Medical Relation Extraction with Manifold Models

Chang Wang
IBM T. J. Watson Research Center
Yorktown Heights, New York, 10598
changwangnk@gmail.com

James Fan
IBM T. J. Watson Research Center
Yorktown Heights, New York, 10598
fanj@us.ibm.com

Abstract

In this paper, we present a manifold model for medical relation extraction. Our model is built upon a medical corpus containing 80M sentences (11 gigabyte text) and designed to accurately and efficiently detect the key medical relations that can facilitate clinical decision making. Our approach integrates domain specific parsing and typing systems, and can utilize labeled as well as unlabeled examples. To provide users with more flexibility, we also take label weight into consideration. Effectiveness of our model is demonstrated both theoretically with a proof to show that the solution is a closed-form solution and experimentally with positive results in experiments.

1 Introduction

There exists a vast amount of knowledge sources and ontologies in the medical domain. Such information is also growing and changing extremely quickly, making the information difficult for people to read, process and remember. The combination of recent developments in information extraction and the availability of unparalleled medical resources thus offers us the unique opportunity to develop new techniques to help healthcare professionals overcome the cognitive challenges they face in clinical decision making.

Relation extraction plays a key role in information extraction. Using question answering as an example (Wang et al., 2012): in question analysis, the semantic relations between the question focus and each term in the clue can be used to identify the weight of each term so that better search queries can be generated. In candidate answer generation, relations enable the background knowledge base to be used for potential candidate answer generation. In candidate answer scoring, relation-based matching algorithms can go beyond explicit lexical and syntactic information to detect implicit semantic relations shared across the question and passages.

To construct a medical relation extraction system, several challenges have to be addressed:

- The first challenge is how to identify a set of relations that has sufficient coverage in the medical domain. To address this issue, we study a real-world diagnosis related question set and identify a set of relations that has a good coverage of the clinical questions.

- The second challenge is how to efficiently detect relations in a large amount of medical text. The medical corpus underlying our relation extraction system contains 80M sentences (11 gigabytes pure text). To extract relations from a dataset at this scale, the relation detectors have to be fast. In this paper, we speed up relation detectors through parsing adaptation and replacing non-linear classifiers with linear classifiers.

- The third challenge is that the labeled relation examples are often insufficient due to the high labeling cost. When we build a naïve model to detect relations, the model tends to overfit for the labeled data. To address this issue, we develop a manifold model (Belkin et al., 2006) that encourages examples (including both labeled and unlabeled examples) with similar contents to be assigned with similar scores. Our model goes beyond regular regression models in that it applies constraints to those coefficients, such that the topology of the given data manifold will be respected. Computing the optimal weights in a regression model and preserving manifold topology are conflicting objectives, we

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 828–838,
Baltimore, Maryland, USA, June 23-25 2014.

present a closed-form solution to ideally balance these two goals.

The contributions of this paper on medical relation extraction are three-fold:

- The problem setup is new. There is a "fundamental" difference between our problem setup and the conventional setups, like i2b2 (Uzuner et al., 2011). In i2b2 relation extraction task, entity mentions are *manually* labeled, and each mention has 1 of 3 concepts: 'treatment', 'problem', and 'test'. To resemble real-world medical relation extraction challenges where perfect entity mentions do not exist, our new setup requires the entity mentions to be *automatically* detected. The most well-known tool to detect medical entity mentions is MetaMap (Aronson, 2001), which considers all terms as entities and automatically associates each term with a number of concepts from UMLS CUI dictionary (Lindberg et al., 1993) with more than 2.7 million distinct concepts (compared to 3 in i2b2). The huge amount of entity mentions, concepts and noisy concept assignments provide a tough situation that people have to face in real-world applications.
- From the perspective of relation extraction applications, we identify "super relations"- the key relations that can facilitate clinical decision making (Table 1). We also present approaches to collect training data for these relations with a small amount of labeling effort.
- From the perspective of relation extraction methodologies, we present a manifold model for relation extraction utilizing both labeled and unlabeled data. Our approach can also take the label weight into consideration.

The experimental results show that our relation detectors are fast and outperform the state-of-the-art approaches on medical relation extraction by a large margin. We also apply our model to build a new medical relation knowledge base as a complement to the existing knowledge bases.

