Using Soft Constraints in Joint Inference for Clinical Concept Recognition

Prateek Jindal and Dan Roth
Dept. of Computer Science, UIUC
201 N. Goodwin Ave, Urbana, IL 61801, USA
{jindal2, danr}@illinois.edu

Abstract

This paper introduces IQPs (Integer Quadratic Programs) as a way to model joint inference for the task of concept recognition in clinical domain. IQPs make it possible to easily incorporate soft constraints in the optimization framework and still support exact global inference. We show that soft constraints give statistically significant performance improvements when compared to hard constraints.

1 Introduction

In this paper, we study the problem of concept recognition in the clinical domain. State-of-the-art approaches (de Bruijn et al., 2011; Patrick et al., 2011; Minard et al., 2011; Jiang et al., 2011; Xu et al., 2012; Roberts and Harabagiu, 2011) for concept recognition in clinical domain (Uzuner et al., 2011) use some sequence-prediction models like CRF (Lafferty et al., 2001), MEMM (McCallum et al., 2000) etc. These approaches are limited by the fact that they can model only local dependencies (most often, first-order models like linear chain CRFs are used to allow tractable inference).

Clinical narratives, unlike newswire data, provide a domain with significant knowledge that can be exploited systematically. Knowledge in this domain can be thought of as belonging to two categories: (1) Background Knowledge captured in medical ontologies like UMLS, MeSH and SNOMED CT and (2) Discourse Knowledge expressed in the fact that the narratives adhere to specific writing style. While the former can be used by generating more expressive knowledge-rich features, the latter is more interesting from our current perspective, since it provides global constraints on what output structures are likely and what are not. We exploit this structural knowledge in our global inference formulation.

Integer Linear Programming (ILP) based approaches have been used for global inference in many works (Roth and Yih, 2007; Punyakanok et al., 2004; Marciiniak and Strube, 2005; Bramsen et al., 2006; Barzilay and Lapata, 2006; Riedel and Clarke, 2006; Clarke and Lapata, 2008; Denis et al., 2007; Chang et al., 2011). However, in most of these works, researchers have focussed only on hard constraints while formulating the inference problem.

Formulating all the constraints as hard constraints is not always desirable because in many cases, constraints are not perfect. In this paper, we propose Integer Quadratic Programs (IQPs) as a way of formulating the inference problem. IQPs is a richer family of models than ILPs and it enables us to easily incorporate soft constraints into the inference procedure\(^1\). Our experimental results show that soft constraints indeed give much better performance than hard constraints.

2 Methodology

Task Description

Input consists of clinical reports in free-text (unstructured) format. The task is: (1) to identify the boundaries of medical concepts and (2) to assign types to such concepts. Each concept can have 3 possible types, namely (1) Test, (2) Treatment and (3) Problem. We would refer to these three types by \textsc{test}, \textsc{tre} and \textsc{prob} in the following discussion.

\(^1\)It should be noted that it is possible to reduce IQPs to ILPs using variable substitution. However, resulting ILPs can be exponentially larger than original IQPs. Thus, IQPs provide a strict modeling advantage compared to ILPs.
**Our Approach** In the first step, we identify the concept boundaries using a CRF (with BIO encoding). Features used by CRF include the constituents given by MetaMap (Aronson and Lang, 2010), shallow parse constituents, surface form and part-of-speech of words in a window of size 3. We also use conjunctions of the features.

After finding concept boundaries, we determine the probability distribution for each concept over 4 possible types (TEST, TRE, PROB or NULL). These probability distributions are found using a multi-class SVM classifier (Chang and Lin, 2011). Features used for training this classifier include concept tokens, full text of concept, bi-grams, headword, suffixes of headword, capitalization pattern, shallow parse constituent, Metamap type of concept, MetaMap type of headword, occurrence of concept in MeSH and SNOMED CT, MeSH and SNOMED CT descriptors.

**Inference Procedure:** The final assignment of types to concepts is determined by an inference procedure. The basic principle behind our inference procedure is: “Types of concepts which appear close to one another are often closely related. For some concepts, type can be determined with more confidence. And relations between concepts’ types guide the inference procedure to determine the types of other concepts.” We will now explain it in more detail with the help of examples. Figure 1 shows two sentences in which the concepts are shown in brackets and correct (gold) types of concepts are shown above them.

