# EFFICIENT INFORMATION EXTRACTION USING STATISTICAL RELATIONAL LEARNING

BY

JOSE MANUEL PICADO LEIVA

A Thesis Submitted to the Graduate Faculty of

WAKE FOREST UNIVERSITY GRADUATE SCHOOL OF ARTS AND SCIENCES

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

Computer Science

May 2013

Winston-Salem, North Carolina

Approved By:

Sriraam Natarajan, Ph.D., Advisor

David J. John, Ph.D., Chair

William Turkett, Ph.D.

# Dedication

To my family. Thanks for all your love, support, and courage.

# Acknowledgements

First, I would like to express my deepest gratitude and appreciation to my advisor, Professor Sriraam Natarajan. Your guidance, patience, and immense knowledge allowed me to reach the point where I am now. Your enthusiasm for research motivated me to work hard and made me become passionate for research as well. Thanks for giving me the opportunity to work with you, as well as for supporting me and preparing me for the next steps of my academic life.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor David John and Professor William Turkett, for their support, feedback, and insightful comments.

This thesis would not have been possible without the work of many people. Thanks to Tushar Khot, Dr. Kristian Kersting, Dr. Jude Shavlik, Dr. Christopher Ré, Dr. Vitor Santos Costa, Dr. David Page, and Dr. Michael Caldwell for all your effort, help, and feedback in producing the content for this thesis. I would also like to thank the STRAIT team members for sharing all meetings, talks, and conversations we had in our group.

I would like to thank the Department of Computer Science for giving me the opportunity to pursue a graduate degree at Wake Forest University. Thanks to the entire faculty for the valuable education inside and outside of classes. Thanks to all other graduate students with whom I shared this experience.

I would like to express my gratitude to Professor Daniel Cañas for his support before and during my stay in the United States, as well as his hospitality since the first day I came to the country. Besides an excellent professor, I consider you my friend and enjoyed all the conversations we had throughout these years.

Finally, I must thank my family for the unconditional love and support during my stay in the United States. I owe you everything I am, and will always be infinitely grateful. Thanks for your visits, for all your messages, and for the daily phone calls that made me feel as if I was never away from home. Thanks for encouraging me to set myself high goals and providing the support to achieve them.

# Table of Contents

List of Figures ..... vi

List of Tables ..... vii

Abstract ..... viii

Chapter 1 Introduction ..... 1
1.1 Challenges of Information Extraction ..... 1
1.2 Contributions ..... 2
1.2.1 Language-Independent Approach ..... 2
1.2.2 Application of Information Extraction ..... 3
1.3 Thesis Outline ..... 4

Chapter 2 Background ..... 5
2.1 Information Extraction ..... 5
2.1.1 Introduction ..... 5
2.1.2 Supervised Learning Approach ..... 6
2.1.3 Weak Supervision Approach ..... 8
2.1.4 Biomedical Information Extraction ..... 9
2.2 Statistical Relational Learning ..... 10
2.2.1 Introduction ..... 10
2.2.2 Relational Functional Gradient Boosting ..... 11
2.2.3 Markov Logic Networks ..... 12

Chapter 3 Exploiting Commonsense Knowledge for Weak Supervision ..... 15
3.1 Generating Examples Using Commonsense Knowledge ..... 15
3.1.1 Knowledge in Information Extraction ..... 17
3.1.2 Markov Logic for Knowledge Representation ..... 17
3.1.3 Weak Supervision Phase ..... 18
3.2 Learning for Information Extraction Using Weakly Supervised Examples 20
3.2.1 Information Extraction Phase ..... 20
3.3 Experimental Results ..... 21
3.3.1 Relation Extraction ..... 22
3.3.2 Text Categorization ..... 25

Chapter 4 Text-Based Evaluation of Adverse Drug Events Discovery . . . . . 29
4.1 Adverse Drug Events Discovery . . . . . 29
4.2 Evaluating Adverse Drug Events Using Natural Language Processing 30
4.2.1 Approach Overview . . . . . 30
4.2.2 String Similarity Phase . . . . . 31
4.2.3 Semantic Relation Extraction Phase . . . . . 31
4.3 Experimental Results . . . . . 33
4.3.1 Task Description . . . . . 34
4.3.2 Dataset . . . . . 34
4.3.3 Setup . . . . . 34
4.3.4 Results . . . . . 35
4.4 Discussion . . . . . 35

Chapter 5 Conclusion . . . . . 39
5.1 Future Work . . . . . 40

Bibliography . . . . . 41

Curriculum Vitae . . . . . 46

# List of Figures


2.1 Information extraction. . . . . 5

3.1 Sample text annotated with entities and relations. . . . . 16
3.2 Steps involved in creation of weakly supervised examples. . . . . 19
3.3 Steps involved in learning using probabilistic examples. . . . . 20
3.4 Results of predicting *winners* and *losers* in NFL domain, varying gold standard examples and keeping weakly supervised examples constant. 24
3.5 Results of predicting *winners* and *losers* in NFL domain, varying weakly supervised examples and keeping gold standard examples constant. . 26
3.6 Results of document classification. . . . . 28

4.1 Steps involved in the evaluation of adverse drug events (ADEs). . . . 31
4.2 ROC curve for identification of ADEs using OMOP ground truth. . . 36

# List of Tables

2.1 Example annotated corpus. . . . . 7

3.1 A sample of WMLN clauses used for NFL task. . . . . 18
3.2 All WMLN clauses used for NFL task with their corresponding weights. 23

4.1 A sample of the relation extraction knowledge. . . . . 33
4.2 AUCROC and p-values for the three settings of ADE evaluation. . . . 35
4.3 Examples of ADE pairs. . . . . 37

# Abstract

Information extraction has gained significant importance due to the dramatic increase of information stored in the form of natural language text. In this thesis we explore a machine learning-based approach to support a natural language processing (NLP) algorithm, and an application of information extraction. One of the challenges of learning-based approaches is the requirement of human annotated examples. Current successful approaches alleviate this problem by employing some form of distant supervision. In this work, we take a different approach – we create weakly supervised examples for relations by using commonsense knowledge. The key innovation is that this commonsense knowledge is completely independent of the natural language text. This helps when learning the full model for information extraction as against simply learning the parameters of a known model. We demonstrate on two domains that this form of weak supervision yields superior results when learning structure compared to simply using the gold standard labels. In the second part of this thesis, we consider the problem of Adverse Drug Events (ADEs) discovery. Several methods have been proposed for ADE discovery, exploiting various information sources such as health data, social network data, and scientific literature. We propose a NLP-based method that exploits scientific literature to quantitatively evaluate proposed ADEs. We validate our approach on a common ADE dataset, where we find better agreement than state-of-the-art ADE discovery methods.

# Chapter 1: Introduction

During the past decades, there has been a dramatic increase in the amount of information stored in digital archives and the World Wide Web. It is estimated that there exist around 30 trillion individual pages on the Web, and the number is constantly growing [1]. While there are existing sources that contain structured information, it is well known that most of the information is stored in the form of natural language text. Textual information is a form of unstructured data, which is impossible for machines to query, organize and analyze. Because of this, there has been a growing interest in the task of automatically extracting information from text.

Information extraction consists in finding facts from text. The search engine Google has been estimated to process over 1 billion search requests every day, many of them coming from people looking for facts about all kinds of topics. The goal of information extraction is to organize information so that it is useful to people and to make it semantically understandable so that it can be processed by computer algorithms [20].

## 1.1 Challenges of Information Extraction

Information extraction is a subfield of Natural Language Processing (NLP). Some of the core challenges of information extraction are [27]:

- Linguistics analyses are highly ambiguous, and have to deal with contradictions and errors.
- In natural language text, there are many variations of expressing the same meaning.
- Writing styles depend on the source the text is taken from, e.g., informational news, sports news, blog posts and scientific papers have different styles.
- Documents are heterogeneous, so they may contain information about multiple topics, e.g., sports, finance, science.
- In many cases, facts are not given explicitly in the text, and they must be inferred from other facts.
- Information may need to be combined across several sentences or even documents.

Let us consider the following text snippet, from which facts can be identified using information extraction techniques.

> Steve Jobs co-founded Apple Inc. in 1976. After a power struggle with the board of directors in 1985, Jobs was fired from Apple and founded NeXT. Thirteen years later, he returned to Apple and became CEO of the company.

From this text we can get the following facts:

| **Entity** | **Relation** | **Entity** | **Year** |
|---|---|---|---|
| Steve Jobs | founded | Apple Inc. | 1976 |
| Steve Jobs | founded | NeXT | 1985 |
| Steve Jobs | CEO-of | Apple Inc. | 1998 |

Note that we extracted facts from predefined relations, such as *founded* and *CEO-of*, while ignoring information such as Jobs being fired. In the third sentence, the word "he" refers to Jobs, so coreference resolution must be performed. Finally, Jobs becoming CEO of Apple in 1998 is very difficult to be automatically extracted because it is not explicitly mentioned in the text, but should be inferred using the phrase "thirteen years later".

## 1.2 Contributions

As can be seen, information extraction is not an easy task. Consequently, several approaches have been proposed in the past. In this thesis, we explore an approach to information extraction and an application of information extraction. Our first contribution is a method that uses a language-independent technique to create examples that support an NLP algorithm. Our second contribution is using a text-based NLP algorithm to quantitatively evaluate extracted facts from other machine learning systems. If applied together, both contributions can be complementary. However, this combination remains an interesting future research direction.

### 1.2.1 Language-Independent Approach

In the past, several information extraction algorithms based on machine learning have been proposed [11, 21, 33, 38, 45]. One of the challenges of these algorithms is the requirement of human annotated examples. Human annotation is expensive, as it very time consuming and tedious. Several approaches have alleviated this problem by employing some form of distant supervision [21], i.e., consider knowledge bases such as Freebase as a source of supervision to create more examples. However, an important property of such methods is that the quality of these labels is crucially dependent on the heuristic that is used to map the relations to the knowledge base. Consequently, there have been several approaches that aim to improve the quality of

these labels, ranging from casting the problem as multi-instance learning [33, 13, 39] to predicting text patterns to remove wrong labels [40]. However these approaches still have some disadvantages. Multi-instance learning does not work when wrong labels are assigned to entity pairs that appear only once in a corpus. On the other hand, predicting text patterns for some relations may be difficult, making it impossible to remove wrong labels.

We take a different approach for creating more examples to the supervised learner based on *weak supervision* [6]. We propose to use commonsense knowledge to create sets of entities that are "potential" relations. This commonsense knowledge is written by a domain expert in a probabilistic logic formalism called as *Markov Logic Networks* (MLN) [8], and is **language-independent** – it is completely independent of the natural language text. The algorithm then learns the parameters of these MLN clauses (we call them as *world MLN* – WMLN – to reflect that they are non-linguistic models) from some knowledge base. Unlabeled text is parsed through some entity resolution parser to identify potential entities. Then these entities are provided as queries to the world MLN which uses data from non-NLP sources to then predict the posterior probability of relations between these entities. These predicted relations become the probabilistic (weakly supervised) examples. Once the examples are created, we employ a Statistical Relational NLP algorithm to perform information extraction.

Our hypothesis – which we verify empirically – is that the use of world knowledge will help in learning from natural language text. This is particularly true when there is a need to learn a model without any prior structure (a CRF, MRF, or MLN), as is typically done for information extraction, since the number of examples needed to learn the model can be very large. These weakly supervised examples can then augment the human annotated examples to improve the quality of the learned models.

### 1.2.2 Application of Information Extraction

When searching for information extraction systems, it is important to quantitatively evaluate and compare very different methodologies. Our second problem consists in evaluating facts extracted by other intelligent systems using an algorithm that is based on text patterns and similarities. As this algorithm is dependent on the natural language text, we refer to it as **language-dependent**. Particularly, we focus on biomedical information extraction by evaluating Adverse Drug Events (ADEs) extracted by existing methods. Several ADE discovery methods exist, such as the one proposed by White et al. [43] that uncovers potential ADEs from internet search engine queries, and work by Page et al. [25] that contructs rules defining the ADEs from ICD codes, laboratory values, vital signs, procedures, physician defined lexicons, etc. all in a fashion not biased by previously-defined health outcomes of interest (HOIs).

We propose a novel evaluation method based on applying state-of-the-art text and NLP algorithms to abstracts from the scientific literature in order to score the

evidence for proposed ADEs. We employ syntactic and semantic measures captured in an MLN [8] to assign scores to the proposed ADEs. We evaluate our NLP approach on a common set of ADEs defined by the Observational Medical Outcomes Partnership (OMOP). We show that our approach has high but not perfect agreement with OMOP's ground truth. We then look more closely at where our system's results disagree with OMOP's ground truth. In some instances this investigation reveals probable errors in OMOP's ground truth, owing either to OMOP's high standard of evidence (drug labels) for ADEs or to discoveries occurring after OMOP initiation. In other instances this investigation reveals shortcomings in our current approach that point to directions for further research.

## 1.3 Thesis Outline

The rest of the thesis is organized as follows: In Chapter 2, we provide an overview of the background in information extraction, existing information extraction approaches, and Statistical Relational Learning. In Chapter 3, we present our first task of information extraction to support an NLP algorithm. We present our proposed approach along with experimental results where we show the effectiveness of our approach. In Chapter 4, we present our second task of using information extraction for biomedical event extraction. After describing our approach, we present our experimental results. Finally, in Chapter 5, we present our conclusions and outline further directions in research.