2 Background

2.1 Medical Ontologies and Sources

Medical domain has a huge amount of natural language content found in textbooks, encyclopedias, guidelines, electronic medical records, and many other sources. It is also growing at an extremely high speed. Substantial understanding of the medical domain has already been included in the Unified Medical Language System (UMLS) (Lindberg et al., 1993), which includes medical concepts, relations, definitions, etc. The 2012 version of the UMLS contains information about more than 2.7 million concepts from over 160 source vocabularies. Softwares for using this knowledge also exist: MetaMap (Aronson, 2001) is able to identify concepts in text. SEMREP (Rindflesch and Fiszman, 2003) can detect some relations using hand-crafted rules.

2.2 Relation Extraction

To extract semantic relations from text, three types of approaches have been applied. Rule-based methods (Miller et al., 2000) employ a number of linguistic rules to capture relation patterns. Feature-based methods (Kambhatla, 2004; Zhao and Grishman, 2005) transform relation instances into a large amount of linguistic features like lexical, syntactic and semantic features, and capture the similarity between these feature vectors. Recent results mainly rely on kernel-based methods. Many of them focus on using tree kernels to learn parse tree structure related features (Collins and Duffy, 2001; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005).

Other researchers study how different approaches can be combined to improve the extraction performance. For example, by combining tree kernels and convolution string kernels, (Zhang et al., 2006) achieved the state of the art performance on ACE data (ACE, 2004). Recently, "distant supervision" has emerged to be a popular choice for training relation extractors without using manually labeled data (Mintz et al., 2009; Jiang, 2009; Chan and Roth, 2010; Wang et al., 2011; Riedel et al., 2010; Ji et al., 2011; Hoffmann et al., 2011; Surdeanu et al., 2012; Takamatsu et al., 2012; Min et al., 2013).

Various relation extraction approaches have been adapted to the medical domain, most of which focus on designing heuristic rules targeted for diagnosis and integrating the medical ontology in the existing extraction approaches. Results of some of these approaches are reported on the i2b2 data (Uzuner et al., 2011).

3 Identifying Key Medical Relations

3.1 Super Relations in Medical Domain

The first step in building a relation extraction system for medical domain is to identify the relations that are important for clinical decision making.

Four main clinical tasks that physicians engage in are discussed in (Demner-Fushman and Lin, 2007). They are **Therapy**- select treatments to offer a patient, taking consideration of effectiveness, risk, cost and other factors (prevention is under the general category of Therapy), **Diagnosis** (including differential diagnosis based on findings and diagnostic test), **Etiology**- identify the factors that cause the disease and **Prognosis**- estimate the patient's likely course over time. These activities can be translated into "search tasks". For example, the search for therapy is usually the therapy selection given a disease.

We did an independent study regarding what clinical questions usually ask for on a set of 5,000 Doctor Dilemma (DD) questions from the American College of Physicians (ACP). This set includes questions about diseases, treatments, lab tests, and general facts[1]. Our analysis shows that about 15% of these questions ask for treatments, preventions or contraindicated drugs for a disease or another way around, 4% are about diagnosis tests, 6% are about the causes of a disease, 1% are about the locations of a disease, 25% are about the symptoms of a disease, 8% are asking for definitions, 7% are about guidelines and the remaining 34% questions either express no relations or some relations that are not very popular.

Based on the analysis in (Demner-Fushman and Lin, 2007) and our own results, we decided to focus on seven key relations in the medical domain, which are described in Table 1. We call these relations "super relations", since they cover most questions in the DD question set and align well with the analysis result in (Demner-Fushman and Lin, 2007).

3.2 Collect Training Data

This section presents how we collect training data for each relation. The overall procedure is illustrated in Figure 1.

[1]Here's an example of these questions and its answer: **Question:** The syndrome characterized by joint pain, abdominal pain, palpable purpura, and a nephritic sediment. **Answer:** Henoch-Schonlein purpura.

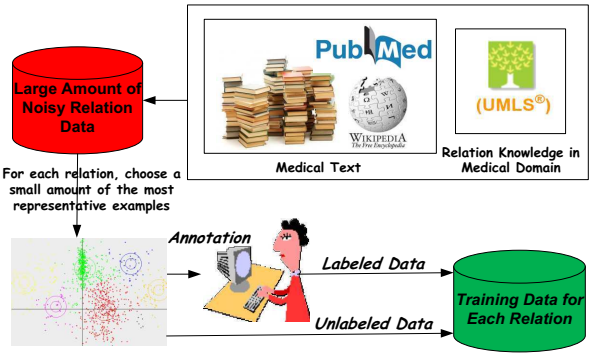

Figure 1: Collect Training Data

Our medical corpus has incorporated a set of medical books/journals[2] and MEDLINE abstracts. We also complemented these sources with Wikipedia articles. In total, the corpus contains 80M sentences (11 gigabyte pure text).

