Journal of Biomedical Informatics 46 (2013) 275–285

Contents lists available at SciVerse ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Detecting concept relations in clinical text: Insights from a state-of-the-art model

Xiaodan Zhu *, Colin Cherry, Svetlana Kiritchenko, Joel Martin, Berry de Bruijn

Institute for Information Technology, National Research Council Canada, 1200 Montreal Road, Ottawa, ON, Canada K1A 0R6

ARTICLE INFO

Article history:
Received 26 April 2012
Accepted 17 November 2012
Available online 4 February 2013

Keywords:
Text mining
Natural language processing
Electronic Health Records
Artificial Intelligence
Algorithms

ABSTRACT

This paper addresses an information-extraction problem that aims to identify semantic relations among medical concepts (*problems*, *tests*, and *treatments*) in clinical text. The objectives of the paper are twofold. First, we extend an earlier one-page description (appearing as a part of [5]) of a top-ranked model in the 2010 I2B2 NLP Challenge to a necessary level of details, with the belief that feature design is the most crucial factor to the success of our system and hence deserves a more detailed discussion. We present a precise quantification of the contributions of a wide variety of knowledge sources. In addition, we show the end-to-end results obtained on the noisy output of a top-ranked concept detector, which could help construct a more complete view of the state of the art in the real-world scenario. As the second major objective, we reformulate our models into a composite-kernel framework and present the best result, according to our knowledge, on the same dataset.

1. Introduction

The increasing availability of digitalized medical texts, e.g., those having already been converted to and encoded with ASCII or Unicode,[1] has actually opened a promising way to collect real-life, real-time, and large-sample-based knowledge from real patients, in contrast or in complement to knowledge that is obtained in laboratories, with more controlled experiments, or based on a relatively smaller number of samples. While the order of magnitude of textual data continues to make manual analysis less affordable, computers' ability in understanding human languages has improved in the past decades, particularly through the use of data-driven approaches.

This paper addresses a core medical *information extraction* problem–identifying semantic relations among medical concepts in discharge summaries and progress reports, i.e., relations existing between medical *problems*, *tests*, and *treatments*. The problem has been formally defined for the 2010 i2b2/VA Challenge [18,19] and will be presented in Section 2.

The objectives of this paper are twofold. We first extend an earlier one-page description (as a part of [5]) of a top-ranked model in the i2b2-2010 Challenge to a necessary level of details, with the criterion that an interested reader should be able to reimplement all our models–we believe that feature design is the most crucial factor to the success of our system and hence deserves a more detailed discussion. We also precisely quantify the contributions of a wide variety of knowledge sources with very different nature. More exactly, the features we have carefully explored for the task range from superficial word/concept statistics to explicit/implicit domain semantics, shallow and deep syntactic features, and knowledge learned from unlabelled data. In addition, we introduce the end-to-end result obtained on the noisy output of a top-ranked concept detector: we hope this would help construct a more complete view of the state of the art in the real-world scenario that was not evaluated in i2b2-2010 itself.

As the second objective, we reformulate our models into a composite-kernel framework, which results in the best result, according to our knowledge, on the i2b2-2010 dataset. Unlike an open-domain task that often addresses newswire data, the domain-specific problem here involves abundant domain semantics, including not only manually created resources (e.g., UMLS) but also automatically extracted knowledge (e.g., from both MEDLINE and unlabelled data). Our results allow us to conclude that complex syntactic structures can further improve the modeling quality for the semantic task, even when abundant domain-specific semantics has already been carefully explored. However, we also observed that not all syntactic kernels effective in the open domain are useful in our task here.

2. Problem

The problem that we are concerned with here is to identify semantic relations between medical concepts in plain clinical

* Corresponding author.
E-mail addresses: Xiaodan.Zhu@nrc-cnrc.gc.ca (X. Zhu), Colin.Cherry@nrc-cnrc.gc.ca (C. Cherry), Svetlana.Kiritchenko@nrc-cnrc.gc.ca (S. Kiritchenko), Joel.Martin@nrc-cnrc.gc.ca (J. Martin), Berry.DeBruijn@nrc-cnrc.gc.ca (B. de Bruijn).

[1] In a more general viewpoint, one should consider the transcripts of spoken content, e.g., those of doctors' voice recordings, to be a special case of such textual data, where human medical transcription (MT) and editing (MTE) have already formed their own market, and automatic speech recognition (ASR) has started to play a more important role.

1532-0464/$ - see front matter
http://dx.doi.org/10.1016/j.jbi.2012.11.006

texts. To begin with an example, a sentence in a patient's discharge summary reads:

> ...*He was a poor candidate for anticoagulation because of his history of metastatic Melanoma*....

The above sentence mentions a relation between a treatment (*anticoagulation*) and a problem (*metastatic Melanoma*); that is, the treatment is not appropriate for the medical problem. Our work here follows the definition of the 2010 i2b2/VA NLP Challenge, which aims to recognize three types of relations: *treatment-problem*, *test-problem*, and *problem-problem* relations. The definitions of relations, with examples, are shown in Table 1.

The definitions of these categories of relations are rather self-explanatory, while details can be further found in [18,19]. We note that although the task is defined as such, the methodology we discuss in this paper could be applied to a broader type of clinical semantic relations; e.g., our model has been extended to detect temporal relations defined in the 2012 i2b2 Challenge and achieved also a top-ranked performance there. As shown in Table 1, there are three general types of relations that contain 6, 3, and 2 specific categories, respectively, including the negative categories indicating no relations between two concepts. For clarity, in the remainder of the paper, if not otherwise noted, a *type* of relation refers to one of the three general classes of relations, while a *category* refers to a specific relation in each *type*; e.g., we say *treatment-problem* is a *type* of relation, while *TrIP* is a *category* of relation.

The overall task is to classify the relation between a pair of medical concepts into one of the relation categories defined in the table. Following the definition[2] of the 2010 i2b2/VA Challenge task, this paper only concerns with the relations defined above and that appear in the same sentence. Sparse relations between two concepts from different sentences are not discussed here. Note that Table 1 provides only simple examples in short sentences, while the real relations we have to deal with could be more complicated. For the i2b2/VA-2010, gold concepts were given to the participants of the relation-detection task. In this paper, we also report the performance of a full pipeline, where a state-of-the-art concept detector is applied first to find medical concepts automatically.

Table 1
Definitions of medical-concept relations by i2b2. Concepts in the examples are in brackets, with *pr*, *tr*, and *te* representing *problem*, *treatment*, and *test*, respectively.

Type 1: treatment-problem relations	
TrIP	Treatment improves problem *[hypertension]/pr was controlled on [hydrochlorothiazide]/tr*
TrWP	Treatment worsens problem *He was discharged to home to be followed for [her coronary artery disease]/pr following [two failed bypass graft procedure]/tr*
TrCP	Treatment causes problem *[Hypothyroidism]/pr following near total [thyroidectomy]/tr*
TrAP	Treatment administered for problem *[antibiotic therapy]/tr for presumed [right forearm phlebitis]/pr*
TrNAP	Treatment is not administered because of medical problem *He was a poor candidate for [anticoagulation]/tr because of his history of [metastatic Melanoma]/pr.*
NTrP	No relation between a treatment and a problem
Type 2: test-problem relations	
TeRP	Test reveals problem *patient noted to have [acute or chronic Hepatitis]/pr by [chemistries]/te*
TeCP	Test conducted to investigate problem *[chest xray]/te done to rule out [pneumonia]/pr*
NTeP	No relation between a test and a problem
Type 3: problem-problem relations	
PIP	Medical problem indicates medical problem *a history of [noninsulin dependent diabetes mellitus]/pr, now presenting with [acute blurry vision on the left side]/pr.*
NPP	No relation between two medical problems

3. Methods

3.1. Concept annotation

Our relation-detection task takes as input the clinical documents with medical concepts annotated. In the i2b2/VA-2010 Challenge set-up, medical concepts were manually annotated by human judges. This setup would help us understand the upper-bound performance of relation detection without considering noise in concept detection. To observe the performance under a more realistic scenario, where concepts are detected automatically, we annotated concepts with a top-ranked concept recognizer [5]. In addition, we also used the same recognizer to leverage unlabeled data in order to further improve the relation-detection performance, as discussed later. So, we briefly introduce this concept recognition system here.