# Chapter 2: Background

In this chapter, we provide a brief introduction to Information Extraction (IE) and Statistical Relational Learning (SRL). We discuss previous approaches that have been proposed for IE, focusing on supervised and weakly-supervised learning methods. We then provide an introduction to SRL, discuss how SRL is naturally applicable to IE, and introduce two SRL methods that we employ in the following chapters.

## 2.1 Information Extraction

### 2.1.1 Introduction

Information Extraction is the process of automatically extracting structured information from unstructured data, where unstructured data consists of machine-readable documents. The goal is to extract relevant information from these documents and convert them to structured format, such as a knowledge base. Information extraction itself comprises many subtasks, such as named entity recognition, coreference resolution, relation extraction and text categorization. In this thesis we focus on the latter two tasks.

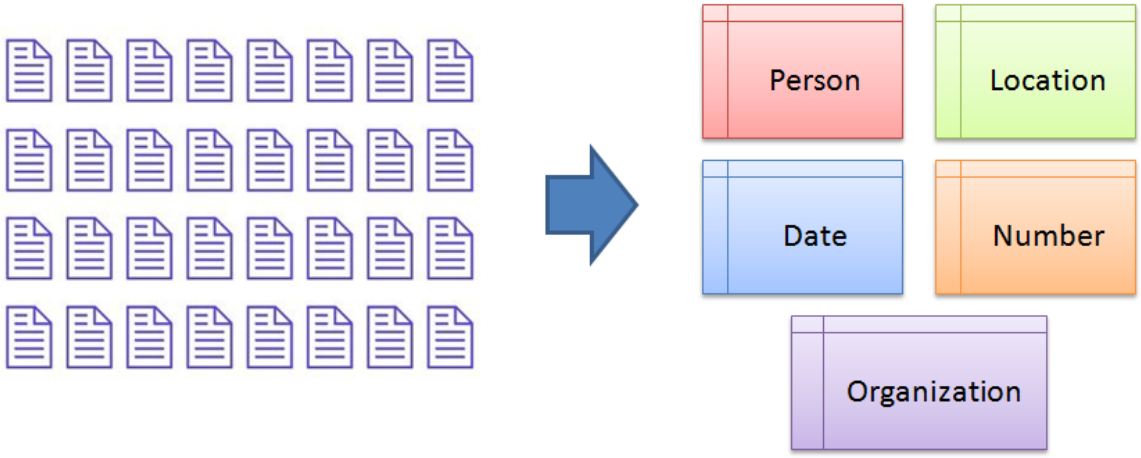


Figure 2.1: Information extraction.

The task of *relation extraction* denotes learning relationships between entities of interest. For instance, consider the sentence

> Steve Jobs was an American entrepreneur and inventor, best known as the co-founder of Apple Inc.

From this text, as an example, we can extract the relation *founder* between the entities *Steve Jobs* and *Apple Inc.* Although relations can exist between more than two entities, in this work, we focus on binary relations. We assume that the entities are recognized in a preprocessing step. We also limit our work to extracting only one relation per sentence, but this can be easily extended to multiple relations.

The task of *text categorization* denotes assigning text to one or more classes or categories. Text can be in the form of documents (document classification) or sentences (sentence classification), and it should be classified according to their subject or other attributes of interest.

Much research have focused on Information Extraction. Several evaluation programs have been initiated to measure the progress of research, such as: Message Understanding Conference (MUC)[1], which explored military themes through seven conferences; Automatic Content Extraction (ACE)[2], which focused on automatic processing of human language from newswires, broadcast conversations and weblogs; and Text Analysis Conference (TAC)[3], which explored several tracks such as entity linking and slot filling.

### 2.1.2 Supervised Learning Approach

Different approaches have been applied to the task of information extraction [21, 33, 38, 40, 45, 46]. One of the popular approaches is supervised learning, where a set of training examples $< \mathbf{x}, f(\mathbf{x}) >$ is given and the goal is to find a good approximation of $f$. The vector $\mathbf{x}$ consists of a set of features that describe each example and $f(\mathbf{x})$ is the label. In natural language processing tasks, features usually consist of lexical, syntactic, semantic, and contextual properties of the text, and $f$ specifies the desired output, depending on the task (i.e., relations between entities, document classification, sentence classification).

In supervised learning approaches [11, 38, 45, 46], an annotated training corpus is required to learn a model. For instance, a typical training corpus in supervised relation extraction consists of hand-labeled sentences, where mentions of entities are annotated and relations between these entities are specified. Each pair of entities and a specified relation between those entities become an example, in this case called *gold-standard examples*. For example, consider the annotated corpus in Table 2.1. As can be seen, sentences are annotated with mentions of entities, and the relations holding between these entities are provided. The goal of supervised classifiers is to be able to extract relation mentions between two entities in the test set.

Several supervised learning methods have been used in the task of relation extraction. Zhou et al. [45] constructed a feature-based relation extraction system using

[1] http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html

[2] http://www.itl.nist.gov/iad/mig/tests/ace/

[3] http://www.nist.gov/tac/about/index.html

| Text | Relation |
|---|---|
| Steve Jobs was an American entrepreneur and inventor, best known as the co-founder of Apple Inc. | founded(Steve Jobs, Apple Inc.) |
| Microsoft was founded by Bill Gates in 1975. | founded(Bill Gates, Microsoft) |

Table 2.1: Example annotated corpus.

Support Vector Machines (SVMs). They used lexical, syntactic and semantic features, and explored the effectiveness of each of them. They evaluated their system on the ACE corpus, specifically on the ACE Relation Detection and Characterization (RDC) task, and got competitive results compared to the previous relation extraction systems that were used for this task. They reported precision, recall and F-measure to compare the contribution of different features. Entity type features (e.g., PERSON, LOCATION, ORGANIZATION) and chunking features, which are text phrases identfied through a shallow parse, were very useful and gave the biggest improvements of performance. On the other hand, mention level features (e.g., NAME, NOMINAL, PRONOUN), semantic features (e.g. country names and personal relative trigger words), and structured features (e.g., dependency and parse trees) just gave a limited increase in performance.

Surdeanu and Ciaramita [38] constructed a feature-based system for entity and relation extraction, which used a combination of variants of the Perceptron algorithm [34]. They proposed a joint approach, where entity mentions are first classified, and then relation mentions are detected using as input the output from the first step. Although this was a simple and novel approach, and got competitive results in the ACE Entity Mention Detection (EMD) and Relation Mention Detection (RMD) tasks, it only uses partial syntactic information and ignores semantic and contextual features.

Zhou et al. [46] proposed a different approach for relation extraction. Instead of focusing on a feature-based system, they proposed a context-sensitive convolution tree kernel for this task. Kernel-based methods, contrary to feature-based methods, can exploit implicit structured information by exploring various features in a high dimensional space. By combining flat features and structured features (i.e., parse trees), they were able to get competitive results in the ACE RDC task.

Although these methods obtain reasonable precision and recall, supervised information extraction approaches suffer from a major problem: generating hand-labeled examples is expensive. Thus, these approaches are limited in scalability, which is a key problem when trying to perform information extraction in huge text repositories such as the World Wide Web. For this reason, there has been a growing interest to find ways of creating bigger amounts of training data in a less expensive way.

### 2.1.3 Weak Supervision Approach

A very popular and successful approach for information extraction is to use structured information resources as sources of supervision. This approach is commonly known as *weak supervision* or *distant supervision* [21]. Craven and Kumlien [6] and Morgan et al. [22] made the observation that in many fields, such as Biology, there exist rich information sources, such as lexicons, ontologies, knowledge bases and databases, that can be heuristically coupled with text documents to automatically create annotated corpora. Examples of information sources are Freebase[4], DBpedia[5], Yeast Protein Database[6], FlyBase[7], among others.

Craven and Kumlien [6] developed a method that maps the Yeast Protein Database to MEDLINE abstracts, in order to extract training examples about proteins and the subcellular locations in which they are found. Then, a Naïve Bayes model and a relational learning model were trained using the generated dataset. Both methods showed that using weakly-labeled examples resulted in similar or better results when compared to using hand-labeled examples, and generating the weakly-labeled examples was less expensive than obtaining hand-labeled examples.

Mintz et al. [21] used Freebase to obtain relations and entity pairs, and then matched these facts with Wikipedia text articles. Then, they trained a multiclass logistic regression classifier based on lexical and syntactic features, using the weakly-supervised examples (which are potentially noisy) as training set.

One of the key issues with the above methods is the inherent limitation of distant supervision. The *distant supervision assumption* [33] states the following:

> *"If two entities participate in a relation,* ***all sentences*** *that mention these two entities express that relation".*

Two entities may appear in the same sentence, but that does not mean that the sentence is expressing the relation between the entities. This may result in noisy training examples, which affect the performance of the information extraction systems.

Several methods have been proposed to overcome the previous issue. One of them is to cast the problem as multi-instance learning, where the distant supervision assumption is relaxed. Multi-instance learning is based on the *expressed-at-least-once assumption* [33]:

> *"If two entities participate in a relation, at least one sentence that mentions these two entities might express that relation".*

[4] http://www.freebase.com/

[5] http://dbpedia.org/

[6] http://www.proteome.com/YPDhome.html

[7] http://flybase.org/

Bunescu and Mooney [29] combined weak supervision and multi-instance learning to develop a system for relation extraction. By just having a small set of pairs of entities that are known to belong or not belong to a given relationship, they extracted sentences that contain both entities. Then, they used a Support Vector Machine model with a subsequence kernel to perform relation extraction.

Riedel et al. [33] combined weak supervision and multi-instance learning by training a graphical model, which was trained by posing distant supervision as a constraint-driven semi-supervision problem. Semi-supervised learning consists in augmenting labeled data with a large amount of unlabeled data to build better classifiers. In this case, training instances are divided into bags, and some bags are labeled to indicate that they contain at least one positive example, while the remaining ones contain only negative examples. They used sentence-level binary hidden variables to denote whether a relation between a pair of entities appears in a sentence, and an aggregate-level hidden variable to indicate the relation that is expressed in the corpus. Results showed that the model with the expressed-at-least-once assumption outperformed the distant supervision baseline.

Riedel's approach [33] was extended by Hoffmann et al. [13] to allow overlapping relations between the same pair of entities (e.g., founded(Jobs, Apple) and CEO-of(Jobs, Apple)). This was achieved by allowing the sentence-level variables to take the value of any relation (instead of being binary), and employing multiple binary aggregate-level variables. Weak supervision was modeled by treating sentence-level variables as latent, and using facts from a database as supervision for the aggregate-level variables. Allowing overlapping relations significantly improved performance while reducing running time.

Multi-instance learning approaches can fail when a labeled entity pair is mentioned only once in a corpus, and the assigned label is incorrect [40]. If there is only one mention of the entity pair in the corpus, because of the expressed-at-least-once assumption, multi-instance learning assumes that the sentence that contains the mention is a positive example, which is incorrect. In other words, multi-instance learning is reduced to distant supervision, which results in a noisy example. To overcome this issue, Takamatsu et al. [40] proposed a novel approach of reducing the number of wrong labels by using patterns. These patterns consist of entity types of an entity pair along with the sequence of words between the two entities, and were predicted using a generative model. They used a multi-class logistic classifier to perform relation extraction, and cleaned the labeled data with the predicted patterns. Their approach obtained comparable or higher precision at most recall levels compared to distant supervision [21] and multi-instance learning [13] approaches.

### 2.1.4 Biomedical Information Extraction

Several approaches have been proposed for knowledge extraction from biomedical literature. On the task of sentence classification, Gurulingappa et al. [11] developed

a system for the automatic identification of adverse drug event assertive sentences. They used several supervised learning methods, such as Naïve Bayes, Decision Trees, Maximum Entropy and Support Vector Machines, and exploited lexical, syntactic and contextual features divided in feature sets. Results showed that almost all the feature sets resulted in varying improvements of the classifier performances, and that the Maximum Entropy classifier got the best performance. They performed error analysis to find out sources of errors. False positives occurred because of several reasons, such as lack of explicit mentioning of the observed medical problem and sentences containing adverse effects associated with different forms of medical treatments. False negatives occurred because of sentences containing adverse effects not contained in the lexicon, long sentences, and sparsely defined association between a drug and a medical problem.

Riedel et al. [30] and Poon and Vanderwende [28] proposed approaches based on Markov Logic Networks (MLNs) [8] to perform biomedical event extraction, getting competitive results in the BioNLP'09 Shared Task [16]. They employed MLNs to capture the models for the extraction of nested-bio-molecular events from research abstracts, and then performed joint inference using these models. MLNs have become popular in biomedical extraction tasks, as has been demonstrated in the BioNLP'11 Shared Task, where the top systems [31, 32] employed approaches based on MLNs.

## 2.2 Statistical Relational Learning

### 2.2.1 Introduction

Algorithms that operate on real world problems must deal with uncertainty and complex relational structures. Several statistical learning frameworks, such as Bayesian and Markov networks, have been proposed to handle uncertainty by representing and reasoning with probabilistic models. On the other hand, relational learning frameworks, such as first-order logic, focus on exploiting the structure and relations in the data. Statistical Relational Learning (SRL) aims to combine both statistical learning and relational learning into one unified approach, where the expressiveness of first-order logic is combined with the ability of probability theory to model uncertainty [23].