The UMLS 2012 Release contains more than 600 relations and 50M relation instances under around 15 categories. The RO category (RO stands for "has Relationship Other than synonymous, narrower, or broader") is the most interesting one, and covers relations like "may_treat", "has_finding_site", etc. Each relation has a certain number of Concept Unique Identifier (CUI) pairs that are known to bear that relation. In UMLS, some relation information is redundant. Firstly, half of these relations are simply inverse of each other (e.g. the relation "may_treat" and "may_be_treated_by"). Secondly, there is a significant amount of redundancy even among non-inverse relations (e.g. the relation "has_manifestation" and "disease_has_finding").

From UMLS relations, we manually chose a subset of them that are directly related to the super relations discussed in Section 3.1. The correspondences between them are given in Table 1. One thing to note is that super relations are more general than the UMLS relations, and one super relation might integrate multiple UMLS relations. Using the CUI pairs in the UMLS relation knowl-

[2]This is a full list of the books and journals used in our corpus: ACP-Medical Knowledge Self-Assessment Program, EBSCO-Dynamed, EBSCO-Quick Lessons, EBSCO-EBCS, EBSCO-Clinical Review, Wiley-Essential Evidence Plus: EBMG Guidelines, Wiley-Essential Evidence Topics, Wiley-Essential Evidence Plus: EBMG Summaries, Wiley-POEMs, Wiley-The Breast Journal, New England Journal of Medicine, Journal Watch, NCCN-CME, NCCN-GUS, NCCN-Compendium, NCCN-Templates, NCCN-Guidelines for Patients, NCCN-Physician Guidelines, Merck Manual of Diagnosis and Therapy, and UpToDate.

Table 1: Super relations & their arguments, UMLS sources and noise% in the annotation data

Super Relations	Argument 1	Argument 2	UMLS Sources	Noise% in Annotation Data
treats	disease	treatments	may_treat, treats	16%
prevents	disease	treatments	may_prevent	49%
contraindicates	disease	treatments	contraindicated_drug	97%
diagnoses	disease	tests	may_diagnose	63%
causes	disease	causes	cause_of, causative_agent_of	66%
location_of	disease	locations	has_finding_site disease_has_primary_anatomic_site	41%
symptom_of	disease	symptoms	disease_has_finding disease_may_have_finding has_manifestation has_definitional_manifestation	66%

edge base, we associate each super relation with a set of CUI pairs.

To collect the training data for each super relation, we need to collect sentences that express the relation. To achieve this, we parsed all 80M sentences in our medical corpus, looking for the sentences containing the terms that are associated with the CUI pairs in the knowledge base. This (distant supervision) approach resulted in a huge amount of sentences that contain the desired relations, but also brought in a lot of noise in the form of false positives. For example, we know from the knowledge base that "antibiotic drug" may treat "Lyme disease". However the sentence "This paper studies the relationship between antibiotic drug and Lyme disease" contains both terms but does not express the "treats" relation.

The most reliable way to clean the training data is to ask annotators to go through the sentences and assign the sentences with positive/negative labels. However, it will not work well when we have millions of sentences to vet. To minimize the human labeling effort, we ran a K-medoids clustering on the sentences associated with each super relation and kept the cluster centers as the most representative sentences for annotation. Depending on the number of the sentences we collected for each relation, the #clusters was chosen from 3,000 - 6,000. The similarity of two sentences is defined as the bag-of-words similarity of the dependency paths connecting arguments. Part of the resulting data was manually vetted by our annotators, and the remaining was held as unlabeled data for further experiments.

Our relation annotation task is quite straightforward, since both arguments are given and the decision is a Yes-or-No decision. The noise rate of each relation (#sentences expressing the relation / #sentences) is reported in Table 1 based on the annotation results. The noise rates differ significantly from one relation to another. For "treats" relation, only 16% of the sentences are false positives. For "contraindicates" relation, the noise rate is 97%.

To grow the size of the negative training set for each super relation, we also added a small amount of the most representative examples (also coming from K-medoids clustering) from each unrelated UMLS relation to the training set as negative examples. This resulted in more than 10,000 extra negative examples for each relation.

3.3 Parsing and Typing

The most well-known tool to detect medical entity mentions is MetaMap (Aronson, 2001), which considers all terms as entities and automatically associates each term with a number of concepts from UMLS CUI dictionary (Lindberg et al., 1993) with 2.7 million distinct concepts.