Detecting medical concepts can be generally treated as a named entity recognition (NER) problem, similar to that defined and evaluated early in Message Understanding Conference (MUC) [9] and more recently in Automatic Content Extraction (ACE) [8]. While open-domain NER that identifies persons, organizations, and locations from news articles can often achieve a high performance, sometimes comparable to human performance, NER in the clinical domain is still an open problem, as was revealed in the i2b2/VA-2010 Challenge itself.

The automatic concept recognition system used in this paper is a discriminative semi-Markov model [5], trained with passive-aggressive online updates. This model allows for conducting the task without requiring a Begin/Inside/Outside (BIO) tagging formalism, and provides at least two major advantages. First, by labeling multi-token spans, labels cohere naturally, which allows the tagger to perform well without tracking the transitions between labels. Second, semi-Markov models allow for more flexibility in feature construction as one has access to the entire text span of a concept; for example, it is ready to include features like *concept lengths*. At the same time, the model was also designed to consider features that are unique to BIO, e.g., those pertaining to the beginning of a concept, by creating copies of all word-level features that indicate if the word begins or ends a concept. The model was trained using an online algorithm called the Passive-Aggressive (PA) algorithm [4] with a 0–1 loss. This concept recognition system achieves a 0.852 *F*-measure on the test set of the i2b2 concept detection task.

3.2. Relation detection

One important goal of information extraction is to reveal the relations between concepts, which is the problem we address in this paper. Relation detection is a typical multi-class categorization task: a relation between two concepts has to be classified into one of the categories defined above in Table 1.

In the development phase, we have explored different classifiers in our development phase, such as maximum entropy (ME), support vector machine (SVM) with different kernels, logistic regression, and *k*-nearest neighbor (kNN), but did not observe significant difference in performance according to our cross validation conducted on the training data. This suggests that we should focus our attention more on the knowledge used in this decision-making process, than on the classification algorithms themselves. In the

[2] Note that the actual realization of the definition is through data annotation by human judges, who may disagree on some cases, while i2b2 has not made available such statistics yet.

remainder of this paper, the results presented were acquired by using ME, which is relatively less memory demanding (e.g., compared with the memory-based kNN) and less computationally expensive (e.g., compared with SVM). This allowed us to explore the performance of a wide variety of features of different nature, e.g., with cross validation, when preparing the i2b2 competition itself in the given time, it also facilitated our further analysis of the problem. For example, we were able to train approximately 900 models for our regression analysis, as discussed later in Section 7.3. The whole task is evaluated with an asymmetric metric, i.e., the micro-averaged *F*-measure, in which a false-positive error and a false-negative error can have a different effect. We will detail the evaluation measure in the experiment set-up session later.

A maximum entropy model (ME) follows the principle of *maximum-entropy estimation* to infer the unknown parameters of a discriminative model. The basic idea is that while a model satisfies all given constraints imposed by the training data, it maximizes the (conditional) entropy defined over the training data and the labels, i.e., preferring a uniform distribution as much as possible when satisfying the constraints. As it has been shown, an ME model always conforms to an exponential form:

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

$$Z(x) = \sum_x \exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

For our task here, x stands for a concept pair and its context in the sentence, y is the corresponding relation label, and $f_i(x,y)$ is a feature function with λ_i being a model parameter that needs to be estimated to weight the contribution of the feature. $Z(x)$ is a normalizing coefficient to ensure a proper probabilistic distribution. During testing, an assignment of y is found to maximize $p(y|x)$ above, while during training, given a set of training data that introduce constraints, the model parameters are adjusted to maximize the likelihood of generating these training data, typically tuned with a greedy algorithm called generalized iterative scaling (GIS). A good introduction to ME in natural language processing (NLP) settings can be further found in [7]. Specifically in our experiments below, we utilize the ME package of OpenNLP [16]. We train three different ME classifiers, one for each type of categories defined in Table 1.

With the limited amount of labeled data, we also situate the relation-detection task in a semi-supervised setting. The efforts are twofold. First, we apply a bootstrapping process to unlabeled data that are expected to obey the same distribution as the provided training data: these data are clinical texts of the same kinds from the same healthcare institutes as the annotated training data are (Section 4.4). Second, as we will explain later, the bootstrapping is used together with a down-sampling process in order to balance positive/negative relation categories (Section 6.1.1).

4. Knowledge sources and features

4.1. Word/concept statistics

We exerted intensive efforts in exploring the usefulness of various superficial word/phrase/concept features, expecting that a careful exploration of such basic features is not only of great importance for improving the system's performance for the competition, but would also help us more accurately assess the usefulness of extra, more advanced knowledge such as the syntactic structures, additional domain semantics, and knowledge embedded in unlabelled data that we will further discuss.

We first borrowed the features used by a successful system [13] on a task of extracting medication events, by taking the following word/concept statistics into account,[3] which we refer to as *basic* word/concept statistics.

- Three words before and after each of the two concepts.
- All words between the two concepts.
- Words contained in the two concepts.
- Concepts appearing between the two concepts.

In our implementation, we made these statistics order-sensitive when applicable, motivated by the consideration that the distribution of features could be different under these different circumstances. For example, an order-sensitive feature used to classify *treatment-problem* relations looks like:

$$f_i(x_i, y_i) = \begin{cases} 1 & \text{word "}\textit{with}\text{" appears between a problem and treatment\& the problem is before the treatment } \&y_i = \text{"}\textit{TrAP}\text{"} \\ 0 & \text{otherwise} \end{cases}$$

In this example, we can see that this feature takes the value of 1 only if the problem under concern appears before (to the left of) the treatment. In general, we enforced a wider range of word/concept features and constraints. For example, we included 7-bit hierarchical word clusters calculated on the unlabeled data with Brown's clustering algorithm [2]. The algorithm clusters words (i.e., word types) hierarchically into a binary tree, with each word expressed with a leaf. Each non-leaf node merges semantically similar words or a sub-cluster of words. Since the two edges connecting a non-leaf node and its children are given a label of 0 and 1, respectively, each leaf (word) is associated with a unique bit string representing the path from the root to it, which encodes the semantic-category information and is used as a feature in this work. Specifically, we take the leftmost 7 bits for a word to represent the cluster it belongs to. A well-known application of this algorithm in NLP, specifically in information extraction, is discussed in [12].