Many tasks have been addressed by SRL because of the need of reasoning under uncertainty and modeling complex relational structures. Examples include link prediction, collective classification, social network modeling and analysis, link-based clustering, etc. [8, 23]. A task that is particularly suited for SRL methods is Natural Language Processing (NLP) [3, 27]. Till recently, most NLP approaches employed propositional methods, where they defined a set of features relevant to the task and used methods such as logistic regression. To obtain these features, they used structured output such as parse trees, dependency graphs, etc. obtained from an NLP toolkit.

SRL is naturally applicable NLP for several reasons. On one hand, NLP methods must handle uncertainty because of syntatic and semantic ambiguities, as well as contradictions and errors found in natural language text [27]. NLP is comprised of several subtasks, such as part-of-speech tagging, phrase chunking, syntactic parsing, entity recognition, word-sense disambiguation, etc., all of which require handling uncertainty [3]. On the other hand, NLP methods must deal with an arbitrary number of entities (e.g., people, places, organizations), which may contain an arbitrary set of complex relationhips between them [3]. Entity mentions in text are composed of word or phrases, and each of them have several features, such as part-of-speech tags, phrase types, and word lemmas. Complex relationships exist between words and phrases, represented by parse trees and dependency paths, as well as between their features. SRL methods are capable of handling these complex relationships, as well as the uncertainty mentioned above.

Many tasks such as BioNLP [16] and TempEval [42] involve multiple relations that need to be extracted jointly. Moreover, there are constraints on these relations, which are either defined by the task or by the user. To address these issues, Chambers and Jurafsky [4] defined the constraints using Integer Linear Programming to jointly extract a consistent set of temporal relations. SRL models, on the other hand, can define the constraints much easily using first-order logic and can learn the model based on these constraints. As a result, SRL models (e.g., MLNs) have been used extensively for these tasks [30, 28, 44].

### 2.2.2 Relational Functional Gradient Boosting

In our first contribution, we use the boosted Relational Dependency Network (RDN) learner by Natarajan et al. [23] to learn a joint model for several target relations. This method is the state-of-the-art learning algorithm for learning SRL models from data. This algorithm is based on Functional Gradient Boosting [23, 15, 14, 7, 10], and in this case learns RDNs. RDNs are SRL models that approximate a joint distribution as a product of conditional distributions over ground atoms. These conditional distributions are learned independently, so RDNs are significantly easier to learn compared to relational versions of Bayesian networks. Another advantage is that the models are allowed to be cyclic, hence they are capable of capturing cyclic dependencies that might exist in the data [23]. In this section we briefly describe the boosted RDN learner algorithm.

Assume that the training examples are of the form $(\mathbf{x_i}, y_i)$ for $i = 1, ..., N$ and $y_i \in \{1, ..., K\}$. We use $\mathbf{x}$ to denote the vector of features and $y$ corresponds to target relations. Relational models consider training instances as "mega examples" where each example represents all instances of a particular group (e.g., a university, a research group, etc.). Given the above definitions, the goal is to fit a model $P(y|\mathbf{x}) \propto e^{\psi(y,\mathbf{x})}$ for every target relation $y$. To this effect, we use Functional Gradient Boosting to calculate the potential of the mentioned model.

Functional Gradient Boosting is based on functional gradient ascent. Functional gradient ascent starts with an initial model $\psi_0$, which returns a constant regression value for all examples, and iteratively adds gradients $\Delta_i$. The gradient at episode $m$ is given by

$$\Delta_m = \eta_m \times E_{x,y}[\partial/\partial\psi_{m-1} log\, P(y|x;\psi_{m-1})] \tag{2.1}$$

where $\eta_m$ is the learning rate. After $m$ iterations, the potential is given by

$$\psi_m = \psi_0 + \Delta_1 + ... + \Delta_m. \tag{2.2}$$

Relational Functional Gradient Boosting (RFGB) extends Functional Gradient Boosting to relational domains where the goal is to learn models in the presence of rich relational structures. In RFGB, the potential functions are represented by sums of relational regression functions (typically relational regression trees [2]) that are grown stage-wise.

We start with an initial model, which can be provided by a domain expert or can be set to capture the uniform distribution of the experiment. At each iteration, we make predictions using the current model, which consists in assigning probabilities to each training example. Then, we compute the functional gradient for each example. It was shown by Natarajan et al. [23] that the functional gradient of the likelihood for each example $(\mathbf{x_i}, y_i)$, with respect to $\psi(y_i = 1; \mathbf{x_i})$, is given by:

$$\frac{\partial log P(y_i; \mathbf{x_i})}{\partial \psi(y_i = 1; \mathbf{x_i})} = I(y_i = 1; \mathbf{x_i}) - P(y_i = 1; \mathbf{x}_i) \tag{2.3}$$

where $I$ is the indicator function that is 1 if $y_i = 1$ and 0 otherwise. Thus, the gradient is the difference between the true label and current predicted probability, i.e., the adjustment of the predicted probability to match the observed value $(y_i)$ for each example. We use these gradients as weights to learn a relational regression tree [2] that fits the regression examples and add it to the current model. We now compute the gradients based on the updated model and repeat the process. Therefore, in every subsequent iteration, we *fix* the errors made by the model.

Hence, the key idea is to consider the conditional probability distribution of the target predicates as a set of relational regression trees [2]. By computing the gradients at each iteration for each training example and adding them to the current model, we are maximizing the likelihood of the distributions with respect to the potential functions. For further details about RFGB, we refer the readers to Natarajan et al. [23].

### 2.2.3 Markov Logic Networks

We also employ the formalism of Markov Logic Networks (MLNs) [8] to capture useful knowledge for information extraction tasks. This knowledge includes world knowledge

to generate weakly supervised examples, and knowledge about the relationships described in textual data using text patterns and string similarities. We employ MLNs because they provide an easy way to capture knowledge that can be supplied by a domain expert.

MLN [8] is a statistical relational framework that combines Markov networks and first-order logic to handle uncertainty and complex relational structures. An MLN consists of a set of pairs $(F_i, w_i)$, where $F_i$ is formula or *clause*, similar to first-order logic, and $w_i$ is the weight of the formula. Clauses express the structural relationships between the random variables. Weights, which are real numbers, express the strength of the relationships.

Intuitively, the weight of a formula $F_i$ is simply the log-odds between a world where $F_i$ is true and a world where $F_i$ is false, other things being equal. Because weights are log-odds, they can take values ranging from $-\infty$ to $+\infty$. A weight of $\infty$ means that the clause is a "hard" constraint on the set of possible worlds, i.e., if a world violates the clause, it has zero probability. A weight different from $\infty$ means that the clause is a "soft" contraint, i.e., if a world violates the clause, it is less probable but not impossible.

For example, consider the following clauses:

$$\begin{array}{ll} 0.5 & smokes(X) \rightarrow cancer(X) \\ 1.0 & friends(X,Y) \wedge smokes(X) \rightarrow smokes(Y) \end{array}$$

As can be seen, clauses are written as first-order formulas. The first clause expresses the knowledge that if a person (denoted by X) smokes, then he/she is likely to have cancer. The second clause expresses the knowledge that if a person (denoted by Y) is friends with a person (denoted by X) who smokes, then he/she is likely to smoke as well. The weights on the left of the formulas represent the *likelihood* of each clause. Because the numbers are log-odds, the probability of friends having similar smoking habits is $log(1.0/0.5)$ times more likely in the world compared to smoking causing cancer. Note that X and Y are variables that can be instantiated with values such as *Adam, Bob, Cathy, David,* etc.

MLNs can be seen as a template to construct Markov networks [26]. Given a finite set of constants, each ground predicate becomes a node in the network and each formula becomes a clique in the network. Each node in the Markov network is binary and takes the value of 1 if the ground predicate is true, and 0 otherwise. All groundings of the same formula are assigned the same weight, leading to the following joint probability distribution over all atoms:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right) \tag{2.4}$$

where $n_i(x)$ is the number of times the $i$th formula is satisfied by possible world $x$ and $Z$ is a normalization constant (as in Markov networks). Intuitively, a possible world

where formula $F_i$ is true one more time than a different possible world is $e^{w_i}$ times as probable, all other things being equal. Several weight learning, structure learning and inference algorithms have been proposed for MLNs. We refer to the book by Domingos and Lowd [8] for more details.

The weights of the clauses can be assigned manually by a domain expert, or they can be learned. Effective algorithms exist for learning the weights of these clauses given data [19, 37], as well as performing inference. Inference in MLNs can be performed using standard Markov network inference techniques. In our work, we use the Tuffy system [24] to learn the weights and perform inference on the MLNs. One of the key attractions of the Tuffy system is that it can scale to millions of documents and thus can provide a very efficient tool.

# Chapter 3: Exploiting Commonsense Knowledge for Weak Supervision

In this chapter, we describe our proposed method for using commonsense knowledge to automatically create (noisy) training examples from text. One of the most important challenges facing many natural language processing tasks is the paucity of "gold-standard" examples. For this reason, we take a different approach for creating more examples for the supervised learner based on weak supervision [6]. Our method consists of two distinct phases: *weak supervision phase*, where we create weakly supervised examples based on commonsense knowledge, and *information extraction phase*, where we learn the structure and parameters of the models that perform the relation extraction and text categorization tasks using textual features. In this chapter we describe both phases, as well as the experiments where we empirically validate our proposed approach on two natural language domains.

## 3.1 Generating Examples Using Commonsense Knowledge

Let us consider two sentences from a news article about a National Football League (NFL) game given in Figure 3.1. From these sentences, humans can extract valuable information. First, we can identify named entities: people (Aaron Rodgers, Nick Collins), organizations (Green Bay Packers, Pittsburgh Steelers), dates (Sunday), and numbers (three, 31-25). With some knowledge about NFL, readers would know that Green Bay Packers and Pittsburgh Steelers are NFL teams, and "Super Bowl" is the name of the championship game of the NFL. Furthermore, a NFL fan would probably know that Aaron Rodgers and Nick Collins are NFL players that played together for the Green Bay Packers. Moreover, the score "31-25" gives us a clue that the article is most likely about an American football game instead of other sports, such as soccer, baseball, or basketball. Finally, there are clue words and phrases that let us know that the Packers won the game, such as "turned the Green Bay Packers into Super Bowl champions" and "leading the Packers to a 31-25 victory".

As can be seen, a human can extract lots of information from a small amount of unstructured text. The goal of information extraction is to be able to extract this information, by combining different natural language processing tasks: named entity recognition (identify entities such as "Aaron Rodgers" and "Green Bay Packers"), relation extraction (Packers winning the game against the Steelers), and text categorization (identify the article to be about American football). In this chapter we focus on the tasks of relation extraction and text categorization. We assume that named entity recognition is performed by a standard tool such as Stanford NLP toolkit [9, 17].

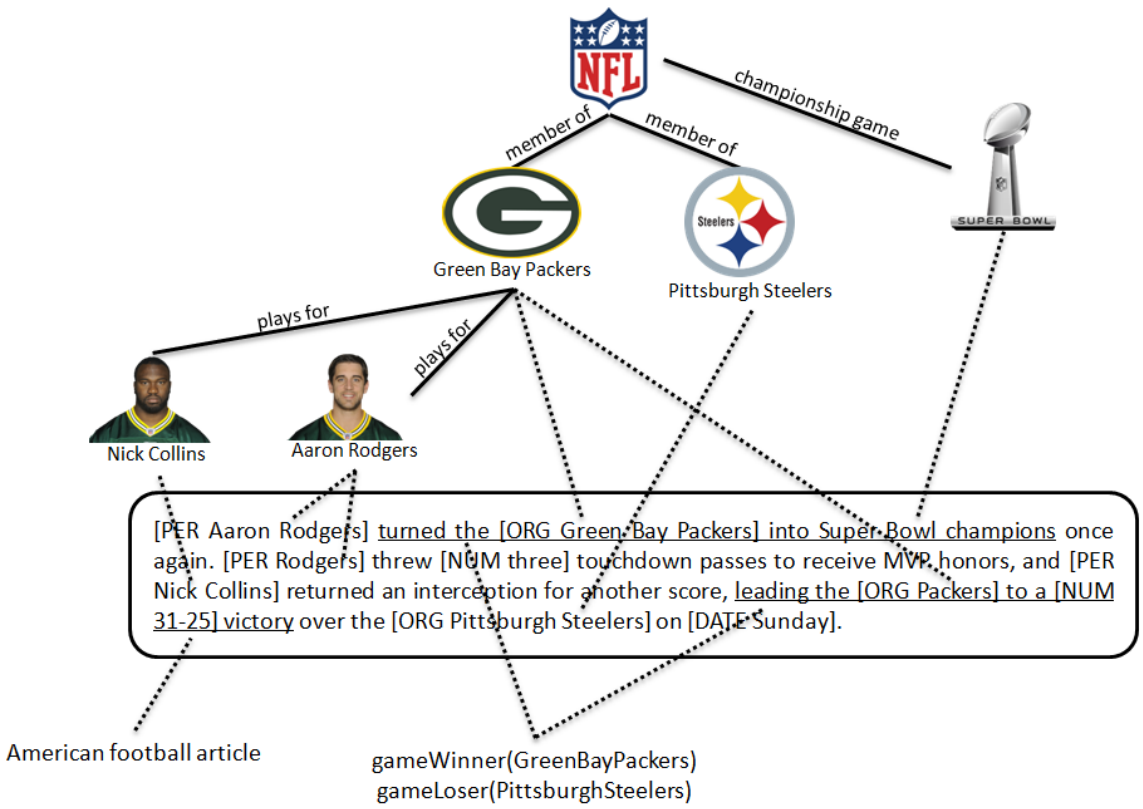


Figure 3.1: Sample text annotated with entities and relations.