The parser used in our system is MedicalESG, an adaptation of ESG (English Slot Grammar) (McCord et al., 2012) to the medical domain with extensions of medical lexicons integrated in the UMLS 2012 Release. Compared to MetaMap, MedicalESG is based on the same medical lexicons, 10 times faster and produces very similar parsing results.

We use the semantic types defined in UMLS (Lindberg et al., 1993) to categorize argument types. The UMLS consists of a set of 133 subject categories, or semantic types, that provide a consistent categorization of more than 2M concepts represented in the UMLS Metathesaurus. Our system assigns each relation argument with one or more UMLS semantic types through a two step process. Firstly, we use MedicalESG to process the input sentence, identify segments of text that correspond to concepts in

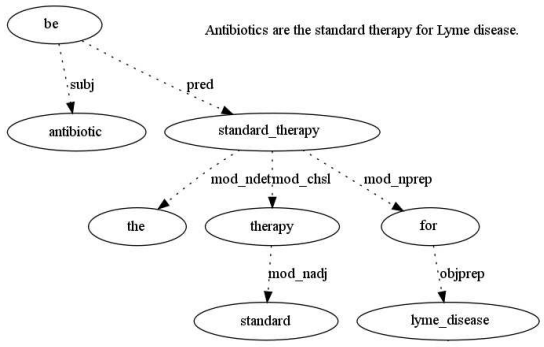

Figure 2: A Parse Tree Example

the UMLS Metathesaurus and associate each of them with one or more UMLS CUIs (Concept Unique Identifier). Then we do a CUI lookup in UMLS to find the corresponding semantic types for each CUI.

Most relation arguments are associated with multiple semantic types. For example, the term "tetracycline hydrochloride" has two types: "Organic Chemical" and "Antibiotic". Sometimes, the semantic types are noisy due to ambiguity of terms. For example, the term "Hepatitis b" is associated with both "Pharmacologic Substance" and "Disease or Syndrome" based on UMLS. The reason for this is that people use "Hepatitis b" to represent both "the disease of Hepatitis b" and "Hepatitis b vaccine", so UMLS assigns both types to it. This is a concern for relation extraction, since two types bear opposite meanings. Our current strategy is to integrate all associated types, and rely on the relation detector trained with the labeled data to decide how to weight different types based upon the context.

Here is an illustrative example. Consider the sentence: *"Antibiotics are the standard therapy for Lyme disease"*: MedicalESG first generates a dependency parse tree (Figure 2) to represent grammatical relations between the words in the sentence, and then associates the words with CUIs. For example, "Antibiotics" is associated with CUI "C0003232" and "Lyme disease" is associated with two CUIs: "C0024198" and "C0717360". CUI lookup will assign "Antibiotics" with a semantic type "Antibiotic", and "Lyme disease" with three semantic types: "Disease or Syndrome", "Pharmacologic Substance" and "Immunologic Factor". This sentence expresses a "treats" relation between "Antibiotics" and "Lyme disease".

4 Relation Extraction with Manifold Models

4.1 Motivations

Given a few labeled examples and many unlabeled examples for a relation, we want to build a relation detector leveraging both labeled and unlabeled data. Following the manifold regularization idea (Belkin et al., 2006), our strategy is to learn a function that assigns a score to each example. Scores are fit so that examples (both labeled and unlabeled) with similar content get similar scores, and scores of labeled examples are close to their labels. Integration of the unlabeled data can help solve overfitting problems when the labeled data is not sufficient.

4.2 Features

We use 8 groups of features to represent each relation example. These features are commonly used for relation extraction.

- (1) Semantic types of argument 1, such as "Antibiotic".
- (2) Semantic types of argument 2.
- (3) Syntactic features representing the dependency path between two arguments, such as "subj", "pred", "mod_nprep" and "objprep" (between arguments "antibiotic" and "lyme disease") in Figure 2.
- (4) Features modeling the incoming and outgoing links of both arguments. These features are useful to determine if a relation goes from argument 1 to argument 2 or vice versa.
- (5) Topic features modeling the words in the dependency path. In the example given in Figure 2, the dependency path contains the following words: "be", "standard therapy" and "for". These features as well as the features in (6) are achieved by projecting the words onto a 100 dimensional LSI topic space (Deerwester et al., 1990) constructed from our medical corpus.
- (6) Topic features modeling the words in the whole sentence.
- (7) Bag-of-words features modeling the dependency path. In (7) and (8), we only consider the words that have occurred in the positive training data.

Notations:

The input dataset $X = \{x_1, \cdots, x_m\}$ is represented as a feature-instance matrix.