We also included so-called *rigid features*, which means that any of their occurrences invalidate the use of all other regular, non-rigid features. The intuition behind this approach is to incorporate into our statistic models some strict rules. For example, if the following feature appears, we are sure that the two medical problems under concern have no relation associated: "only conjunction words or phrases appear in between two medical problems". Without enforcing such features rigidly (i.e., just considering them as regular features), we could not eliminate obvious mistakes made on these cases in our held-out dataset. We also included n-gram features and features related to punctuations, e.g., "stronger separators such as semicolons appear in between the two concepts under concern". In addition, we found some carefully, well designed features to be particularly useful. For example, we already have features like " number of concepts in a sentence"; however, features such as "the number of concepts in the sentence is exactly two (i.e., the two concepts whose relation is being classified)" were found to give additional improvement. Intuitively, if a sentence contains only two concepts exactly, these two concepts are more likely to have some positive relationship, which in turn needs to be reflected in the feature design. We also added sentence boundaries ⟨S⟩ and ⟨/S⟩ at the beginning and the end of each sentence to reflect if the words or concepts are close to the sentence boundaries. In the evaluation section, we refer to these augmented word/concept features as "rich word/concept features".

[3] Our baseline model here only uses the minimal features among several possible explanations of the features used in [13].

4.2. Domain semantics

Intuitively, domain knowledge is expected to be one of the major keys to finding the correct relations. Actually such knowledge has already been implicitly captured by the classifier trained on the provided training data with the word/concept statistics described above. For example, in the sentence "*the patient had a non-ST elevation MI with evidence of a high percent mid LAD lesion*", the domain knowledge–the problem "*mid LAD lesion*" often indicates another problem "*elevation MI*" – is likely to be learned directly from the training data if such examples appear frequently enough. In another example, "*...nitroglycerin 0.3 mg sublingually p.r.n. chest pain or shortness of breath...*", the role of the abbreviation *p.r.n.* in predicting medical relations is also learnable, again, if it is frequent enough, even when the system does not necessarily know what *p.r.n.* really stands for (it refers to *pro re nata*, meaning *as the circumstance arises*).

The limited availability of labeled data, however, often results in sparseness and restricts the domain knowledge that can be directly acquired from training cases. To have more a comprehensive understanding of the role of domain semantics, we explore the effectiveness of (1) manually created, explicit domain knowledge, and (2) automatically acquired domain semantics from a large volume of domain texts. Those additional texts do not necessarily obey the same feature distribution as the training set does, but still contain relevant domain knowledge.

4.2.1. Manually-authored domain semantics

We explored two different types of manually-authored domain semantics. The first is generic medical knowledge bases that were manually created for a general purpose of automatic healthcare text processing. Namely, we used the well-known UMLS/MetaMap meta-thesaurus along with the MetaMap entity recognition tool. In addition, we manually built word/phrase clusters specific to clinical summaries and progress reports, in order to provide some degree of smoothing on these lexicalized features.

4.2.1.1. UMLS/MetaMap. The first explicit, manually created knowledge base we incorporate is the Unified Medical Language System (UMLS) [17], created and maintained by the U.S. National Library of Medicine (NLM) to "facilitate the development of computer systems that behave as if they 'understand' the meaning of the language of biomedicine and health." This knowledge base contains a unified thesaurus and ontology, the mapping between different terminology systems and disparate databases, as well as the corresponding software tools that perform on these data. The UMLS Meta-thesaurus covers over 1 million biomedical concepts and 5 million concept names, and was created from more than 100 different vocabulary sources with human intervention of editing and reviewing.

Specifically in this study, we applied MetaMap [1], which is a widely-used entity recognition tool in the biomedical domain. We used MetaMap to recognize lexical variations of medical concepts from UMLS within their context. With the MetaMap matching results, we can represent words by their domain-specific semantic categories, i.e., UMLS semantic types such as "sign or symptom" and "therapeutic or preventive procedure". These labels are used as features to hopefully smooth the sparseness of lexicalized features. The semantic-type labels are associated with words in our systems: when MetaMap assigns a label to a multi-word phrase, we break the phrase into words and assign the same label to each word to acquire flexibility in feature construction. More specifically, we use the unigram UMLS labels of the three words before and after the two concepts in question, of the words between them and of the words contained in them. In addition, we use UMLS label pairs associated with each word pair from the two concepts, i.e., one label from each concept.

4.2.1.2. Domain word/phrase clusters. We have also manually created word/phrase clusters specifically for clinical text to further smooth data sparseness. For example, we created a list to include words, phrases, and doctors' shorthands that express *indication* such as "p/w", "have to do with", " secondary to", " assoc w/". Another example is a *resistance* list containing words/phrases such as "unresponsive", "turn down", and "hold off". We extracted these features in a way similar to that described in Section 4.1. That is, we identify if words in a domain-word list appear within three words before or after the two concepts in question, and extract these as binary features. Similarly, we check the occurrences of these domain word/phrase lists among words between the two concepts and words in the concepts. An example of such features is: "a *resistance* word appears within three words after a *problem*."

4.2.2. Automatically-acquired domain knowledge

In addition to the explicit domain semantics that are created manually, there is abundant domain knowledge embedded in much larger free-text. We are also curious about its usefulness in this task. MEDLINE, for example, is a bibliographic database of life sciences and biomedical information. It includes 5000 selected resources and covers such publications from 1950s to the present, including health-related fields such as medicine, nursing, pharmacy, dentistry, veterinary medicine, preclinical sciences, and healthcare. The database contains more than 18 million records approximately and has been widely used in various healthcare-related research. In this work, we calculate the pointwise mutual information (PMI) between the two given concepts in all the abstracts of MEDLINE articles to estimate their relatedness. The motivation is that such information could provide evidence to help determine the likelihood that the two concepts have a positive relation, though not necessarily their specific relation categories.

4.3. Syntax

4.3.1. Dependency structures

We investigate if the syntactic relationship between two medical concepts in a parsing tree provides useful information to help discriminate their semantic relations defined in Table 1, and if so, how effective it is. As an example, Fig. 1 shows an automatically generated dependency parsing tree for the given sentence.

Intuitively, the word "*revealed*", which connects the problem "*partial decompression of the spinal canal*" and the test "a *postoperative CT scan*", seems to be very indicative in predicting their relation. It also seems that even the word distance between these two concepts in the tree (i.e., the number of words on the path that connects them) is more indicative than its counterpart in the word/phrase features discussed earlier, e.g., "number of words between two concepts in the (sequential) sentence". We can see the former (distance in the tree) can effectively skip the noun

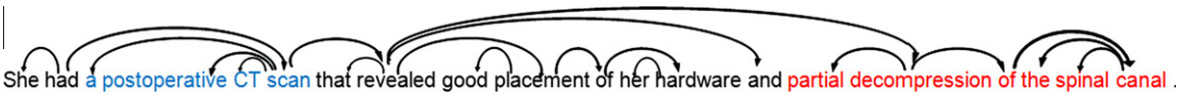

Fig. 1. A dependency parsing tree of an example sentence.

phrase "*good placement of her hardware*", while the latter (distance in the sequence) is likely to suffer from noise so introduced, though we observed in our development phase that it is still a useful feature.

To acquire the empirical evidence of the usefulness of sentential syntax, we parsed the input text using the Charniak's ME reranking parser [3] with its improved, self-trained biomedical parsing model [11]. These were then converted into Stanford dependencies [6].[4] The features we extracted from the dependency parsing trees included words, their dependency tags, and arc labels on the dependency path between the two minimal trees that cover each of the two concepts, respectively, along with the word type and tags of their common ancestor, as well as the minimal, average and maximal tree distances between these two minimum-covering trees and their common ancestor.