### 3.1.1 Knowledge in Information Extraction

To address the problem of information extraction, we employ a method that is commonly taken by humans. When reading a text document, humans can understand and infer facts from the text, even if they are not stated explicitly in the text. This is possible by having some previous knowledge about the topic. In information extraction, previous knowledge can be represented by semantic information. Furthermore, we can infer facts from the text by using commonsense knowledge. For instance, consider reading a newspaper sports section about a particular sport (say NFL). Before we even read the article, we have an inherent *inductive bias* – we expect a high ranked team (particularly if it plays at home) to win. In other words, we rarely expect "upsets".

Thus, our approach consists in using semantic information and commonsense knowledge (which we call world knowledge) to extract facts from text documents. We aim to formalize the notion of world knowledge by employing a model that captures the inductive bias, which is used to label examples for the supervised learner.

### 3.1.2 Markov Logic for Knowledge Representation

We employ the formalism of Markov Logic Networks (MLNs) to capture some commonsense knowledge about the domain of interest. We call this as *WMLN* (world MLN). For instance, consider the NFL domain. We can use commonsense knowledge such as "If a team is the winner of the game, there must exist a team different from the winner team that is the loser of the game" (and vice versa), which can be captured by the WMLN, as can be seen in the first two rules in Table 3.1. Furthermore, we can have knowledge such as "Home team is more likely to win the game" and "High ranked team is more likely to win the game", as can be seen in the third to sixth rules in Table 3.1. Another clause that is particularly useful is to make the conjunction of the heads of the previous two rules, that is to say that "A team that is higher ranked and is the home team is more likely to win the game", as can be seen in the last two rules in Table 3.1. In these rules, $g$ is a game, $t, t1, t2, etc.$ are teams, and $y$ is the year.

Note that these rules are written without having the news articles or natural language semantics in mind. They are simply written by the domain expert. Some rules use predicates that may not be available as evidence, but can be inferred from evidence. For instance, one could simply define a higher ranking using the following MLN clause, where $t$ denotes a team, $r$ its rank, $y$ the year of the ranking and $hR$ the higher rank:

$$\infty \quad rank(t1, r1, y), rank(t2, r2, y), t1! = t2, r1 < r2 \rightarrow hR(t1, t2, y)$$

Another important aspect is that this commonsense knowledge, unlike text patterns or other type of knowledge, is completely independent of the natural language

| |
|---|
| exist t2 winner(g, t1), t1 != t2 → loser(g, t2) |
| exist t2 loser(g, t1), t1 != t2 → winner(g, t2) |
| home(g, t) → winner(g, t) |
| away(g, t) → loser(g, t) |
| tInG(g, t1), tInG(g, t2), hR(t1, t2, y) → winner(g, t1) |
| tInG(g, t1), tInG(g, t2), hR(t1, t2, y) → loser(g, t2) |
| home(g, t1), tInG(g, t1), tInG(g, t2), hR(t1, t2, y) → winner(g, t1) |
| away(g, t2), tInG(g, t1), tInG(g, t2), hR(t1, t2, y) → loser(g, t2) |

Table 3.1: A sample of WMLN clauses used for NFL task. $t$ denotes a team, $g$ denotes a game, $y$ denotes the year, $tInG$ denotes that the team $t$ plays in game $g$, $hR(t1, t2, y)$ denotes that $t1$ is ranked higher than $t2$ in year $y$.

text. This is the key innovation of our approach, as the knowledge to support the creation of weakly supervised examples won't be biased by writing styles and other challenges in natural language processing that were mentioned before.

This knowledge captures an inductive bias, which means that we expect the rules to hold, but it will not always be the case (e.g., an away team winning an NFL game). For this reason, rules must be softened. This can be done in two ways. The first is to have a domain expert assign weights to the clauses manually. This method can be employed when the knowledge expressed by the clauses in the MLN can be quantified reasonably. Note that in many cases, the absolute weights may not matter as much as the relative weights between the rules (recall the log-odds explanation pointed earlier). If this is not the case, weights can be learned using a knowledge base. There exist several knowledge bases that provide structured data, such as Wikipedia Infoboxes[1], DBpedia[2], Freebase[3], YAGO[4], or more domain specific knowledge bases such as Pro-Football-Reference[5].

### 3.1.3 Weak Supervision Phase

In the weak supervision phase, we create weakly supervised examples based on commonsense knowledge. Our proposed approach for weak supervision is presented in Figure 3.2.

The first step is to employ an MLN that captures the commonsense knowledge (WMLN), as has been described in the previous section. This step involves writing

[1]http://www.wikipedia.org/

[2]http://www.dbpedia.org/

[3]http://www.freebase.com/

[4]http://www.mpi-inf.mpg.de/yago-naga/yago/

[5]http://www.pro-football-reference.com/

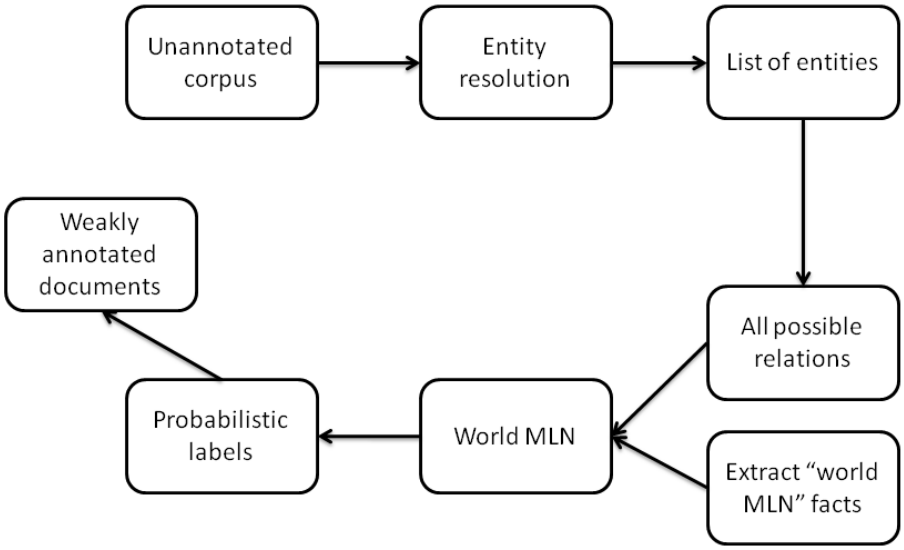


Figure 3.2: Steps involved in creation of weakly supervised examples.

the rules that capture this knowledge and assigning or learning weights for these rules.

Once the WMLN is written and the weights are learned, we proceed to create weakly supervised learning examples. To this effect, we identify interesting (unannotated) documents – for example, sport articles from different news websites. We use a standard NLP tool such as the Stanford NLP toolkit [9, 17] to perform entity resolution to identify the potential entities of interest. Once entities of interest are identified, we query the WMLN for obtaining the posterior probability on the relations between these entities. For instance, if we are interested in game winner and loser relations from NFL articles, we extract potential teams, games and the year of the document. Then we query the WMLN that contains knowledge about NFL to obtain the posterior probabilities on the game winner and loser relations.

Recall that to perform inference, evidence is required. For instance, we might need to know which team was playing home and which team was playing away in an NFL game to predict who won the game. Hence, we use the games that have been potentially played between the two teams (from previously played games in the year of the article) to identify evidence of interest, such as the home and away team. The result of the inference process is the posterior probabilities of the relations between the entities extracted in the documents. The resulting relations are then used as annotations to create weakly supervised learning examples.

There exist two possible annotations schemes. One simple scheme is using the maximum a posteriori probability (MAP) estimate to find out the most likely possible world (i.e., if the probability of a team being a winner is greater than the probability of the team being a loser, the relation becomes a positive examples for winner and a negative example for loser). The second annotation scheme is marginal inference,

which consists in directly using the probabilistic labels and learning a model to predict these probabilities. Choosing the MAP scheme would make a strong commitment about several examples on the borderline. Note that since our world knowledge is independent of the text, it may be the case that in some examples perfect labeling is not possible. In such cases, using a softer labeling method would be more beneficial. Hence, we use the marginal inference annotation approach.

As it is necessary to learn from noisy labels, the learning algorithm is adapted to learn from probabilistic examples, as we present in the next section. Now the examples are ready for out next step – learning the model for *information extraction*.

## 3.2 Learning for Information Extraction Using Weakly Supervised Examples

### 3.2.1 Information Extraction Phase

Once the weakly supervised examples are created, the next step is inducing the relations. In order to do so, we employ the procedure presented in Figure 3.3.

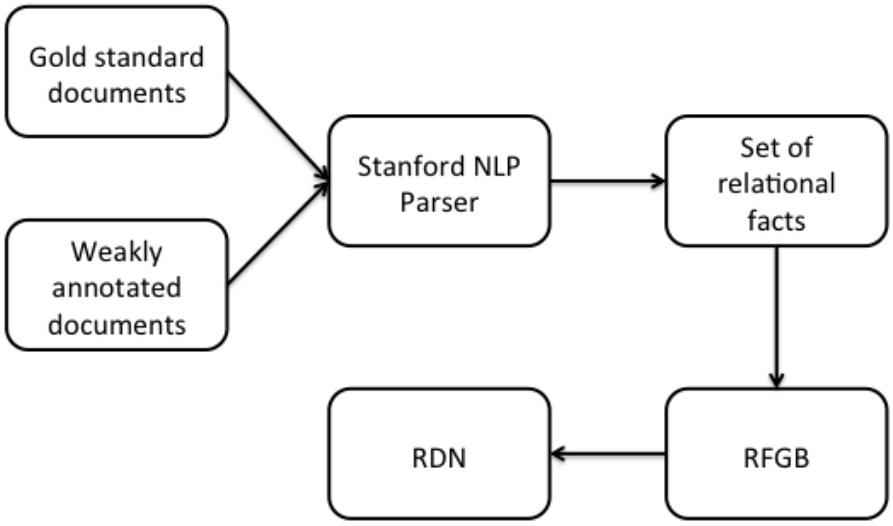


Figure 3.3: Steps involved in learning using probabilistic examples.

We run both the gold standard and weakly supervised annotated documents through the Stanford NLP toolkit [9, 17] to create relational linguistic features. The created relational features are lexical, syntactic and semantic features, such as part-of-speech tags, phrase types, word lemmas, parse trees and dependency paths, which provide a representation of grammatical relations between words in a sentence. Once these features are created, we run the boosted Relational Dependency Network (RDN) learner by Natarajan et al. [23]. This allows us to create a joint model between the

target relations (e.g., the relations *gameWinner* and *gameLoser*). We now describe the adaption of the boosted RDN to this task.

Training examples are of the form $(\mathbf{x}_i, y_i)$ for $i = 1, ..., N$ and $y_i \in \{1, ..., K\}$, where $\mathbf{x}$ denotes the vector of features and $y$s corresponds to target relations. In our work, we consider each document to be a mega example and we do not learn across mega examples, i.e., we do not consider cross document learning. In our formalism, $y$ corresponds to the target relations, for example *gameWinner* and *gameLoser* relations between a team and game mentioned in a sentence. $\mathbf{x}$ corresponds to all the relational facts associated with these mentions.

Given the above defitions, we learn a model for every target relation using the boosted RDN learner by Natarajan et al. [23]. Since we use a probabilistic model to generate the weakly supervised examples, our training input examples will have probabilities associated with them. The relational functional gradient boosting approach was extended to handle probabilistic examples by defining the loss function as the KL-divergence between the observed probabilities (shown using $P_D$) and predicted probabilities (shown using $P$). The functional gradients for the KL-divergence loss function can be shown to be the difference between the observed and predicted probabilities.

$$\begin{aligned} \Delta_m(x) &= \frac{\partial}{\partial \psi_{m-1}} \sum_{\hat{y}} P_D(y = \hat{y}) \log \left( \frac{P_D(y = \hat{y})}{P(y = \hat{y} | \psi_{m-1})} \right) \\ &= P_D(y = 1) - P(y = 1 | \psi_{m-1}) \end{aligned}$$

Hence the key idea in our work is to use probabilistic examples that we obtain from the weakly supervised phase as input to our structure learning phase along with gold standard examples (with $p = 1$ for positive examples and $p = 0$ for negative examples). Then a RDN is induced by learning to predict the different target relations jointly. Since we are learning a RDN, we do not have to explicitly check for acyclicity. We chose to employ RDNs as they have been demonstrated to have the state-of-the-art performance in many tasks [23].

To summarize, we use the weakly supervised examples generated from the previous step and their documents along with the gold standard labels and their associated documents. The entire set of documents is taken as input to the boosted RDN learner with the gold standard positive examples having a probability of 1 (and negative examples having a probability of 0), while the weakly supervised examples' probability is determined by the inference process in WMLN. Given these examples, an RDN is learned using linguistic features created by the Stanford NLP toolkit.