The desired label vector $Y = \{y_1, \cdots, y_l\}$ represents the labels of $\{x_1, \cdots, x_l\}$, where $l \leq m$.

W is a weight matrix, where $W_{i,j} = e^{-\|x_i - x_j\|^2}$ models the similarity of x_i and x_j.

$\|x_i - x_j\|$ stands for the Euclidean distance between x_i and x_j in the vector space.

D is a diagonal matrix: $D_{i,i} = \sum_j W_{i,j}$.

$\mathcal{L} = D^{-0.5}(D - W)D^{-0.5}$ is called normalized graph Laplacian matrix.

Δ is a user defined $l \times l$ diagonal matrix, where Δ_i represents the weight of label y_i.

$\mathcal{A} = \begin{pmatrix} \Delta & 0 \\ 0 & 0 \end{pmatrix}$ is an $m \times m$ matrix.

$V = [y_1, \cdots y_l, 0, \cdots, 0]$ is a $1 \times m$ matrix.

μ is a weight scalar.

$()^+$ represents pseudo inverse.

Algorithm:

1. **Represent each example using features: $X = \{x_1, \cdots, x_m\}$, where x_i is the ith example.**
2. **Construct graph Laplacian matrix $\mathcal{L}$ modeling the data manifold.**
3. **Construct vector $V = [y_1, \cdots y_l, 0, \cdots, 0]$.**
4. **Compute projection function f for each relation: $f = (X(\mathcal{A} + \mu\mathcal{L})X^T)^+ X\mathcal{A}V^T$.**

Figure 3: Notations and the Algorithm to Train a Manifold Model for Relation Extraction

- (8) Bag-of-words features modeling the whole sentence.

In relation extraction, many recent approaches use non-linear kernels to get the similarity of two relation examples. To classify a relation example, a lot of dot product computations are required. This is very time consuming and becomes a bottleneck in using relation extraction to facilitate clinical decision making. To speed up the classifier during the apply time, we decided to use a linear classifier instead of non-linear classifiers.

We represent all features in a single feature space. For example, we use a vector of 133 entries (UMLS contains 133 semantic types) to represent the types of argument 1. If argument 1 is associated with two types: "Organic Chemical" and "Antibiotic", we set the two corresponding entries to 1 and all the other entries to 0. Similar approaches are used to represent the other features.

4.3 The Main Algorithm

The problem we want to solve is formalized as follows: given a relation dataset $X = \{x_1, \cdots, x_m\}$, and the desired label $Y = \{y_1, \cdots, y_l\}$ for $\{x_1, \cdots, x_l\}$, where $l \leq m$, we want to construct a mapping function f to project any example x_i to a new space, where $f^T x_i$ matches x_i's desired label y_i. In addition, we also want f to preserve the manifold topology of the dataset, such that similar examples (both labeled and unlabeled) get similar scores. Here, the label is '+1' for positive examples, and '-1' for negative examples. Notations and the main algorithm to construct f for each relation are given in Figure 3.

4.4 Justification

The solution to the problem defined in Section 4.3 is given by the mapping function f to minimize the following cost function:

$$C(f) = \sum_{i \leq l} \alpha_i (f^T x_i - y_i)^2 + \mu \sum_{i,j} W_{i,j} (f^T x_i - f^T x_j)^2.$$

The first term of $C(f)$ is based on labeled examples, and penalizes the difference between the mapping result of x_i and its desired label y_i. α_i is a user specified parameter, representing the weight of label y_i. The second term of $C(f)$ does not take label information into account. It encourages the neighborhood relationship (geometry of the manifold) within X to be preserved in the mapping. When x_i and x_j are similar, the corresponding $W_{i,j}$ is big. If f maps x_i and x_j to different positions, f will be penalized. The second term is useful to bound the mapping function f and prevents overfitting from happening. Here μ is the weight of the second term. When $\mu = 0$, the model disregards the unlabeled data, and the data manifold topology is not respected.

Compared to manifold regularization (Belkin et al., 2006), we do not include the RKHS norm term. Instead, we associate each labeled example with an extra weight for label confidence. This weight is particularly useful when the training data comes from "Crowdsourcing", where we ask

multiple workers to complete the same task to correct errors. In that scenario, weights can be assigned to labels based upon annotator agreement.