4.3.2. cTAKES

Based on IBM's Unstructured Information Management Architecture, cTAKES (clinical Text Analysis and Knowledge Extraction System) [15] provides a pipeline that conducts a series of language processing on free-text clinical notes, e.g., tokenization, spelling checking, POS (part-of-speech) tagging, shallow parsing, negation annotating, and word sense disambiguation. In our work, we employed the POS tags of the cTAKES pipeline to capture words' different roles of grammatical categories. For example, a verb appearing between a treatment and a problem, particularly those in a past tense, could be more likely to indicate an existence of relation, than a present tense. More exactly, we use unigram POS tags of words appearing between the two concepts and those of the tree words before and after each concept.

4.4. Unlabeled data

With the often limited availability of labeled data, we also hope to further understand the usefulness of unlabelled clinical texts that are expected to obey the same feature distribution as the training data. For this purpose, we leveraged 827 extra raw discharge summaries or progress reports provided by i2b2. These documents are from the same healthcare centers as the training data, but not manually annotated with either concepts nor relations. We first applied a top-ranked concept-recognition model (described in Section 3.1) to tag the three types of medical concepts, i.e., *problems*, *treatments*, and *tests*. Then we applied our best relation-detection model trained on manually labeled training data to annotate relations between these automatically recognized concepts on the unlabelled data, from which we extracted all the features we have discussed above to retrain our model and repeated this bootstrapping process multiple times. More exactly, based on the improvement achieved on micro *F*-measure scores, the process was conducted twice. For each epoch, filtering was applied to balance the categories with the method described in Section 6.1.1, using a down-sampling ratio decided by the improvement achieved during development. In fact, system voting could be an additional choice here but we did not adopt it in the current implementation.

4.5. Other models in i2b2-2010

In this section, we briefly discuss several other models that achieve good results in i2b2-2010. The model proposed in [23] identifies relations in two steps: (1) finding concept pairs that have positive relations; (2) classifying these pairs into different relation categories. These two phases use features similar to our word/concept features discussed in Section 4.1 with an SVM classifier. The model also includes several other interesting types of features. First, Wikipedia is used as a knowledge source, where the links and hierarchies among Wikipedia articles are used to estimate the relatedness of two concepts, if the concepts can be mapped to some Wikipedia articles. Another interesting feature set is the inexact matching features, which calculate an edit distance for the contextual strings between two relations. The models also use some simple syntactic features such as the predicates associated with each concept, but not the full parse trees as we use in this paper. The best model in [23] achieves an *F*-measure statistically tied with that of our model, and has a higher recall and lower precision than our model, which could be due to its use of Wikipedia and the inexact matching: both could increase the recall. These two models, ours and that in [23], statistically outperform the other 14 models submitted to the i2b2-2010 Challenge. The approach proposed in [24] combines a rule-based model with a supervised classifier, forming an interesting model. The significantly better performance of the top two models suggests the importance of feature design and the usefulness of rich features extracted from a wide range of sources.

5. Composite kernels

With the above word/concept, syntactic, and semantic features, we have trained a maximum-entropy classifier and achieved a top-ranked performance in the i2b2-2010 evaluation, statistically tying with another system [23]. In this section, we reformulate our models into a composite-kernel framework, which has achieved encouraging results in open-domain tasks [20]. We show that the performance of our composite-kernel-based model is significantly better than that of our previous top-ranked model, and it is also the best result reported, according to our knowledge, on the i2b2-2010 relation dataset. As we have mentioned earlier, unlike a open-domain task (often using newswire articles), our domain-specific task here has abundant domain-specific semantic information. Our results allow us to conclude that complex syntactic information can further improve the modeling quality for the semantic task, even when abundant domain semantics has already been carefully leveraged.

Our composite-kernel-based framework consists of two components: the so-called *wrapping kernels* and the *convolution kernel*. We use the first component to wrap up our old models in order to take all their advantages, which will be then combined with the second component, a convolution tree kernel that is employed to explore an implicit, high-dimensional syntactic space.

5.1. Kernels

In both machine learning and natural language processing, kernel based methods have been widely studied and applied. In general, kernel methods are a way to extend a low dimensional feature space to a high dimensional one, with inexpensive computation (i.e., the *kernel trick*). More exactly, based on the fact that many machine learning algorithms, e.g., the *k*-nearest neighbor and perceptron, involve only a dot product between two feature vectors, simply replacing a dot product with a kernel function will map the original feature space to a high dimensional one. As such, a linearly non-separable problem could often become more separable. Mathematically, as long as a function is symmetric and the resulting kernel matrix is positive semi-definite, the function is a valid kernel function. Typical kernels include linear, polynomial, and radial basis functions, among others.

[4] We observed no improvement when extracting features directly from the phrase-structure parsing trees, although the features we extracted from dependency parsing trees should have also existed in the corresponding constituency parsing trees. This could be due to the advantages of dependency structures, as discussed in [10], among many others.

Among many properties of kernel functions, an important one for our problem here is that the sum of given kernels is still a valid kernel. With this combinational property, we can use a composite-kernel-based framework to combine our previous best model with a convolution tree kernel in order to explore complex syntactic structures, as suggested first in [20].

5.2. Wrapping kernels

To take all the advantages of our previous best model, we use two types of kernels to incorporate all the features discussed in Section 4, and we call them wrapping kernels.

5.2.1. Concept kernels

The first type of wrapping kernel is *concept kernel* K_c, which incorporates features that can be associated with a medical concept and takes the following form:

$$K_c(R_1,R_2)=\sum_{i=1,2}K_c(R_1\cdot C_i,R_2\cdot C_i)=\sum_{i=1,2}\sum_{k}I(R_1\cdot C_i\cdot f_k,R_2\cdot C_i\cdot f_k)$$

In the formula, R_1 and R_2 are two relation instances, each of them involving two concepts; for example, $R_1\cdot C_1$ and $R_1\cdot C_2$ refer to the two concepts in R_1, while f_k is the kth feature of the corresponding concept. $I(x,y)$ is an indicator function taking the value 1 if $x=y$ and 0 otherwise, and it will be replaced by a kernel function in our experiments, where the form of the function is determined by its performance on a held-out set. Again, all features that can be associated with a concept are incorporated here. For example, among concept/word features described in Section 4.1, "three words before and after each of the two concepts" would be incorporated into the concept kernels. In semantic features, we incorporate the UMLS/MetaMap features and the domain word/phrase cluster features, among others.

5.2.2. Connection kernels

The connection kernel K_n is used to represent the sequences connecting the two concepts in a relation and it takes the following form:

$$K_n(R_1,R_2)=\sum_{i}K_n(R_1\cdot S_i,R_2\cdot S_i)=\sum_{i}\sum_{k\in S_i}I(R_1\cdot S_i\cdot f_k,\ R_2\cdot S_i\cdot f_k)$$

In the formula, $R_1\cdot S_i$ refers to a sequence connecting the two concepts in R_1. Note that *sequence* here is a general term: it refers to any forms of connections between the two concepts. For example, it can be a word sequence between the two concepts or a path in a dependency parse tree that connects the two concepts. Accordingly, $R_1\cdot S_i\cdot f_k$ is the kth feature of the sequence $R_1\cdot S_i$.

5.3. Convolution tree kernels

The relations between two medical concepts could involve more complex syntactic structures, although we have incorporated some explicit syntactic features in Section 4.3. As pointed out in [22], many NLP tasks, involving "a parse tree that tracks all subtrees", have an input domain that "cannot be neatly formulated as a subset of $\mathbb{R}^d$"; i.e., expressing such features in the original d-dimensional vector space is not straightforward and therefore such features cannot be included into the wrapping kernels in a straightforward way. As an example, given two relation instances, R_1 and R_2, and the two sentences they occur in, we first find two minimal trees that cover the two concepts in R_1 and those in R_2, respectively. Our aim is to estimate the similarity (a dot product) between these two minimal trees in the vector space formed by all their subtrees. Unfortunately, a naive algorithm that lists all possible subtrees is intractable as the vector size is exponential to the number of tree nodes.