## 3.3 Experimental Results

In this section, we present the results of empirically validating our proposed approach on two natural language domains. We explore two information extraction tasks:

relation extraction and text categorization. The first domain consists in predicting NFL games winners and losers relations by reading NFL news articles. The second domain consists in classifying documents either as being American football or soccer articles. In both domains, we compared the use of augmenting with weakly supervised examples against simply using gold standard examples.

Besides parameter estimation, our approach also performs structure learning. This means that we are able to learn a model without knowing any prior structure. As this is one of our contributions, we do not compare to other weak supervision methods directly, but instead point out the state-of-the-art results in the problem.

### 3.3.1 Relation Extraction

#### Task Description

The goal of this task is to read NFL news articles and identify concepts such as *winner* and *loser* in the text, i.e., perform relation extraction. As an example, consider the text "Packers defeated Cowboys 28-14 in Saturday's Superbowl game". The task is to identify *Green Bay Packers* and *Dallas Cowboys* as winner and loser respectively.

#### Dataset

To evaluate our method, we used the National Football League (NFL) dataset from LDC corpora[6]. This dataset consists of articles about NFL games over the past two decades. Some of these articles are annotated with the target concepts. As we used these articles and annotations as gold standard examples, we considered only articles that have annotations of positive examples. We found 66 annotations of the relations, of which 16 were used as test set and the rest as training set.

In addition to the gold standard examples, we used articles from the official NFL website[7] for weak supervision. We downloaded 150 news articles about NFL games, and extracted 400 weakly supervised examples. These are used to evaluate the impact of the weakly supervised examples using our approach.

#### World Knowledge

The world MLN (WMLN) is key in our approach, as it is used as the source of weak supervision. The rules were written by a domain expert and they were softened using a knowledge base. In this experiment, we used the Pro-Football-Reference[8] knowledge base, which is a collection of football data that has been collected for several years.

[6]http://www.ldc.upenn.edu

[7]http://www.nfl.com

[8]http://www.pro-football-reference.com/

The rules and the resulting weights are presented in Table 3.2. We used the games played in the last 20 years to compute these weights.

| | |
|---|---|
| $\infty$ | winner(g, t) $\rightarrow$ tInG(g, t) |
| $\infty$ | winner(g, t) $\rightarrow$ !loser(g, t) |
| $\infty$ | exist t2 winner(g, t1), t1 != t2 $\rightarrow$ loser(g, t2) |
| $\infty$ | exist t2 loser(g, t1), t1 != t2 $\rightarrow$ winner(g, t2) |
| 0.33 | home(g, t) $\rightarrow$ winner(g, t) |
| 0.33 | away(g, t) $\rightarrow$ loser(g, t) |
| 0.27 | tInG(g, t1), tInG(g, t2), hR(t1, t2, y) $\rightarrow$ winner(g, t1) |
| 0.27 | tInG(g, t1), tInG(g, t2), hR(t1, t2, y) $\rightarrow$ loser(g, t2) |

Table 3.2: All WMLN clauses used for NFL task with their corresponding weights. $t$ denotes a team, $g$ denotes a game, $y$ denotes the year, $tInG$ denotes that the team $t$ plays in game $g$, $hR(t1, t2, y)$ denotes that $t1$ is ranked higher than $t2$ in year $y$.

As has been said, to perform inference, evidence is required. We used the games that have been potentially played between two teams, based on the date of the article being read. This way, we could identify relevant evidence such as home and away teams. As ranking is also used in the WMLN, we used the rankings at the start of the year of the game as a pseudo reflection of the relative ranking between the teams.

### Setup

We evaluated the impact of the weakly supervised examples by comparing two different settings. In the first setting, we used no weakly supervised examples, and simply varied the number of gold standard examples, going from 10 up to 50 gold standard examples. In another setting, we used the same gold standard examples as the previous setting, and added all weakly supervised examples. We kept the number of weakly supervised examples constant. The results were averaged over 5 runs of random selection of gold standard examples.

To evaluate how much impact weakly supervised examples have, we performed experiments where we varied the number of weakly supervised examples and kept the number of gold standard examples constant. We did this for three different settings: 10, 30 and 50 gold standard examples. Weakly supervised examples were varied from 100 to 400. The results were averaged over 5 runs of random selection of weakly supervised examples.

### Results

We measured the area under the curves (AUC) for both Receiver Operating Characteristic (ROC) and Precision-Recall (PR) values. Simply measuring the accuracy on

the test set will not suffice in most structured problems, since predicting a majority class can lead in high performance. Hence we present AUC.

The results of the first experiment are presented in Figure 3.4, where the performance measure is presented by varying the number of gold standard examples. As can be seen, in both metrics, the weakly supervised examples improve upon the usage of gold standard examples. The use of weakly supervised examples allows a jump start, a steeper learning curve and, in the case of PR, a better convergence.

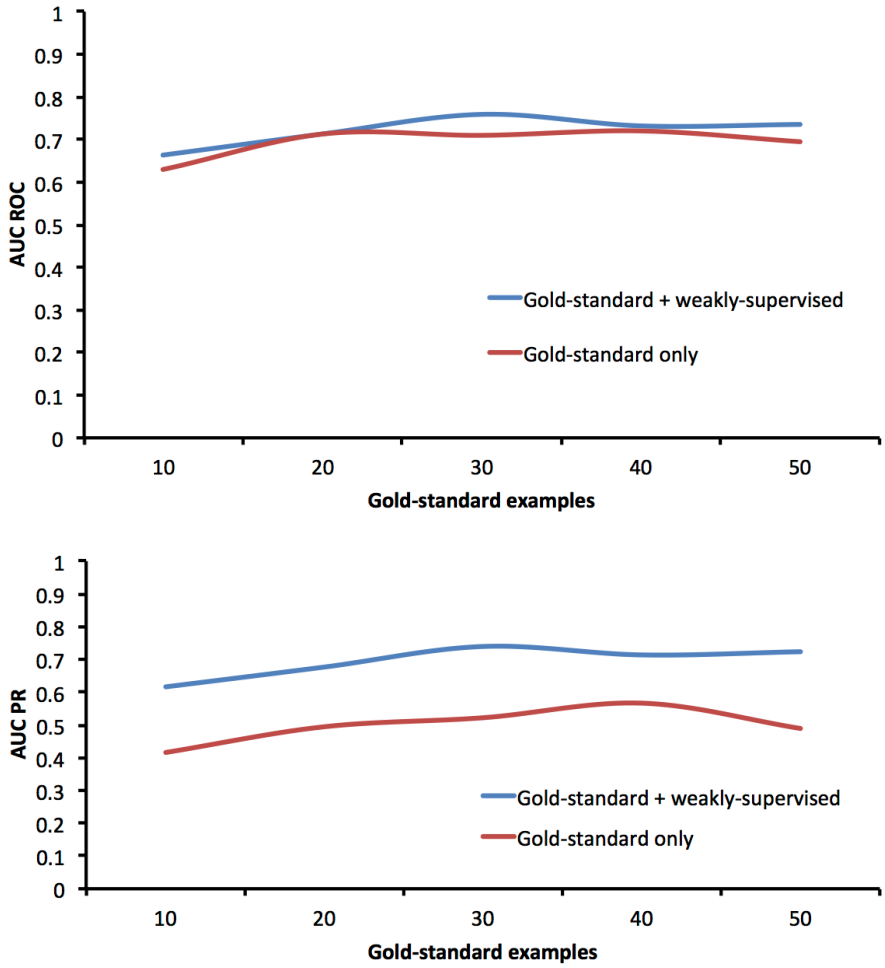


Figure 3.4: Results of predicting *winners* and *losers* in NFL domain, varying gold standard examples and keeping weakly supervised examples constant. **(Top)** AUC ROC results. **(Bottom)** AUC PR results.

It should be mentioned that while plotting every point, the set of the gold standard examples is kept constant for every run and the only difference is whether there are any weakly supervised examples added. For example, when plotting the results of 10 examples, for every run, the set of gold standard examples is the same. For the blue

curve, we add 400 more weakly supervised examples and this is repeated for 5 runs in which the 10 gold examples are drawn randomly.

We also performed t-tests on all the points of the PR and ROC curves. For the PR curves, the use of weakly supervised learning yields statistically superior performance over the gold standard examples for all the points on the curves (with p-value $< 0.05$). For the ROC curves, significance occurs when using 10, and 30 examples. Since PR curves are more conservative than ROC curves, it is clear that the use of these weakly supervised examples improves the performance of the structure learner significantly.

To understand whether weak supervision clearly helps, we performed another experiment using a baseline where we randomly assigned labels to the 400 examples. When combined with 50 gold standard examples, the performance decreased dramatically with AUC values of 0.58 for both ROC and PR curves which clearly shows that the weakly supervised labels help when learning the structure.

The results of the second experiment, where we evaluated the impact of weakly supervised examples using a constant number of gold standard examples, are presented in Figure 3.5. As can be seen, for the settings of 30 and 50 gold standard examples, the performance increased with the number of weakly supervised examples. However this is not the case for the setting of 10 gold standard examples, where the AUC ROC is decreased. This indicates that it is important to keep a balance between the number of gold standard and weakly supervised examples, as using many weakly supervised examples (which are potentially noisy) together with few gold standard examples may affect the performance.

### 3.3.2 Text Categorization

#### Task Description

In the second domain we perform text categorization, particularly document classification. The goal is to classify documents as either being *football (American)* or *soccer* articles. Hence, the relation in this case is on the article (i.e., *gameType(article, type)*).

#### Dataset

We extracted 30 football articles from the official NFL website[9] and 30 soccer articles from the official English Premier League website[10]. These articles were used as gold standard examples, so we annotated them manually as being football and soccer respectively. In addition, we extracted 45 articles from the same websites for weak supervision. We used only the first paragraph of the articles for learning the models

[9] http://www.nfl.com

[10] http://www.premierleague.com

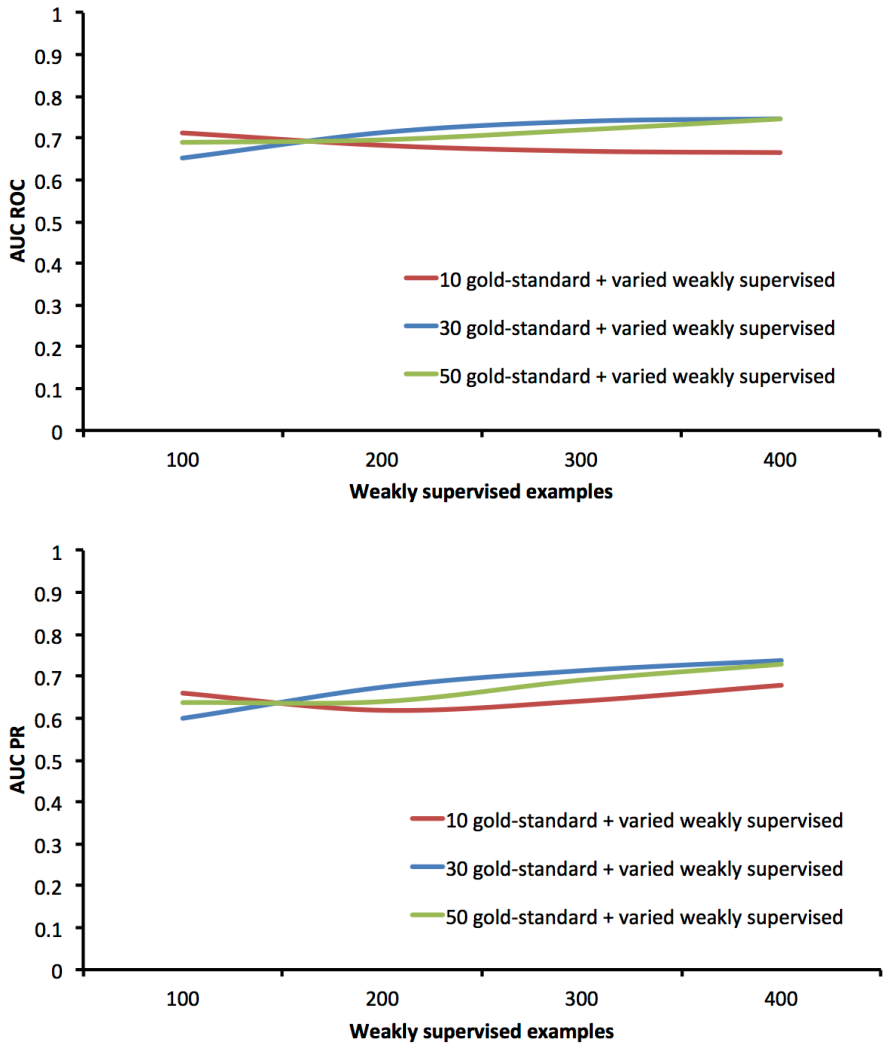


Figure 3.5: Results of predicting *winners* and *losers* in NFL domain, varying weakly supervised examples and keeping gold standard examples constant. **(Top)** AUC ROC results. **(Bottom)** AUC PR results.

since it appeared that enough information is present in the first paragraph for learning a useful model.

### Dataset

In this domain, we used a simpler world knowledge. We used rules such as:

- If the article mentions a NFL league and a NFL team, then it is a football article.
- If the article mentions a NFL team, then it is a football article.
- If the scores of both teams are greater than 10, then it is a football article.
- If the scores of both teams are less than 10, then it is a soccer article.
- If the scores of both teams are 0, then it is a soccer game.