First, consider first and second concepts in Figure 1a. These concepts follow the pattern: [Concept1] gave positive evidence for [Concept2]. In clinical narratives, such a pattern strongly suggests that Concept1 is of type TEST and Concept2 is of type PROB. Table 1 shows more of such patterns.

**Table 1:** Some patterns that were used in constraints.

Next, consider different concepts in Figure 1b. All these concepts are separated by commas and hence, form a list. It is highly likely that such concepts should have the same type.

### 3 Modeling Global Inference

Inference is done at the level of sentences. Suppose there are \( m \) concepts in a sentence. Each of the \( m \) concepts has to be assigned one of the following types: TEST, TRE, PROB or NULL. To represent this as an inference problem, we define the indicator variables \( x_{i,j} \) where \( i \) takes values from 1 to \( m \) (corresponding to concepts) and \( j \) takes values from 1 to 4 (corresponding to 4 possible types). \( p_{i,j} \) refers to the probability that \( i^{th} \) concept is of \( j^{th} \) type.

So, we can write the following optimization problem to find the optimal concept types:

\[
\max \sum_{i=1}^{m} \sum_{j=1}^{4} x_{i,j} \cdot p_{i,j} \quad (1)
\]

subject to

\[
\sum_{j=1}^{4} x_{i,j} = 1 \quad \forall i \quad (2)
\]

\[
x_{i,j} \in \{0, 1\} \quad \forall i, j \quad (3)
\]

The objective function in Equation (1) expresses the fact that we want to maximize the probability of
assignment of concept types. Equation (2) enforces the constraint that each concept has a unique type. We would refer to these as Type-1 constraints.

3.1 Constraints Used

In this subsection, we will describe two additional types of constraints (Type-2 and Type-3) that were added to the optimization procedure described above. Whereas Type-1 constraints described above were formulated as hard constraints, Type-2 and Type-3 constraints are formulated as soft constraints.

3.1.1 Type-2 Constraints

Certain constructs like comma, conjunction, etc. suggest that the 2 concepts appearing in them should have the same type. Figure 1b shows an example of such type of constraints. Suppose, there are \( n_2 \) such constraints. Also, assume that \( l^{th} \) constraint says that the concepts \( R_l \) and \( S_l \) should have the same type. Now, we define a variable \( w_l \) as follows:

\[
w_l = \sum_{m=1}^{4} (x_{R_l,m} - x_{S_l,m})^2
\]

Now, if the concepts \( R_l \) and \( S_l \) have the same type, then \( w_l \) would be equal to 0. Also, if the concepts \( R_l \) and \( S_l \) don't have the same type, then \( w_l \) would be equal to 2. So, \( l^{th} \) constraint can be enforced by subtracting \( (\rho_2 \cdot w_l) \) from the objective function given by Equation (1). Thus, a penalty of \( \rho_2 \) would be enforced iff \( l^{th} \) constraint is violated.

3.1.2 Type-3 Constraints

Some short patterns suggest possible types for the concepts which appear in them. Each such pattern, thus, enforces constraint on the types of concepts which appear in them. Figure 1a shows an example of such type of constraints. Suppose there are \( n_3 \) such constraints. Also, assume that the \( k^{th} \) constraint says that the concept \( A_{1,k} \) should have the type \( B_{1,k} \) and that the concept \( A_{2,k} \) should have the type \( B_{2,k} \). Equivalently, \( k^{th} \) constraint says the following in boolean algebra notation: \((x_{A_{1,k},B_{1,k}} = 1) \land (x_{A_{2,k},B_{2,k}} = 1)\). For \( k^{th} \) constraint, we introduce one more variable \( z_k \in \{0,1\} \) which satisfies the following condition:

\[
z_k = 1 \iff x_{A_{1,k},B_{1,k}} \land x_{A_{2,k},B_{2,k}}
\]

Using boolean algebra, it is easy to show that Equation (5) can be reduced to a set of linear equalities. Thus, we can incorporate the \( k^{th} \) constraint in the optimization problem by adding to it the constraint given by Equation (5) and by subtracting \((\rho_3 (1 - z_k))\) from the objective function given by Equation (1). Thus, a penalty of \( \rho_3 \) is imposed iff \( k^{th} \) constraint is not satisfied \((z_k = 0)\).