A convolution tree kernel has therefore been proposed by [22] for such rich, high-dimensional representations. Specifically, to measure the dot product between two trees in the space formed by all their subtrees, the convolution tree kernel employs recursive computation to calculate the similarity in terms of sub-structures. More exactly, through some simple algebra, the dot product mentioned above can be calculated with the following formula:

$$K_t(T_1,T_2)=\sum_{n_1\in N_1}\sum_{n_2\in N_2}C(n_1,n_2)$$

$$\text{where, } C(n_1,n_2)=\sum_{i}I_i(n_1)I_i(n_2)$$

In the equation, the sets of nodes in tree T_1 and T_2 are N_1 and N_2, respectively. $C(n_1,\ n_2)$ is simply the count of common subtrees rooted at node n_1 in T_1 and n_2 in T_2, where $I_i(n_1)$ is a binary indictor function which takes the value 1 if and only if the subtree T_i is rooted at node n_1. As such, $C(n_1,n_2)$ can be calculated recursively in polynomial time $O(|N_1||N_2|)$ with the following steps:

(1) $C(n_1,n_2)=0$ if the context-free rule production at node n_1 is different from that at node n_2.
(2) $C(n_1,n_2)=1$ if the rule production at node n_1 is same as that at node n_2, and both nodes are pre-terminals (nodes directly above words in the surface string, e.g., POS tags).
(3) Otherwise, the follow formula is used:

$$C(n_1,n_2)=\prod_{j=1}^{nc(n_1)}(1+C(ch(n_1,j),ch(n_2,j))$$

where $nc(n_1)$ denotes the number of children of node n_1; $ch(n_1,j)$ and $ch(n_2,j)$ are the jth child of n_1 and n_2 respectively.

In our experiments, we used the following formula to integrate the tree types of kernels discussed above, which was found to be better than several other candidates. The parameters in the formula were determined with a held-out dataset ($\alpha=0.15$, $\beta=0.15$, $d=3$).

$$K(R_1,R_2)=\alpha\cdot(1+K_c(R_1,R_2))^d+\beta\cdot(1+K_n(R_1,R_2))^d+(1-\alpha-\beta)K_t(T_1,T_2)$$

6. Experiment setup

6.1. Data

The data used in our experiments are the relation data of the i2b2/VA-2010 Challenge, which are real-life discharge summaries and progress notes recorded at three healthcare and medical centers.[5] For privacy considerations, all records have been fully de-identified before any manual annotation and data distribution. We present the detailed statistics of the training and test data in Table 2. We also utilized an unlabeled data set (Section 4.4), which contains 827 documents (details not presented here) and is about 2.3 times as large as the training data.

Table 2 reveals the unbalanced distribution of data points; e.g., the ratio of data points between PIP and NPP is roughly at 6:1, which needs to be coped with accordingly, particularly in the situation where such an unbalance could be further amplified when the bootstrapping process discussed in Section 4.4 is applied-bootstrapping could bias towards larger categories with already-learned bias from the previous round.

[5] Discharge summaries are from three resources: Partners HealthCare, Beth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center, progress notes being from the University of Pittsburgh Medical Center.

Table 2
Statistics of the i2b2 training and test data.

	Training set	Evaluation set
Documents	349	477
Concepts		
Problems	11,968	18,500
Test	7369	12,899
Treatment	8500	13,560
Relations		
Treatment-problem	4319	6949
TrIP	107	198
TrWP	56	143
TrCP	296	444
TrAP	1423	2486
TrNAP	296	191
NTrP	2331	3487
Test-problem	3573	6072
TeRP	1734	3033
TeCP	303	588
NTeP	1536	2451
Problem-problem	8589	13,176
PIP	1239	1986
NPP	7350	11,190

6.1.1. Down sampling

To address the data imbalance problem discussed above, in all experiments presented below, we down-sampled the negative data points in problem-problem relations before training our models, to a positive/negative ratio between 1:2 and 1:4, the aim of which is to alleviate the classifiers' bias towards the larger negative categories. We performed the down-sampling process in each round of bootstrapping when it is applied, since using already-biased output to train a new model might amplify the unbalance, if not intervened, as discussed in Section 4.4. During our development phase, we found that the down-sampling improved the *F*-measure by about 0.3–0.5 points consistently in our 5-fold cross-validation tuning.

6.2. Evaluation metric

The evaluation metrics used in this paper, same as in the i2b2 Challenge, is micro-averaged *F*-measure, i.e., a harmonic average of the micro-precision and micro-recall that are calculated with the formulas below, where TP_i, FP_i, and FN_i are true positive, false positive, and false negative counts for the i^{th} category of relations, respectively. $|C|$ is the total number of positive categories. In the final evaluation, $|C|$ equals to 8, which considers all three types of positive categories together, as listed in Table 1 or 2.

$$P_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \qquad R_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)}$$

7. Results and discussion

We present in this section the experimental results to show how our model incorporates the various sources of knowledge of different nature, to achieve the state-of-the-art performance. We also present the performance achievable when the model encounters noisy input, a typical real situation in which medical concepts are automatically identified by a state-of-the-art concept recognizer. In addition, a non-linear regression analysis is also conducted to help understand the usefulness of more annotated data (if consistently labeled with the i2b2 training data) and the effectiveness of our current use of the provided unlabeled data.

7.1. Performance

Our best system that applies the ME model in a simple semi-supervised set-up (as discussed above in Section 3.2) to leverage all knowledge (as discussed in Section 4), in both labeled and unlabeled data, achieves an overall 0.731 micro-averaged *F*-measure [5], which ranks the second in the i2b2/VA-2010 Challenge. We note that this result has no statistically significant difference from that of the first-placed system, where both systems are statistically significantly better than all the rest 14 competing systems.[6]

To explore the effectiveness of different knowledge sources in this decision-making process, Fig. 2 illustrates the contributions of different feature sources on coarse category level, i.e., word/phrase statistics (*W*), domain semantics (*D*), and syntax (*S*), and their combinations. From the figure, we can first see that when considered individually, *W* is the best individual feature category, meaning that a baseline model built on superficial lexical-level features can have already achieved a very strong performance.

Even with this strong baseline, extra sentential syntax and domain semantics can still significantly improve performance on the test set, as shown in Fig. 2, from the *F*-measure of 0.715 (*W*) to 0.720 (*SW*) and 0.721 (*DW*), respectively. The domain semantic knowledge complements all other features very well: although individually the feature set *D* (0.485) is much less effective than syntax *S* (0.661), it improves the performance of word/concept features (*W*) more that the syntax does. This actually confirms the benefit of efforts on constructing those domain resources such as UMLS. We will present more details on this below. In total, integrating all knowledge (*DSW*) together pushes the best performance to 0.727, where removing any of them would result in a decrease of performance significantly; e.g., removing *S*, *D*, or *W* from *DSW* reduces the performance from 0.727 to 0.721, 0.720, or 0.670 respectively. This, again, indicates the complementary property of these sources of knowledge for the task. We note also that, as discussed earlier, the syntactic features used here were automatically extracted from the dependency trees generated by an automatic parser, meaning that the achieved improvement has already considered parsing errors (McClosky et al. [11] reported ~84% *F*-measure), though we have no evaluation data here to measure the parsing performance separately.