Note that all the rules mentioned above are essentially considered as "soft" rules. The weights of these rules were simply set to 100, 10 and 1 to reflect the log-odds. We could learn these weights as in the previous domain, but the rules in this task are relatively simple, hence we simply set the weight manually.

### Setup

As in the previous domain, we evaluated the impact of the weakly supervised examples. We compared two different settings. In the first setting, we used only gold standard examples. In the second setting, we used the same gold standard examples and added 45 weakly supervised examples. In both settings, we varied the number of gold standard examples, going from 5 to 30. In the second setting we kept the number of weakly supervised examples constant. The results were averaged over 5 runs of random selection of gold standard examples.

### Results

As in the previous domain, we measured the area under the curve for ROC and PR. The results of both measurements are presented in Figure 3.6. The resulting figures how that, as with the earlier case, weak supervision helps in improving the performance of the learning algorithm. We get a jump start and a steeper learning curve in this case as well. Again, the results are statistically significant for small number of gold standard examples.

Both experiments conclusively prove that adding probabilistic examples as weak supervision enables our learning algorithm to improve upon its performance in the presence of small number of gold standard data thus validating the hypothesis that world knowledge helps when manual annotations are expensive.

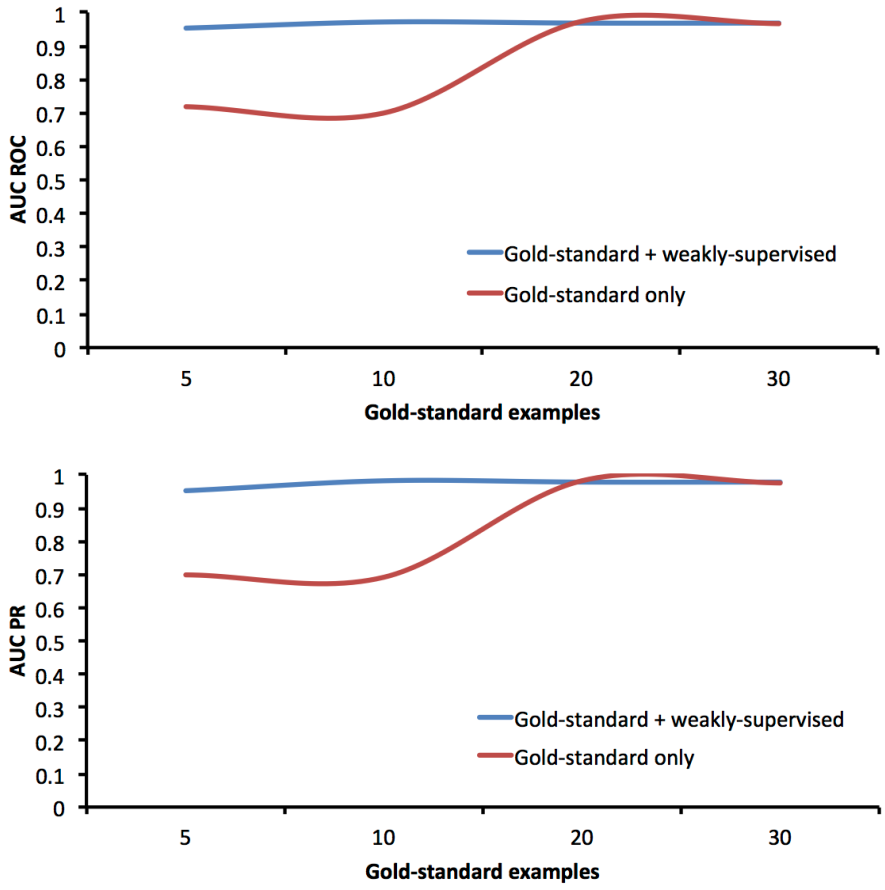


Figure 3.6: Results of document classification. **(Top)** AUC ROC results. **(Bottom)** AUC PR results.

# Chapter 4: Text-Based Evaluation of Adverse Drug Events Discovery

Adverse drug events (ADEs) are a major concern and point of emphasis for the medical professionals, governement and society in general. Several methods have been proposed for ADE discovery from health data (e.g., EHR and claims data), from social network data (e.g., Google and Twitter posts), and from other information sources. In this chapter we propose a NLP-based evaluation methodology for ADE discovery, which applies text and natural language processing algorightms to scientific literature in order to assign scores to proposed ADEs. Our score assigning method consists of two phases: *string simliarity phase*, where we use simple document matching metrics to assign scores, and *semantic relation extracion phase*, where we perform a deeper analysis based on linguistic features. In the following sections we present an introduction to ADE discovery, our proposed approach, and experimental results where we empirically validate our approach.

## 4.1 Adverse Drug Events Discovery

Adverse drug events (ADEs) represent the 4th leading cause of mortality in the US (Institute of Medicine, IOM), and nearly 11% of hospital admissions occurring in older adults are attributable to ADEs [12]. For this reason, there have been several attempts to improve risk evaluation, surveillance, and relative safety associated with drug exposures (IOM). In response to serious gaps in pharmacosurveillance capacity, the Food and Drug Adminisitration (FDA), PhRMA, and the Foundation for the NIH inititated the Observational Medical Outcomes Partnership (OMOP) to evaluate and improve methods for discovering ADEs from observational medical data, such as health insurance claims data, Medicare and Medicaid data, or electronic health record (EHR) data [36]. To facilitate evaluation and comparison of methods and databases, OMOP established: a Common Data Model so that disparate databases could be represented uniformly; definitions for ten ADE-associated "health outcomes of interest" (HOIs); and drug exposure eras for ten widely-used classes of drugs.

The end goal of OMOP's methods development and evaluation is to uncover new (previously unknown and perhaps even unanticipated) ADEs. To evaluate methods it is necessary to use known ADEs as ground truth and determine how well the new methods could have uncovered these ADEs had they been unknown. At its initiation OMOP took a rigorous approach based on available drug label information to associate drug classes with HOI definitions [35]. The result was an "OMOP ground truth" ADE set, where out of 100 possible HOI-drug class pairs, 9 pairs are used as true ADEs, and 44 of the untrue ADEs are used as false ADEs. Our approach presented in the following sections aims to quantitatively evaluate the proposed ADEs,

such as OMOP's ground truth.

## 4.2 Evaluating Adverse Drug Events Using Natural Language Processing

### 4.2.1 Approach Overview

In this section we present our proposed approach for evaluating adverse drug events. The problem can be defined as follows:

**Given:** *A set of <drug(s), condition(s)> tuples.*
**To Do:** *Determine $P(condition(s)|drug(s))$ i.e., output the probability that a given (possibly set of) condition(s) is an adverse event of a (possibly set of) drug(s) by using prior published research as proof for the adverse events.*

In other words, the aim of our work is to determine what the research community knows about the drug-event (DE) pairs. Note that while we refer to the events as drug-event pairs, our work is not restricted to just pairs but can handle complex interactions such as multiple drugs/conditions causing multiple adverse conditions. In this work we restrict ourselves to drug-event pairs (henceforth called as DE) for simpler exposition of the ideas and for comparison to OMOP ground truth.

Our approach is presented in Figure 4.1. Given the previous problem definition, the first step is to obtain previously published literature that provides evidence about the given drug-event pairs, or DE. To this effect, for each of the proposed DEs, we obtain a set of articles by querying PubMed[1] for the DE. We consider only the abstracts of these articles, but our proposed approach can be extended to handle full articles. In particular, we consider the top five articles for each DE. These articles serve as the natural language textual evidence for the pair. They are then used as the input to the next step.

The second step of our approach has two distinct phases: (1) string similarity phase and (2) semantic relation extraction phase. The string similarity phase consists in assigning partial scores to the proposed DE pairs based on simple similarity measures. The semantic relation phase on the other hand assigns scores based on a deeper analysis of the text. Text analysis is done by using text facts, which are extracted using an NLP parser, and the relation extraction knowledge. In the following sections we describe both phases in more detail.

[1] http://www.ncbi.nlm.nih.gov/pubmed

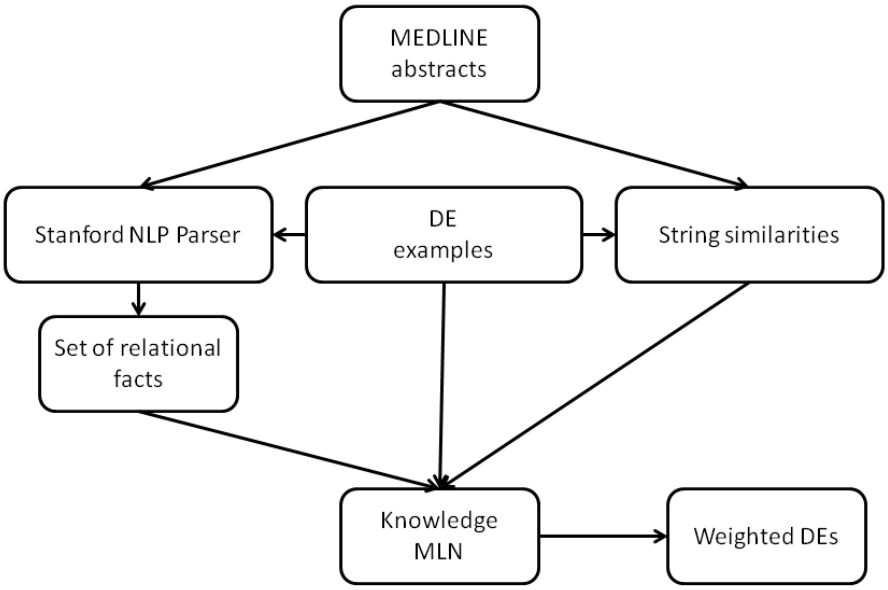


Figure 4.1: Steps involved in the evaluation of adverse drug events (ADEs).

### 4.2.2 String Similarity Phase

The goal of the string similarity phase is to find support of an ADE in scientific literature found on the Web. This is done by obtaining a *syntactic* similarity score between the DE pair and each of the abstracts obtained in the previous step. We use simple document matching metrics such as *cosine similarity*, *Jaccard similarity*, etc. to measure the simliarity between the DE pair and the abstract at hand. Cosine similarity [5] measures the cosine of the angle between two vectors, where the vectors are *frequency of occurrence* vectors of the documents. In our case, vectors store the occurrence of letters. The resulting measures are averaged to obtain a single score for each DE.

Note that the similarity measured in this phase simply searches for mentions and thus does not distinguish whether a given DE is an adverse event or not, or if there is no relationship between the drug and the condition. The resulting string simliarities are used as evidence in the MLN constructed in the next phase.

### 4.2.3 Semantic Relation Extraction Phase

The semantic relation extraction phase on the other hand aims to create features that are used for a deeper analysis of the given text. For each abstract obtained in the first step, we identify the sentences that contain the corresponding DE pair, i.e., sentences that contain both the drug and the condition. Note that we do not distinguish yet whether the DE is an adverse event or not, or simply if the drug and the condition

are related, we just keep the sentences plain text to be used in the next step.

To obtain the relevant features, we run the sentences obtained in the previous step through a standard NLP tool such as the Stanford NLP toolkit [9, 17] to create relational linguistic features. The created relational features are lexical, syntactic and semantic features, such as part-of-speech tags, phrase types, word lemmas, parse trees and dependency paths. In addition to these features, we use an entity recognizer to identify drug and effect mentions. For example consider the following text: *"There is evidence that MI is caused by the intake of Cox2ib"*. This sentence would lead to the features *drug(Cox2ib)* and *effect(MI)*. These features (called as predicates in MLN literature) are then used as ground facts to be combined with the MLN presented next.

As mentioned earlier, we use drugs and effects, and the relationship between them as evidence. Note that after having the information about drugs and effects, we use the predicates $drug$ and $effect$ to define an MLN clause that indicates that the drug $d$ with word $dw$, and effect $e$ with word $ew$, are present in a DE $r$:

$$\begin{aligned} \infty \quad effect(e), effectWord(e, ew), drug(d), drugWord(d, dw), \\ present(r, d), present(e, d) \\ \rightarrow ade(r, d, dw, e, ew) \end{aligned}$$

$dw$ and $ew$ are variables and will be substituted by the corresponding values when performing reasoning. For the example in the above paragraph, $dw$ would correspond to $Cox2ib$ and $ew$ would correspond to $MI$.

Once we have the drugs, effects, DE pairs, string similarities and textual evidence, we employ an MLN that captures the relation extraction knowledge for identifying ADEs using rules about text patterns and string similarities. MLNs have become popular in biomedical extraction tasks, as has been demonstrated in the BioNLP'11 Shared Task, where the top systems [30, 32] employed approaches based on Markov Logic. Some of the rules that capture the text patterns and string similarities that we used are shown in Table 4.1. The first-order rules can be interpreted in English as,

- "If there is a cosine similarity between the DE pair and MEDLINE abstracts, the proposed ADE is true with a weight relative to the cosine similarity" (first rule),
- "If a drug and an effect are present in a proposed ADE and a sentence contains both the drug and effect, the ADE is true" (second rule),
- "If a drug and an effect are present in a proposed ADE, and a sentence contains both the drug and effect, and the sentence contains the pattern 'effect after drug', the ADE is true" (third rule),

- "If a drug and an effect are present in a proposed ADE, and a sentence contains both the drug and effect, and the sentence contains the pattern 'drug-induced effect', the ADE is true" (fourth rule),
- "If a drug and an effect are present in a proposed ADE, and a sentence contains both the drug and effect, and the sentence contains the pattern 'risk of effect' and drug is a participial modifier of the word 'risk', the ADE is true" (last rule).