Theorem 1: $f = (X(\mathcal{A} + \mu\mathcal{L})X^T)^+ X\mathcal{A}V^T$ minimizes the cost function $C(f)$.
Proof:
Given the input X, we want to find the optimal mapping function f such that $C(f)$ is minimized:

$$f = \arg\min_f C(f).$$

It can be verified that

$$\sum_{i \leq l} \alpha_i (f^T x_i - y_i)^2 = f^T X\mathcal{A}X^T f - 2f^T X\mathcal{A}V^T + V\mathcal{A}V^T.$$

We can also verify that

$$\mu \sum_{i,j} (f^T x_i - f^T x_j)^2 W_{i,j} = \mu f^T X\mathcal{L}X^T f.$$

So $C(f)$ can be written as

$$f^T X\mathcal{A}X^T f - 2f^T X\mathcal{A}V^T + V\mathcal{A}V^T + \mu f^T X\mathcal{L}X^T f.$$

Using the Lagrange multiplier trick to differentiate $C(f)$ with respect to f, we have

$$2X\mathcal{A}X^T f + 2\mu X\mathcal{L}X^T f = 2X\mathcal{A}V^T.$$

This implies that

$$X(\mathcal{A} + \mu\mathcal{L})X^T f = X\mathcal{A}V^T.$$

So

$$f = (X(\mathcal{A} + \mu\mathcal{L})X^T)^+ X\mathcal{A}V^T,$$

where "+" represents pseudo inverse. ∎

4.5 Advantages

Our algorithm offers the following advantages:

- The algorithm exploits unlabeled data, which helps prevent "overfitting" from happening.
- The algorithm provides users with the flexibility to assign different labels with different weights. This feature is useful when the training data comes from "crowdsourcing" or "distant supervision".
- Different from many approaches in this area, our algorithm provides a closed-form solution of the result. The solution is global optimal regarding the cost function $C(f)$.
- The algorithm is computationally efficient at the apply time (as fast as linear regressions).

5 Experiments

5.1 Cross-Validation Test

We use a cross-validation test[3] with the relation data generated in Section 3.2 to compare our approaches against the state-of-the-art approaches. The task is to classify the examples into positive or negative for each relation. We applied a 5-fold cross-validation. In each round of validation, we used 20% of the data for training and 80% for testing. The F_1 scores reported here are the average of all 5 rounds. We used MedicalESG to process the input text for all approaches.

5.1.1 Data and Parameters

This dataset includes 7 relations. We do not consider the relation of "contraindicates" in this test, since it has too few positive examples. On average, each relation contains about 800 positive examples and more than 13,000 negative examples. To eliminate the examples that are trivial to classify, we removed the negative examples that do not bear the valid argument types. This removed the examples that can be easily classified by a type filter, resulting in 3,000 negatives on average per relation. For each relation, we also collected 5,000 unlabeled examples and put them into two sets: unlabeled set 1 and 2 (2,500 examples in each set).

No parameter tuning was taken and no relation specific heuristic rules were applied in all tests. In all manifold models, $\mu = 1$. In SVM implementations, the trade-off parameter between training error and margin was set to 1 for all experiments.

5.1.2 Baseline Approaches

We compare our approaches to three state-of-the-art approaches including SVM with convolution tree kernels (Collins and Duffy, 2001), linear regression and SVM with linear kernels (Schölkopf and Smola, 2002). To adapt the tree kernel to medical domain, we followed the approach in (Nguyen et al., 2009) to take the syntactic structures into consideration. We also added the argument types as features to the tree kernel. In the tree kernel implementation, we assigned the tree structure and the vector corresponding to the argument types

[3] If we take the perfect entity mentions and the associated concepts provided by i2b2 (Uzuner et al., 2011) as the input, our system can directly apply to i2b2 relation extraction data. However, the i2b2 data has a tough data use agreement. Our legal team held several rounds of negotiations with the i2b2 data owner and then decided we should not use it due to the high legal risks. We are not aware of other available medical relation extraction datasets that fit for our evaluations.

Table 2: F_1 Scores from a Five-Fold Cross Validation Experiment

	SVM Tree Kernel	SVM Linear Kernel	Linear Regression	Manifold *Unlabeled*	Manifold *Predicted Labels with Weights*	Manifold *Predicted Labels without Weights*	Manifold *Unlabeled+Predicted Labels with Weights*
treats	0.7648	0.7850	0.7267	0.8025	0.8041	0.7884	**0.8085**
prevents	0.2859	0.3887	0.3922	0.5502	0.5696	**0.6349**	0.6332
causes	0.3885	0.5024	0.5219	**0.5779**	0.5088	0.3978	0.5081
location_of	0.6113	0.6009	0.4968	0.7275	0.7363	0.6964	**0.7454**
diagnoses	0.5520	0.4934	0.3202	0.6468	0.6485	0.5720	**0.6954**
symptom_of	0.4398	0.5611	0.5984	**0.6347**	0.5314	0.4515	0.5968
average	0.5071	0.5553	0.5094	0.6566	0.6331	0.5902	**0.6646**

with equal weights. The SVM with linear kernels and the linear regression model used the same features as the manifold models.