Table 4 provides details of feature effectiveness by subcategories, corresponding to the discussion in Section 4. We can first see that intensively exploring word/concept statistics (i.e., the *rich* subcategories) is beneficial, it can clearly improve the performance of the *Basic* performance from 0.700 to 0.715. Note that all features in this category are relatively computationally inexpensive (e.g., comparing with the dependency features in the *Syntax* category), they bear great importance in real system construction, while adding more advanced features can further statistically significantly improve the performance. In the domain semantics (D) category, manually-constructed domain knowledge is more effective than that from the *automatic PMI*, where the major contribution is from UMLS, confirming the value of putting effort on constructing such human-authored knowledge. We can also observe the unbalanced precision and recall of the *Word/Phrase Clusters* features, suggesting the coverage problem of such kind of smoothing features. We expect details on such a level would not only help understand effectively these knowledge sources, e.g., those from knowledge-based construction and automatically acquired, but also be helpful if the models discussed in this paper need to be constructed or compared with.

[6] In the i2b2-2010 competition, a participating team can submit up to three systems (the output of the systems), and the best performed one is selected as the final competing system to represent that team.

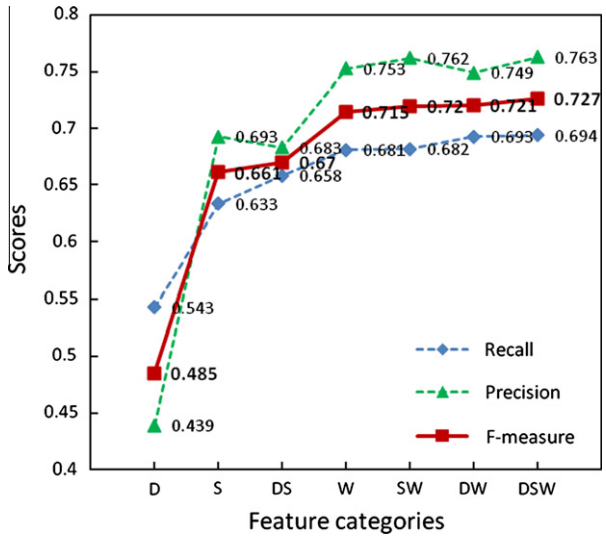

Fig. 2. Contributions of different types of knowledge: word/phrase statistics (*W*), domain semantics (*D*), syntax (*S*), and their combinations.

Table 3
Micro-averaged recall, precision, and *F*-measure of our best model.

	R	*P*	*F*
Best model	.693	.773	.731

Table 4
Observation on feature effectiveness by subcategories.

	R	*P*	*F*
Word/concept statistics (W)	.680	.753	.715
Basic	.671	.731	.700
Rich	.578	.620	.598
Domain semantics (D)	.543	.439	.485
Manual	.530	.423	.470
UMLS/MetaMap	.434	.489	.460
Word/phrase clusters	.230	.622	.336
Automatic PMI	.370	.516	.431
Syntax (S)	.693	.661	.633
Dependency structures	.609	.621	.615
cTAKES	.586	.658	.620

The results in both Fig. 2 and Table 4 correspond to the supervised ME models trained on manually annotated data. As discussed in Section 4.4, we also applied the model in a semi-supervised setting to leverage unlabelled data, which further improved the best performance in Fig. 2, i.e., the 0.727 of *DSW*, to our best result presented in Table 3, i.e., 0.731, where an improvement of 0.5 absolute *F*-measure is observed. Note that although we extracted all types of features from the unlabelled data in the same way as from the training data (but with bootstrapping from the unlabelled data), due to the potential noise introduced by the automatic concept recognizer, we avoided including the knowledge learned from the unlabelled data in our analysis of feature effectiveness above.

To provide an additional intuitive view, Fig. 3 presents training and test data in a reduced-dimensionality space, where the *DSW* feature space is reduced to three dimensions with principal component analysis (PCA). Specifically, subfigures (1)–(3) represent test data for (1) treatment-problem relations, (2) test-problem relations, and (3) problem-problem relation, respectively. Subfigure (4) also demonstrates test-problem relations in the training set, showing a similar distribution to that in (2). In all subfigures, blue dots represent negative examples, where no relations exist between the two candidate concepts; red dots represent the largest positive relations, i.e., TrAP, TeRP, and PIP, respectively (see Table 2 for details), while nodes in other colors present the data points of other positive relations. We can roughly see that in each of these figures, the positive and negative categories are rather discernable from each other even in such a reduced space, in which treatment-problem and test-problem relations are more similar to each other in their distributions, while problem-problem relations are more different.

7.2. Exploring the problem in a more realistic setup

Following the i2b2 evaluation guideline, the results presented above are all based on ideal concepts: the medical concepts in the test set are all manually annotated. This set-up helps understand the ideal state-of-the-art relation-detection performance, without subjecting to noise from concept recognition. However, in a more realistic scenario, a natural question is then: how will the model perform without assuming the concepts are given, but automatically recognized by a real system? To further investigate this problem, we first applied an also top-ranked concept recognizer (see Section 3.1 for more details) to annotate the test set and then used our best model (that in Table 3) to identify relations. The results are presented in Table 5.

The first row is copied from Table 3 for comparison, showing the performance of our best model on idea input, while the second row contains the results of relation detection on the noisy test set with automatically recognized concepts. The performance drops dramatically from an *F*-measure of 0.731 to that of 0.534, where the corresponding *F*-measure of the concept recognition used in the latter is 0.852. Our further manual analysis attributes this significant drop mainly to the stringent evaluation metric used. All errors in concept recognition, even a small shift of a concept boundary from the corresponding gold-standard, where the concept label itself is correct, will result in errors in relation detection, without an exception. On the other hand, errors of mislabeling a concept, misrecognizing its boundaries, or errors on both should have different effects on human perception and also on other applications, let alone with the more subtle situation in which boundary errors themselves could vary in their distances (i.e., word numbers) from the gold boundaries. The current evaluation metric, however, treats all the same.

We believe this should receive more attention from the community for both pragmatic and theoretic consideration, particularly if considering that the impact could be much more prominent in healthcare-related domains, where concepts are often much longer than the general named entities (NE) in newswire data that have received the most intensive attention in the NLP research.

7.3. Effects and sufficiency of human labeling

In addition to these automatic approaches, we are also concerned here with the effects and sufficiency of human labeling efforts in improving the performance, by analyzing the effect of training data sizes on our current performance, the potential benefit of acquiring more labeled data, and the effectiveness of the semi-supervised methods utilizing the provided unlabelled data (Section 4.3). We examine these questions with the ideal i2b2 models (those in Section 6.1), avoiding the impact of the noise introduced from concept recognition.