Note that all rules mentioned above are considered as "soft" rules, and we manually assigned weights to the rules based on the lengths of the dependency paths and the specificity of the rule. Of course, these weights can be learned using data but that is outside the scope of this work. Once the MLN is constructed and weights have been assigned, we query the MLN for the posterior probability on the *adverse* relation, using as evidence the relational linguistic features from the extracted abstracts, as well as drugs, effects, DEs and string similarities.

| *wgt* | cosineSimilarityWeight(r, wgt) → adverse(r) |
|---|---|
| 1 | dpDE(r) → adverse(r) |
| $1.5-3$ | ade(r,d,dw,e,ew), dp(ar,se,ew,dw,dp), contains(dp,"prep_after"), dpl(ar,se,ew,dw,l) → adverse(r, l) |
| 3 | ade(r,d,dw,e,ew), prehw(wo,dw), posthw(wo,postwo), ws(postwo,"induced"), dt(ar,se,ew,wo,amod) → adverse(r) |
| $1.5-3$ | ade(r, d, dw, e, ew), word(wo), ws(wo, "risk"), dp(ar, se, wo, ew, dp1), dp(ar, se, wo, dw, dp2), dpl(ar, se, ew, dw, l), contains(dp1, "prep_of"), contains(dp2, "partmod") → adverse(r, l) |

Table 4.1: A sample of the relation extraction knowledge. *dpDE* denotes that there is a dependency path between the drug and effect in a proposed ADE (they are in the same sentence), *ade* denotes that drug and effect are in a proposed ADE, *dp* denotes the dependency path between two words, *dpl* denotes the length of the dependency path between two words, *prehw* denotes prehyphen word, *posthw* denotes posthyphen word, *ws* denotes word string, *dt* denotes dependency type.

## 4.3 Experimental Results

In this section, we present the results of empirically validating our proposed approach by evaluating the proposed adverse drug events (ADEs). We employed the ADEs dataset provided by the Observational Medical Outcomes Partnership (OMOP). Until now, to our knowledge the OMOP evaluation method is the only quantitative ADE discovery evaluation approach in existence. Therefore we first evaluate our NLP approach on the OMOP ground truth and show that our approach has high but not

perfect agreement with OMOP ground truth. We then look more closely at where our system's results disagree with OMOP's ground truth.

### 4.3.1 Task Description

Adverse drugs events discovery is a relation extraction problem, where the entities are drugs and conditions, and the relations indicate whether a health outcome is a consequence of taking a drug. When evaluating the algorithm, we simply query for the probability that the given health outcome is actually an ADE of the given drug, represented by the *adverse* relation.

### 4.3.2 Dataset

We evaluate our approach on the OMOP ground truth that can be obtained at OMOP's website[2]. OMOP provided true and false ADE pairs, which are composed of widely-used drug classes and health outcomes of interest (HOIs, previously referred to as conditions or effects). Of all the ADE pairs, OMOP classified 9 as positive risks and 44 as negative control ADEs. While plotting the area under the curve of the ROC curve (AUCROC), we used the ground truth values.

### 4.3.3 Setup

For each ADE pair, we extracted 5 MEDLINE abstracts by querying PubMed. From these abstracts, we identified the sentences that contain both the drug class and the HOI, resulting in a total of 104 sentences. We ran these sentences through the Stanford NLP toolkit to create relational linguistic features – lexical, syntactic and semantic features, which were used as evidence. We also stored the drug classes and HOIs, as well as their relationships with ADE pairs, to be used as evidence when querying the MLN.

We compared three different settings. In the first setting, we used the full relation extraction knowledge to evaluate the proposed ADEs. This knowledge consists of text patterns plus the string similarity rules. As this knowledge is encoded in an MLN, we call this setting as MLN. In the second setting, we only used the string similarity rules (denoted as SIM). In the third setting, we used the full extraction knowledge except the string similarities (denoted as MLN - SIM). By comparing these three settings, we were able to study whether there are benefits of using syntactic knowledge through string similarities or deep knowledge about text or both. In other words, we can quantify the benefits of each of the phases: the string similarity phase that merely compares the string similarities, and the relation extraction phase that uses a semantic understanding of the natural language text.

[2]http://omop.fnih.org/sites/default/files/ground%20truth.pdf

### 4.3.4 Results

We used Receiver Operating Characteristic (ROC) to perform performance evaluation. In all settings, we performed five runs, and averaged the area under the ROC curve. We also performed t-tests on all three settings. These results are shown in Table 4.2. For the MLN setting, the average AUCROC was **0.81**. For the SIM setting, the average AUCROC was **0.44**. The ROC curves for the previous settings are presented in Figure 4.2. For the MLN - SIM setting, the average AUCROC was **0.78**. As can be seen, the use of string similarities and semantic understanding (MLN) yields the best performance overall. String similarity analysis helps the semantic understanding by increasing the performance from 0.78 to 0.81, but it is not significant by itself.

| | **AUCROC** | **p-values** | | |
|---|---|---|---|---|
| | | MLN | SIM | MLN - SIM |
| MLN | **0.81** | – | **0.0068** | **0.0125** |
| SIM | 0.44 | 0.0068 | – | 0.0065 |
| MLN - SIM | 0.78 | 0.0125 | 0.0065 | – |

Table 4.2: AUCROC and p-values for the three settings. AUCROC values and p-values were averaged over five runs.

## 4.4 Discussion

The results show that the system performs significantly better than chance, and compares very well with systems designed to extract ADE information from Electronic Health Records (EHRs). The use of text patterns and semantic understanding significantly improves performance compared to only using string similarities.

When considering string similarity only, we observed that several false positive ADEs have high string similarities with literature found on the web. Several of these are even higher than similarities of positive ADEs. Note that the string similarities are simply computing the frequencies that the pair has been mentioned. In some cases, while the number of times the given pair is mentioned could be high, these were essentially negative ADE mentions. The similarity metric ignores phrases such as "negative", "not an effect", "no association", etc. Using text patterns on the other hand, we were able to make a better evaluation of the proposed ADEs since they consider the type of the mention as positive or negative, resulting in a performance improvement.

In Table 4.3, we show some examples of ADE pairs found in the MLN setting in three categories: true positives, i.e., OMOP pairs that we also found to be positive (where the probability of the event being an ADE is high), true negatives, i.e., negative

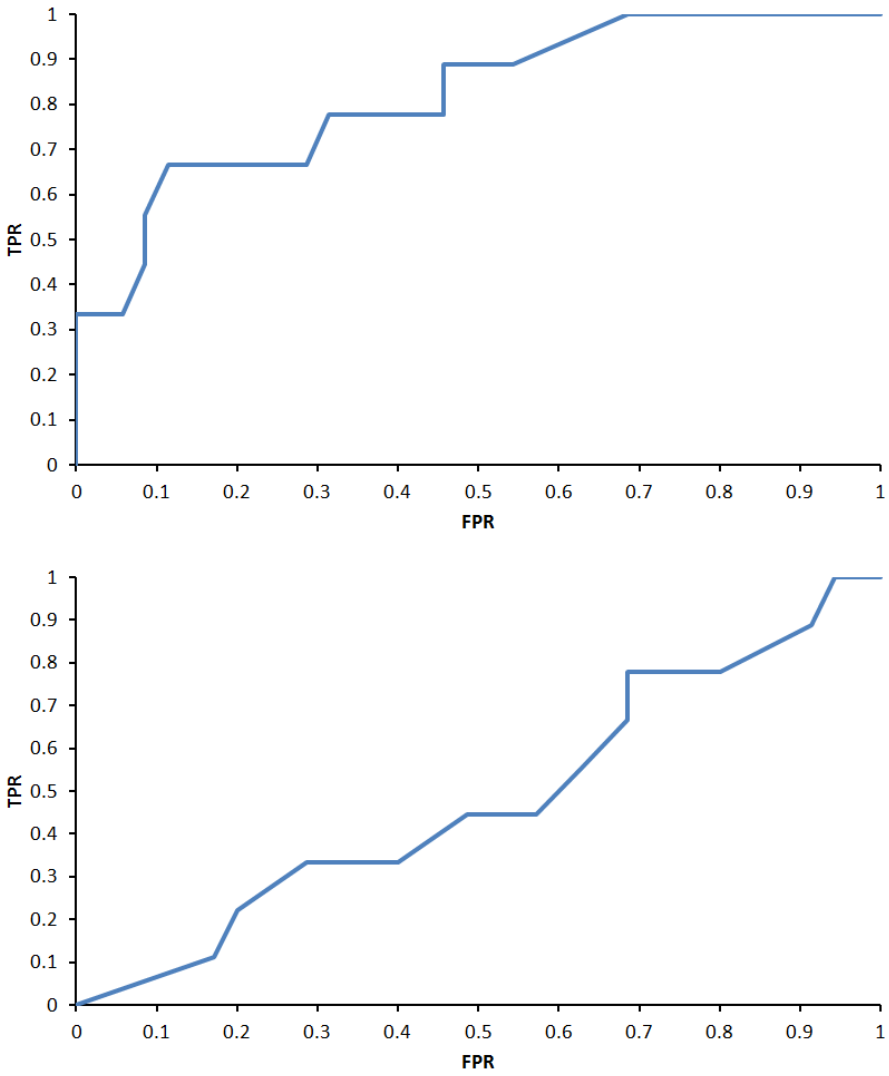


Figure 4.2: Receiver Operating Characteristic (ROC) curve for identification of adverse drug events (ADEs) using OMOP ground truth. **(Top)** Average AUCROC = 0.81 using full relation extraction knowledge (MLN). **(Bottom)** Average AUCROC = 0.44 using only string similarities (SIM).

OMOP pairs that we found to be negative (where the probability of the event being an ADE is low), and false negatives, negative OMOP pairs that we found to be positive (where the probability of the event being an ADE is high).

| **Category** | **Adverse drug event** | **Posterior probability** |
|---|---|---|
| Positive OMOP ADE pairs | ACE Inhibitor - Angioedema | 0.996 |
| | Benzodiazepines - Hip Fracture | 0.994 |
| | Amphotericin B - Acute Renal Failure | 0.934 |
| Negative OMOP ADE pairs | ACE Inhibitor - Aplastic Anemia | 0.506 |
| | Typical Antipsychotic - Upper GI Ulcer | 0.488 |
| | Warfarin - Aplastic Anemia | 0.480 |
| Negative OMOP ADE pairs but positive NLP ADE pairs | Bisphosphonates - Acute Renal Failure | 0.994 |
| | Benzodiazepines - Acute Liver Failure | 0.942 |
| | Warfarin - Acute Renal Failure | 0.936 |

Table 4.3: Examples of ADE pairs.

As expected, some of our results agree with the OMOP ground truth (the top two sets of rows in the table). Note that some of our results have no perfect agreement with OMOP ground truth. This means that some of the negative control ADEs given by OMOP are actually found to be positive by our method. This disagreement reveals some probable directions of investigation for OMOP's ground truth. For instance, consider the ADE "Bisphosphonate causes Acute Renal Failure". This ADE is classified as negative control by OMOP's ground truth. However, it received a high score in our method. When looking closely at the sentences related to this ADE, we found that there is text to support the fact that the ADE is a positive risk, which may contradict OMOP's ground truth. An example sentence that we found from a PubMed article (PMID 11887832) is:

> "Bisphosphonates have several important toxicities: acute renal failure, worsening renal function, reduced bone mineralization, and osteomalacia."

This may happen because of several reasons, such as OMOP's high standard of evidence for ADEs or discoveries occurring after OMOP initiation. Results like this show that our method can be used for ADE evaluation of the ground truth. More importantly, given that the literature is vast, we can find with less human effort ADEs that are already known or have been discovered previously.

A current limitation of our approach is that although it finds evidence for ADEs that were not in the OMOP ground truth (such as a link between bisphosphonates and acute renal failure and a link between antibiotics and increased risk of bleeding with warfarin use) it also falsely interprets some other relationships. For example, it

falsely assigns hip fracture as a warfarin ADE on the basis of sentences such as this one from a PubMed Central article (PMC3195383):

> "There is a need for a national policy for reversing warfarin anticoagulation in patients with hip fractures requiring surgery."

Another error occurs when our approach falsely interprets evidence for a protective effect as evidence for an ADE, interpreting PubMed article with PMID 11826008 as providing evidence that amphotericin B might cause aplastic anemia. Of the ten highest-ranked "false" ADEs by our method from OMOP's ground truth, this is the lowest ranked.

> "We describe a case of primary cutaneous mucormycosis (zygomycosis) in a patient with idiopathic aplastic anemia which responded to surgical debridement and therapy with liposomal amphotericin B."

Other disagreements with OMOP ground truth among the top ten were actual positive evidence for ADEs but with weak evidence in the form of single cases or animal studies.