5.1.3 Settings for the Manifold Models

We tested our manifold model for each relation under three different settings:

(1) Manifold *Unlabeled*: We combined the labeled data and unlabeled set 1 in training. We set $\alpha_i = 1$ for $i \in [1, l]$.

(2) Manifold *Predicted Labels*: We combined labeled data and unlabeled set 2 in training. $\alpha_i = 1$ for $i \in [1, l]$. Different from the previous setting, we gave a label estimation to all the examples in the unlabeled set 2 based on the noise rate (Noise%) from Table 1. The label of all unlabeled examples was set to "+1" when $100\% - 2 \cdot Noise\% > 0$, or "-1" otherwise. Two weighting strategies were applied:

- *With Weights:* We let label weight $\alpha_i = |100\% - 2 \cdot Noise\%|$ for all x_i coming from the unlabeled set 2. This setting represents an empirical rule to estimate the label and confidence of each unlabeled example based on the sampling result.
- *Without Weights:* α_i is always set to 1.

(3) Manifold *UnLabeled+Predicted Labels*: a combination of setting (1) and (2). In this setting, the data from unlabeled set 1 was used as unlabeled data and the data from unlabeled set 2 was used as labeled data (With Weights).

5.1.4 Results

The results are summarized in Table 2.

The tree kernel-based approach and linear regression achieved similar F_1 scores, while linear SVM made a 5% improvement over them. One thing to note is that the results from these approaches vary significantly. The reason for this is that the labeled training data is not sufficient. So the approaches that completely depend on the labeled data are likely to run into overfitting. Linear SVM performed better than the other two, since the large-margin constraint together with the linear model constraint can alleviate overfitting.

By integrating unlabeled data, the manifold model under setting (1) made a 15% improvement over linear regression model on F_1 score, where the improvement was significant across all relations.

Under setting (2), the *With Weights* strategy achieved a slightly worse F_1 score than the previous setting but much better result than the baseline approaches. This tells us that estimating the label of unlabeled examples based upon the sampling result is one way to utilize unlabeled data and may help improve the relation extraction results. The results also show that the label weight is important for this setting, since the *Without Weights* strategy did not perform very well.

Compared to setting (1) and (2), setting (3) made use of 2,500 more unlabeled examples, and achieved the best performance among all approaches. On one hand, this result shows that using more unlabeled data can further improve the result. On the other hand, the insignificant improvement over (1) and (2) strongly indicates that how to utilize more unlabeled data to achieve a significant improvement is non-trivial and deserves more attention. To what extensions the unlabeled data can help the learning process is an open problem. Generally speaking, when the existing data is sufficient to characterize the dataset geometry, adding more unlabeled data will not help (Singh et al., 2008).

We tested the tree kernel-based approach without integrating the medical types as well. That resulted in very poor performance: the average F_1 score was below 30%. We also applied the rules used in SEMREP (Rindflesch and Fiszman, 2003) to this dataset. Since the relations detected by

SEMREP rules cannot be perfectly aligned with super relations, we cannot directly compare the results. Overall speaking, SEMREP rules are very conservative and detect very few relations from the same text.

5.2 Knowledge Base (KB) Construction

The UMLS Metathesaurus (Lindberg et al., 1993) contains a large amount of manually extracted relation knowledge. Such knowledge is invaluable for people to collect training data to build new relation detectors. One downside of using this KB is its incompleteness. For example, it only contains the treatments for about 8,000 diseases, which are far from sufficient. Further, the medical knowledge is changing extremely quickly, making people hard to understand it, and update it in the knowledge base in a timely manner.

To address these challenges, we constructed our own relation KB as a complement to the UMLS relation KB. We directly ran our relation detectors (trained with all labeled and unlabeled examples) on our medical corpus to extract relations. Then we combined the results and put them in a new KB. The new KB covers all super relations and stores the knowledge in the format of (relation_name, argument_1, argument_2, confidence), where the confidence is computed based on the relation detector confidence score and relation popularity in the corpus. The most recent version of our relation KB contains 3.4 million such entries.

We compared this new KB against UMLS KB using an answer generation task on a set of 742 Doctor Dilemma questions. We first ran our relation detectors to detect the relation(s) in the question clue involving question focus (what the question asks for). Then we searched against both KBs using the relation name and the non-focus argument for the missing argument. The search results were then generated as potential answers. We used the same relations to do KB lookup, so the results are directly comparable. Since most questions only have one correct answer, the precision number is not very important in this experiment.

