False (.0)
|   |   |   date-defaultunknown = False: False (545.8/43.7)
|   right-entity-allcategory-greedy > 0.888889
|   |   date-defaultfalse = True: True (62.33/12.53)
|   |   date-defaultfalse = False
|   |   |   left-entity-allcategory-greedy <= 0.333333
|   |   |   |   noun-match-stingy <= 0.5: True (4.25/1.26)
|   |   |   |   noun-match-stingy > 0.5: False (30.32/6.29)
|   |   |   left-entity-allcategory-greedy > 0.333333
|   |   |   |   sentence-match-anaph-base <= 0.642857: False (4.73/1.88)
```

| | | | sentence-match-anaph-base > 0.642857: True (5.35/.04)

Number of Leaves : 40
Size of the tree : 54