Our primary goal in this work is to develop a nimble, general tool for evaluating a wide variety of ADE discovery methods that might be based on search engine queries, social network data, or observational medical data such as health insurance claims or electronic health records. It is possible that the best approach will be an ensemble of all of these, and might itself include our scientific literature-based approach as well. Nevertheless, we see the primary role of this literature-based approach as being for evaluation, since we expect results confirmed and published in the scientific literature to necessarily lag behind the initial signals of an ADE likely to appear in EHRs and claims data, in internet searches, and in social media.

# Chapter 5: Conclusion

We considered two contributions for information extraction. On one side, we proposed a language-independent approach to create examples that support an NLP algorithm, which performs information extraction. On the other side, we presented an evaluation method that relies on a language-dependent NLP algorithm to score facts extracted from other information extraction systems.

One of the key challenges for applying learning methods in many real-world problems is the paucity of good quality labeled examples. This is particularly true in information extraction. While semi-supervised learning methods have been developed, in our first approach we explored another alternative method of weak supervision – where the goal is to create examples of reasonable quality that can be relied upon. We considered the information extraction tasks of relation extraction and text categorization to demonstrate the usefulness of the weak supervision. Our key insight is that weak supervision can be provided by a "domain" expert instead of an "NLP" expert and thus the knowledge is independent of the underlying problem but is close to the average human thought process – for example, sports fans. We used the weighted logic representation of Markov Logic networks to model the expert knowledge, learn the weights based on history and make predictions on the unannotated articles. We employed an adapted functional gradient boosting algorithm to learn relational dependency networks for prediciting the target relations. Our approach requires two key elements: world knowledge about the domain, which we argue is the method taken by humans, and evidence to support the world knowledge. The availability or construction of the evidence certainly depends on the domain, and we have provided some examples of knowledge bases that could be relevant.

Our results demonstrate that our method significantly improves the performance thus reducing the need for human annotated examples. This is particularly useful when performing information extraction in large repositories, such as the World Wide Web. Also, in large domains it is hard to design models with prior structure, such as Confitional Random Fields (CRFs) or Markov Random Fields (MRFs), which are modelling methods commonly used for information extraction. Hence, we are interested in learning the structure – dependencies between the random variables – of our model. In our case, random variables correspond to words and phrases, and we learn the dependencies between features such as part-of-speech tags, phrase types, word lemmas, parse trees and dependency paths by using a successful SRL algorithm [23].

In our second contribution, we presented a novel approach that uses information extraction to evaluate other learning methods. We focused on a biomedical domain: Adverse Drug Events (ADEs) discovery. Our method exploits the publicly available biomedical literature to estimate the probability that a drug may cause a certain

event. We do so by using state-of-the-art text mining and multi-relational machine learning techniques. We evaluated our performance on the reference OMOP ground truth. We found agreement better than state-of-the-art ADE discovery methods, and found that in some of the cases of disagreement our method appears to be correct. Nevertheless, we found that in an equal number of cases our method is incorrect. In the remaining cases of disagreement our method had only weak evidence in support of its findings.

Although having some weaknesses, our second contribution demonstrates that employing a language-dependent technique can be useful in certain domains. In the ADE discovery domain, common text patterns are used throughout many articles to express the occurrence of ADEs. This allowed us to capture a set of text patterns, that when used for ADE discovery, get state-of-the-art results. However, language-dependent techniques are not always appropriate, specifically in large domains.

## 5.1 Future Work

There are several directions for future work. Our first approach is closely related to distant supervision methods [21]. So it will be a very interesting future direction to combine the distant and weak supervision examples for structure learning. Combining weak supervision with other advice taking methods, such as transfer learning using constraints [41] or natural language advice based on explicit conditions [18], is another interesting direction. This method can be seen as giving advice about the examples, but Artificial Intelligence (AI) has a long history of using advice on the model, the search space and examples. Hence, combining them might lead to a strong knowledge based system where the knowledge can be provided by a domain expert and not an AI/NLP expert. Moreover, learning the advice, as is done in Torrey et al. [41], is another interesting direction. Furthermore, it is important to evaluate the proposed model in other similar tasks.

In our second problem, the weaknesses mentioned before may be addressed by improvements in the relation extraction knowledge, going deeper in the semantic analysis. Besides detecting conditions followed by the comission of drugs, the extraction knowledge can be extended to consider conditions before taking a drug. Although our model for relation extraction is given, performing parameter and structure learning in the Markov Logic network is an interesting future direction. We set the weights of our MLN clauses manually, but learning these weights can lead to performance improvements. Finally, peforming structure learning, as we did in our first contribution, will be interesting to compare our given model as well as to aim for better results.

# Bibliography


[1] How search works. http://www.google.com/insidesearch/howsearchworks/thestory/, 2013.

[2] H. Blockeel and L. De Raedt. Top-down induction of first-order logical decision trees. *Artificial Intelligence*, 101:285–297, 1998.

[3] R. Bunescu and R. J. Mooney. Statistical relational learning for natural language information extraction. In *Introduction to Statistical Relational Learning*, pages 535–552. 2007.

[4] N. Chambers and D. Jurafsky. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.

[5] R. Clayton. Calculating similarity (part 1): Cosine similarity. http://www.gettingcirrius.com/2010/12/calculating-similarity-part-1-cosine.html, December 2010.

[6] M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, 1999.

[7] T.G. Dietterich, A. Ashenfelter, and Y. Bulatov. Training conditional random fields via gradient tree boosting. In *Proceedings of the International Conference in Machine Learning*, 2004.

[8] P. Domingos and D. Lowd. *Markov Logic: An Interface Layer for AI*. Morgan & Claypool, San Rafael, CA, 2009.

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

[10] J.H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29, 2001.

[11] H. Gurulingappa, J. Fluck, M. Hofmann-Apitius, and L. Toldo. Identification of adverse drug event assertive sentences in medical case reports. In *First International Workshop on Knowledge Discovery in Health Care and Medicine, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2011.

[12] J. H. Gurwitz, T. S. Field, L. R. Harrold, J. Rothschild, K. Debellis, A. C. Seger, C. Cadoret, L. S. Fish, L. Garber, M. Kelleher, and D. W. Bates. Incidence and preventability of adverse drug events among older persons in the ambulatory setting. In *Journal of the American Medical Association*, pages 1107–1116, 2003.

[13] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 2011.

[14] K. Kersting and K. Driessens. Non–parametric policy gradients: A unified treatment of propositional and relational domains. In *Proceedings of the International Conference in Machine Learning*, 2008.

[15] T. Khot, S. Natarajan, K. Kersting, and J. Shavlik. Learning markov logic networks via functional gradient boosting. In *International Conference on Data Mining*, 2011.

[16] J. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii. Overview of BioNLP'09 shared task on event extraction. In *BioNLP 2009 Workshop Companion Volume for Shared Task*, 2009.

[17] D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 423–430, 2003.

[18] G. Kuhlmann, P. Stone, R. J. Mooney, and J. W. Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *AAAI Workshop on Supervisory Control of Learning and Adaptive Systems*, 2004.

[19] D. Lowd and P. Domingos. Efficient weight learning for markov logic networks. In *In Proceedings of the Eleventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 200–211, 2007.

[20] C. Manning. Information extraction and named entity recognition. http://www.stanford.edu/class/cs124/lec/Information_Extraction_and_Named_Entity_Recognition.pptx.

[21] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009.

[22] A. A. Morgan, L. Hirschman, M. Colosimo, A. S. Yeh, and J. B. Colombe. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, 37(6), 2004.

[23] S. Natarajan, T. Khot, K. Kersting, B. Guttmann, and J. Shavlik. Gradient-based boosting for statistical relational learning: The Relational Dependency Network case. *Journal of Machine Learning*, 86(1):25–56, 2012.

[24] F. Niu, C. Ré, A. Doan, and J. W. Shavlik. Tuffy: Scaling up statistical inference in markov logic networks using an RDBMS. *Proceedings of the Very Large Database Endowment*, 4(6):373–384, 2011.

[25] D. Page, V. Santos Costa, S. Natarajan, A. Barnard, P.L. Peissig, and M. Caldwell. Identifying adverse drug events by relational learning. In *AAAI Conference on Artificial Intelligence*, 2012.

[26] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.

[27] H. Poon and P. Domingos. Machine reading: A "killer app" for statistical relational AI. In *Statistical Relational Artificial Intelligence Workshop in AAAI Conference on Artificial Intelligence*, 2010.

[28] H. Poon and L. Vanderwende. Joint inference for knowledge extraction from biomedical literature. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

[29] R. Mooney R. Bunescu. Learning to extract relations from the web using minimal supervision. In *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007.

[30] S. Riedel, H. Chun, T. Takagi, and J. Tsujii. A markov logic approach to bio-molecular event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, 2009.

[31] S. Riedel and A. McCallum. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *Proceedings of BioNLP Shared Task 2011 Workshop*, 2011.

[32] S. Riedel, D. McClosky, M. Surdeanu, A. McCallum, and C. D. Manning. Model combination for event extraction in bionlp 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, 2011.

[33] S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III*, 2010.

[34] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. In *Psychological Review*, pages 386–408, 1958.

[35] P. Ryan, D. Madigan, P. Stang, J. Overhage, J. Racoosin, and A. Hartzema. Empirical assessment of methods for risk identification in healthcare data: results from the experiments of the observational medical outcomes partnership. In *Statistics in Medicine*, 2012.

[36] P. Ryan, E. Welebob, A. Hartzema, P. Stang, and J. Overhage. Surveying us observational data sources and characteristics for drug safety needs. In *Pharmaceutical Medicine*, pages 231–238, 2010.

[37] P. Singla and P. Domingos. Discriminative training of markov logic networks. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, 2005.

[38] M. Surdeanu and M. Ciaramita. Robust information extraction with perceptrons. In *Proceedings of the NIST 2007 Automatic Content Extraction Workshop*, 2007.

[39] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.

[40] S. Takamatsu, I. Sato, and H. Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012.

[41] L. Torrey, J. Shavlik, T. Walker, and R. Maclin. Transfer learning via advice taking. In *Advances in Machine Learning I*, pages 147–170. 2010.

[42] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, 2007.

[43] R. White, N. Tatonetti, N. Shah, R. Altman, and E. Horvitz. Web-scale pharmacovigilance: listening to signals from the crowd. In *Journal of the American Medical Informatics Association*, 2013.

[44] K. Yoshikawa, S. Riedel, M. Asahara, and Y. Matsumoto. Jointly identifying temporal relations with markov logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, 2009.

[45] G. Zhou, J. Su, J. Zhang, and M. Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

[46] G. Zhou, M. Zhang, D. H. Ji, and Q. Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.

**Jose Manuel Picado Leiva**
3307 University Parkway , Winston-Salem, NC 27106 , jpicado@gmail.com , +1-336-655-0443

**Research Interests**

Artificial intelligence, machine learning, natural language processing, information extraction, statistical relational learning.

**Education**

**Wake Forest University** — Winston-Salem, NC
M.S. in Computer Science — May 2013
Thesis Topic: Efficient Information Extraction Using Statistical Relational Learning
Adviser: Dr. Sriraam Natarajan
GPA: 4.0

**Costa Rica Institute of Technology** — Cartago, Costa Rica
B.S. in Computer Science — February 2011
GPA: 89.93/100

**Experience**

**Wake Forest University** — Winston-Salem, NC
*Research Assistant* — January 2012 - Present

- Developed an information extraction system using commonsense knowledge and weak supervision for relation extraction and document classification.
- Developed a medical event verifier, which takes as input adverse drugs events and assigns scores based on text patterns and similarities with literature found on the web.
- Used a statistical relational learning algorithm to predict coronary heart diseases in a group of individuals.

*Teaching Assistant* — September 2011 - December 2011

- Attended lab sessions and maintained office hours in which undergraduate students asked questions regarding computer architecture, systems, theory, logic, algorithms, and programming.
- Graded lab reports and exams.

**Avantica Technologies** — San Jose, Costa Rica
*Software Engineer* — July 2010 - May 2011

- Developed plugins in Perl and Java for Electric Commander, an integrated building tool developed by Electric Cloud.
- Performed analysis, design, development, testing, and deployment of plugins for the following tools: VMware Lab Manager, VMware ESX, Microsoft Hyper-V, Amazon EC2, Oracle VM VirtualBox, NAnt, Sonar.

**Costa Rica Institute of Technology** — Cartago, Costa Rica
*Research Assistant* — July 2009 - June 2010

- Implemented an OptiPortal cluster for grid visualization using Rocks Cluster and Viz Roll.
- Developed an animation framework used for the development of an animated short film using Ray Tracing image generation.

**Research Papers**

**Jose Picado**, Sriraam Natarajan, Vitor Santos Costa, David Page, Michael Caldwell. A Novel NLP-Based Method for Evaluation of Adverse Drug Event Discovery, *American Medical Informatics Association Symposium*, 2013 (under review).

Sriraam Natarajan, **Jose Picado**, Tushar Khot, Kristian Kersting, Christopher Re, and Jude Shavlik. Using Commonsense Knowledge to Automatically Create (Noisy) Training Examples from Text, *International Workshop on Statistical Relational Artificial Intelligence*, 2013.

**Awards** Upsilon Pi Epsilon, Wake Forest University, 2012.
Academic Honors Scholarship, Costa Rica Institute of Technology, 2008-2010.

**Languages** Fluent in English and Spanish.

**Technical Skills** Java, C, C#, Objective-C, Perl, PHP, JavaScript, HTML/CSS, XML, SQL, CUDA.