If we detect multiple relations in the question, and the same answer is generated from more than one relations, we sum up all those confidence scores to make such answers more preferable. Sometimes, we may generate too many answers from KBs. For example, if the detected relation is "location_of" and the non-focus argument is "skin", then thousands of answers can be generated. In this scenario, we sort the answers based upon the confidence scores and only consider up to p answers for each question. In our test, we considered three numbers for p: 20, 50 and 3,000.

From Table 3, we can see that the new KB outperforms the most popularly-used UMLS KB at all recall levels by a large margin. This result indicates that the new KB has a much better knowledge coverage. The UMLS KB is manually created and thus more precise. In our experiment, the UMLS KB generated fewer answers than the new KB. For example, when up to 20 answers were generated for each question, the UMLS KB generated around 4,700 answers for the whole question set, while the new KB generated about 7,600 answers.

Construction of the new KB cost 16 machines (using 4×2.8G cores per machine) 8 hours. The reported computation time is for the whole corpus with 11G pure text.

Table 3: Knowledge Base Comparison

	Recall@20	Recall@50	Recall@3000
Our KB	135/742	182/742	301/742
UMLS KB	42/742	52/742	73/742

6 Conclusions

In this paper, we identify a list of key relations that can facilitate clinical decision making. We also present a new manifold model to efficiently extract these relations from text. Our model is developed to utilize both labeled and unlabeled examples. It further provides users with the flexibility to take label weight into consideration. Effectiveness of the new model is demonstrated both theoretically and experimentally. We apply the new model to construct a relation knowledge base (KB), and use it as a complement to the existing manually created KBs.

Acknowledgments

We thank Siddharth Patwardhan for help on tree kernels, Sugato Bagchi and Dr. Herbert Chase's team for categorizing the Doctor Dilemma questions. We also thank Anthony Levas, Karen Ingraffea, Mark Mergen, Katherine Modzelewski, Jonathan Hodax, Matthew Schoenfeld and Adarsh Thaker for vetting the training data.

References

ACE. 2004. The automatic content extraction projects, http://projects.ldc.upenn.edu/ace/.

A. Aronson. 2001. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In *Proceedings of the 2001 Annual Symposium of the American Medical Informatics Association*.

M. Belkin, P. Niyogi, and V. Sindhwani. 2006. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, pages 2399–2434.

R. Bunescu and R. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

Y. Chan and D. Roth. 2010. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 152–160.

M. Collins and N. Duffy. 2001. Convolution kernels for natural language. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 625–632.

A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–429.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

D. Demner-Fushman and J. Lin. 2007. Answering clinical questions with knowledge-based and statistical techniques. *Journal of Computational Linguistics*, 56:63–103.

R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

H. Ji, R. Grishman, and H. T. Dang. 2011. Overview of the TAC 2011 knowledge base population track. In *Proceedings of the Text Analytics Conference*.

J. Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1012–1020.

N. Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*.

D. Lindberg, B. Humphreys, and A. McCray. 1993. The Unified Medical Language System. *Methods of Information in Medicine*, 32:281–291.

M. McCord, J. W. Murdock, and B. K. Boguraev. 2012. Deep parsing in Watson. *IBM Journal of Research and Development*, 56.

S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*.

B. Min, R. Grishman, L. Wan, C. Wang, and D. Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1003–1011.

T. Nguyen, A. Moschitti, and G. Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

S. Riedel, L. Yao, and A. McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*.

T. C. Rindflesch and M. Fiszman. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of Biomedical Informatics*, 36:462–477.

B. Schölkopf and A. J. Smola. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

A. Singh, R. D. Nowak, and X. Zhu. 2008. Unlabeled data: now it helps, now it doesnot. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*.

M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. 2012. Multi-instance multilabel learning for relation extraction. In *Proceedings of the 2012*

Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP).

S. Takamatsu, I. Sato, and H. Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of American Medical Informatics Association*, 18:552–556.

C. Wang, J. Fan, A. Kalyanpur, and D. Gondek. 2011. Relation extraction with relation topics. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

C. Wang, A. Kalyanpur, J. Fan, B. Boguraev, and D. Gondek. 2012. Relation extraction and scoring in DeepQA. *IBM Journal of Research and Development*, 56.

M. Zhang, J. Zhang, J. Su, and G. Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL)*.

S. Zhao and R. Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 419–426.