The left part of Fig. 4 (the red squares) helps understand the first question above, i.e., the effects and sufficiency of human labeling efforts in improving the performance. These red squares are acquired by regarding the size of the official i2b2-provided training

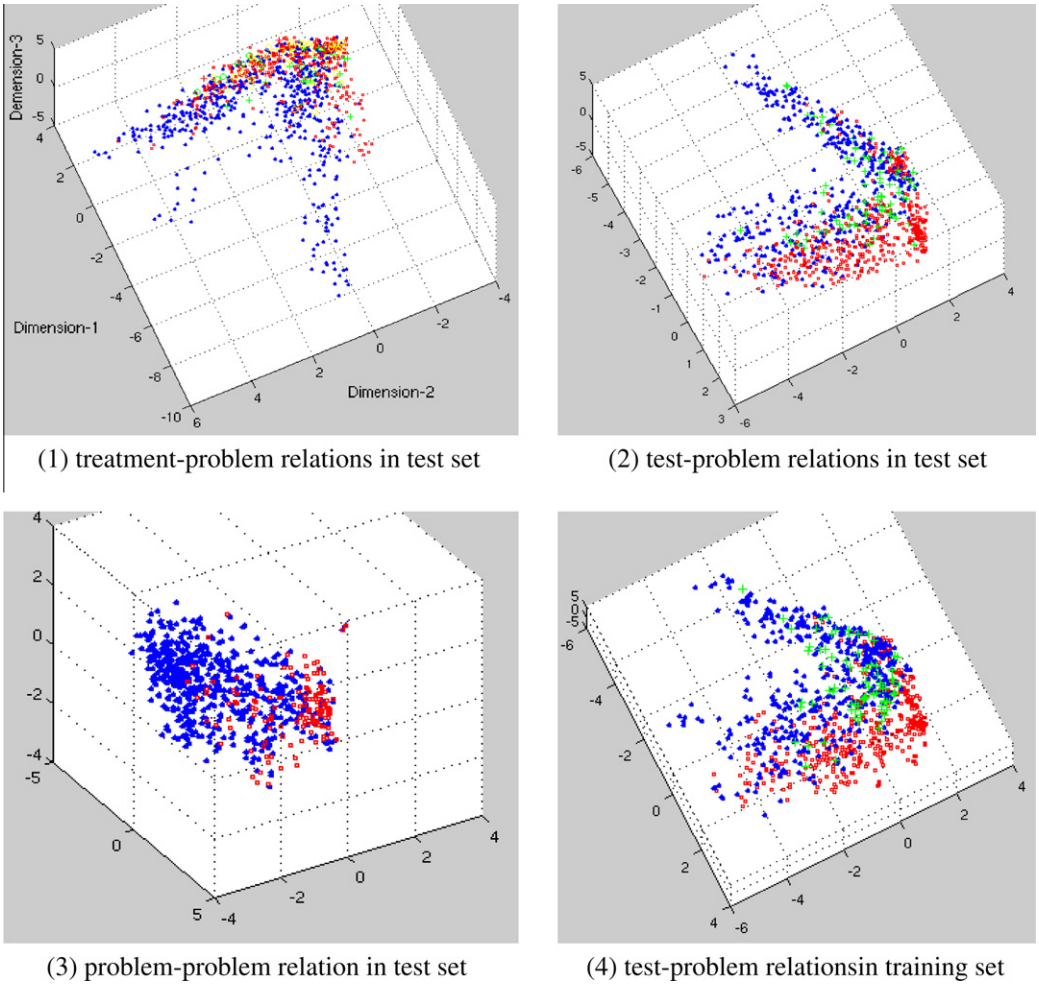

Fig. 3. Training and test data visualized in a dimensionality reduced space acquired with principal component analysis. Blue dots represent negative examples, where no relations exist between two candidate concepts; red dots represent the largest positive relations, i.e., TrAP, TeRP, and PIP, in each type of relation (see Table 2 for details), while green dots are data points of other positive relations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Performance of relation detection on the test set with concepts manually annotated and automatically recognized.

Test set	*R*	*P*	*F*
Concepts manually annotated	.693	.773	.731
Concepts automatically recognized	.488	.589	.534

data as the unit size 1, from which we randomly sampled subsets of different sizes, i.e., 0.1,0.2,. . .,0.9, and trained models with all *DSW* features discussed. Specifically, for each data size, we trained 100 models (in total 900 models) by sampling with replacement, and then calculated the average of the 100 micro-averaged *F*-measures on each size to get the corresponding red square in the figure. We can see that even when only half of the provided training data are used (without using the unlabelled data yet), our model achieves a 0.715 micro-averaged *F*-measure, already ranking at the 2nd place among the i2b2 submissions.

We applied a logarithm regression to fit the *F*-measures at these different training-data sizes (the ten red squares), computed with the Levenberg–Marquardt algorithm with the least-square-error criterion. The acquired blue curve suggests that exerting efforts on annotating more data would likely to further improve the performance, if the annotation is consistent with the training set. For example, if all the i2b2-provided unlabeled data (as discussed in Section 4.4) were annotated and added to the training set, the projected *F*-measure on the curve would be 0.751 (the green dot), which is 2.4 points higher than that trained with the current training set (0.727 at the unit size 1), or 2.0 points higher than the result achieved by utilizing these unlabeled data without human labeling, i.e., through applying a bootstrapping process discussed in Section 4.4, where a 0.731 *F*-measure is observed (the red diamond in the figure).

7.4. Performance of the composite-kernel-based model

Up to now, we have employed the maximum-entropy model as our classifier. In this section, we present the results of our experiments with the composite kernel methods (Section 5). For this purpose, we used SVMLight [25] and the tree kernel toolkit [26] and one-vs-all strategy for the multiclass classification problem. Table 6 shows the performance of the composite-kernel-based model, which achieves a micro *F*-measure score of 0.742, a performance statistically significantly better than that of our top-ranked model (0.731) with 95% confidence. Within the kernel framework, if we remove the convolution tree kernel but keep all others, the best result we observed was 0.733, showing a marginal (non-statistically significant) difference between the

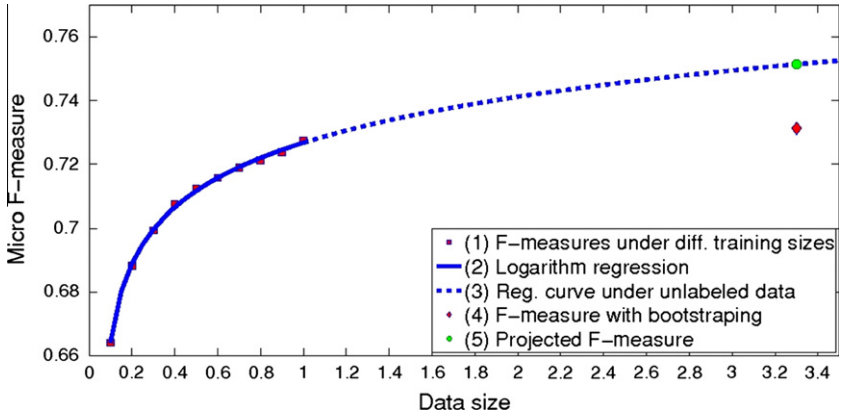

Fig. 4. A logarithm-regression analysis on model performances under different sizes of labeled and unlabeled data. The ten red squares are micro-averaged *F*-measures of the models trained on randomly sampled subsets of the original training data. The green dot is the projected *F*-measure, modeling the performance if the i2b2-provided unlabeled data were all manually annotated. The red diamond is the *F*-measure of our best model, which uses these provided unlabeled data through bootstrapping. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Performance of relation detection on the test set with composite kernels.

	R	*P*	*F*
Composite-kernel-based model	.726	.755	.742

wrapping kernels and the original maximal-entropy framework (0.731). This agrees with our earlier experiments conducted for the competition (Section 3.2), where we used several different classifiers and did not observe a significant difference in performance. The result clearly shows the effectiveness of the convolution tree kernel by significantly improving the *F*-measure to 0.742. This allows us to conclude that complex syntactic structures can further improve the modeling quality for this domain-specific semantic task, even when abundant domain semantics has already been carefully utilized. The *F*-measure of 0.742 is also the best score reported, according to our knowledge, on the i2b2-2010 relation task.

Among several choices, the subtrees we found to be the most effective in calculating convolution kernel are the *path-enclosed trees*, which outperform all other subtree types, which agrees with the observation for open-domain data [20]. We also incorporated context-sensitive constituent parse trees [21] but did not observe further improvement. This may indicate that the contextual semantics that we have extracted in Section 4 have already captured such information well enough, e.g., the surface, syntactic, and semantic features associated with the three words before the first (left) concept and the three after the second (right) concept for a given relation instance.

8. Conclusions and future work

This paper addresses the problem of identifying semantic relations mentioned between two medical concepts in real clinical texts. We introduce a machine-learning model that achieves a top-ranked performance in an international competition. We first explore experimental evidences to help construct a comprehensive understanding of the roles of a wide variety of knowledge sources for this task, given the fact that the difference in performance among state-of-the-art classifiers is less discernable. We show that explicit domain semantics acquired from manually authored knowledge bases (e.g., UMLS) together with that implicitly embedded in domain-specific texts (e.g., MEDLINE), provide complementary knowledge in improving the model performance, although this category of knowledge by itself appear to be less effective, e.g., when compared with syntactic features and superficial statistics directly learned from the training data. Deep syntactic knowledge, even when computed automatically and hence containing noise themselves, i.e., errors from a parser, can still render additional benefit to improve the models. We provide comprehensive introduction and analysis on these knowledge sources, which all together raises performance to a 0.731 micro-averaged *F*-measure.

When we situate the task in a more realistic setup in which concepts are recognized automatically by a concept detector, the performance of relation extraction drops dramatically, which we attribute to the stringent evaluation metric used. We believe this problem should receive more attentions from the community, for both pragmatic and theoretic concerns, considering that the impact could be much more prominent in the healthcare-related domains where concepts are often longer than the general named entities (NE) in newswire data. Also, our non-linear regression analysis suggests the potential benefit of the availability of more annotated data, while our current use of the provided unannotated clinical data in a semi-supervised fashion has already yielded a modest improvement.

Furthermore, we reformulate our models into a composite-kernel framework and achieve a *F*-measure of 0.742, a performance statistically significantly better than that of our previous top-ranked model (0.731). The score is also the best-ever result, according to our knowledge, on the same dataset. The results allow us to conclude that complex syntactic structures can further improve the modeling quality for this semantic task even when abundant domain semantics has already been carefully utilized.

As our immediate future work, we would like to further explore the joint inference problem of relation detection and concept recognition, as well as the associated evaluation problem discussed in Section 6.2. In addition to the evaluation problem mentioned above, the connection between concept recognition and relation detection could deserve a further study. Instead of decoupling them into two independent tasks, joint inference would be an interesting solution. Recent literature on open-domain news data has provided examples of efforts along this direction [10,14]. Unfortunately, all this work assumes the availability of named entity boundaries (but not the labels of entities)—i.e., the positions where entities appear are assumed to be known—to simplify the inference as a joint labeling problem. Recent success of Lagrangian relaxation based methods in many NLP problems, including their special form, dual decomposition, could provide another way to help view our problem here, e.g., to drop the assumption of boundaries and add the constraints between concept recognition and relation detection in a soft way.

Acknowledgments

De-identified clinical records used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY.

References

[1] Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc 2010;17:229–36.
[2] Brown PF, Della Pietra VJ, deSouza PV, Lai JC, Mercer RL. Class-based n-gram models of natural language. Comput Linguist 1992;18(4):467–79.
[3] Charniak E, Johnson M. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In: Proceedings of the 43rd annual meeting of the association for computational linguistics; 2005. p. 173–80.
[4] Crammer K, Dekel O, Keshet J, Shalev-Shwartz S, Singer Y. Online passive-aggressive algorithms. JMLR 2006;7(March):551–85.
[5] de Bruijn B, Cherry C, Kiritchenko S, Martin J, Zhu X. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. J Am Med Inform Assoc (JAMIA) 2011;18(5):557–62.
[6] de Marneffe M, MacCartney M, Manning C. Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC; 2006.
[7] Berger A, Pietra V, Pietra S. A maximum entropy approach to natural language processing. Comput Linguist 1996;22(1).
[8] Doddington G, Mitchell A, Przybocki M, Ramshaw L, Strassel S, Weischedel R. The automatic content extraction program – tasks, data, and evaluation. In: Proceedings of LREC; 2004. p. 837–40.
[9] Grishman R, Sundheim B. Message understanding conference-6: a brief history. In: Proceedings of international conference on computational linguistics; 1996. p. 466–71.
[10] Kate RJ, Mooney R. Joint entity and relation extraction using card-pyramid parsing. In: Proceedings of the fourteenth conference on computational natural, language learning; 2010. p. 203–12.
[11] McClosky D, Charniak E, Johnson M. Effective self-training for parsing. In: Proceedings of HLT-NAACL, Brooklyn, USA; 2006.
[12] Miller S, Guinness J, Zamanian A. Name tagging with word clusters and discriminative training. In: Proceedings of HLT-NAACL, Brooklyn, USA, 2006.
[13] Patrick J, Li M. A cascade approach to extracting medication events. In: Proceedings of Australasian language technology workshop; 2009.
[14] Roth D, Yih W. Global inference for entity and relation identification via a linear programming formulation. In: Getoor L, Taskar B, editors. Introduction to statistical relational learning. Cambridge: MIT Press; 2007.
[15] Savova GK, Kipper-Schuler K, Buntrock JD, Chute CG. UIMA-based clinical information extraction system. In: Proceedings of LREC: towards enhanced interoperability for large HLT systems; 2008.
[16] <sourceforge.net>. The Apache Software Foundation [updated 2010 September 23; cited 2012 January 20]. <http://opennlp.sourceforge.net/projects.html>.
[17] <nih.gov>. National Institutes of Health [updated 2011 August 29; cited 2012 January 20]. <http://www.nlm.nih.gov/research/umls/about_umls.html>.
[18] <i2b2.org>. Informatics for integrating biology & bedside [updated 2011 August 29; cited 2012 January 20]. <https://www.i2b2.org/NLP/Relations/>.
[19] Uzuner Ö, Solti I, Cadag E. Extracting medication information from clinical text. J Am Med Inform Assoc 2010;17:514–8. http://dx.doi.org/10.1136/jamia.2010.003947.
[20] Zhang M, Zhang J, Su J, Zhou G. A composite kernel to extract relations between entities with both flat and structured features. ACL; 2006.
[21] Zhou G, Zhang M, Ji DH, Zhu Q. Tree kernel-based relation extraction with context-sensitive structured parse tree information. EMNLP-CoNLL; 2007. p. 728–36.
[22] Collins M, Duffy N. Convolution kernels for natural language. NIPS 2001:625–32.
[23] Roberts K, Rink B, Harabagiu S. Extraction of medical concepts, assertions, and relations from discharge summaries for the fourth i2b2/VA shared task. In: Fourth i2b2/VA shared-task and workshop challenges in natural language processing for clinical data; 2010.
[24] Grouin C, Abacha A, Bernhard D, Cartoni B, Deléger L, Grau B, et al. CARAMBA: concept, assertion, and relation annotation using machine-learning based approaches. In: Fourth i2b2/VA shared-task and workshop challenges in natural language processing for clinical data; 2010.
[25] Joachims T. Text categorization with support vector machine: learning with many relevant features. ECML-1998.
[26] Moschitti A. A study on convolution kernels for shallow semantic parsing. ACL-2004.