3.2 Final Optimization Problem - An IQP

After incorporating all the constraints mentioned above, the final optimization problem (an IQP) is shown in Figure 2. We used Gurobi toolkit to solve such IQPs. In our case, it solves 76 IQPs per second on a quad-core server with Intel Xeon X5650 @ 2.67 GHz processors and 50 GB RAM.

4 Experiments and Results

4.1 Datasets and Evaluation Metrics

For our experiments, we used the datasets provided by i2b2/VA team as part of 2010 i2b2/VA shared task (Uzuner et al., 2011). The datasets used for shared task contained de-identified clinical reports from three medical institutions: Partners Healthcare (PH), Beth-Israel Deaconess Medical Center (BIDMC) and the University of Pittsburgh Medical Center (UPMC). UPMC data was divided into 2 sections, namely discharge (UPMCD) and progress notes (UPMCP). A total of 349 training reports and 477 test reports were made available to the participants. However, data which came from UPMC (more than 50% data) was not made available for public use. As a result, we had only 170 clinical reports for training and 256 clinical reports for testing. Table 3 shows the number of clinical reports made available by different institutions. The strikethrough text in this ta-
Table 2: Our final system, BKC, consistently performed the best among all 4 systems (B, BK, BC and BKC).

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>BK</th>
<th>BC</th>
<th>BKC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
</tr>
<tr>
<td>TEST</td>
<td>92.4</td>
<td>79.4</td>
<td>85.4</td>
<td>91.9</td>
</tr>
<tr>
<td>TRE</td>
<td>92.1</td>
<td>73.6</td>
<td>81.8</td>
<td>92.0</td>
</tr>
<tr>
<td>PROB</td>
<td>83.6</td>
<td>83.6</td>
<td>83.6</td>
<td>88.9</td>
</tr>
<tr>
<td>OVERALL</td>
<td>88.4</td>
<td>79.4</td>
<td>83.6</td>
<td>90.7</td>
</tr>
</tbody>
</table>

Table 3: Dataset Characteristics

<table>
<thead>
<tr>
<th></th>
<th>PH</th>
<th>BIDMC</th>
<th>UPMC</th>
<th>UPMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>97</td>
<td>73</td>
<td>98</td>
<td>84</td>
</tr>
<tr>
<td>Test</td>
<td>133</td>
<td>123</td>
<td>140</td>
<td>140</td>
</tr>
</tbody>
</table>

Figure 3: These figures show the result of tuning the penalty parameters ($\rho_2$ and $\rho_3$) for soft constraints.

4.2 Results

In this section, we would refer to following 4 systems: (1) Baseline (B), (2) Baseline + Knowledge (BK), (3) Baseline + Constraints (BC) and (4) Baseline + Knowledge + Constraints (BKC). Please note that the difference between B and BK systems is that the B system (unlike BK system) doesn’t use features derived from domain-specific knowledge sources (namely MetaMap, UMLS, MeSH and SNOMED CT) for training the classifiers. Both B and BK systems do not use the inference procedure. BKC system uses all the features and also the inference procedure. In addition to these 4 systems, we would refer to another system, namely, BKC-HARD. This is similar to BKC system. However, it sets $\rho_2 = \rho_3 = 1$ which effectively turns Type-2 and Type-3 constraints into hard constraints by imposing very high penalty.

4.2.1 Importance of Soft Constraints

Figures 3a and 3b show the effect of varying the penalties ($\rho_2$ and $\rho_3$) for Type-2 and Type-3 constraints respectively. These figures show the F1-score of BKC system on the development set. Penalty of 0 means that the constraint is not active. As we increase the penalty, the constraint becomes stronger. As the penalty becomes 1, the constraint becomes hard in the sense that final assignments must respect the constraint.

We observe from Figures 3a and 3b that for Type-2 and Type-3 constraints, global maxima is attained at $\rho_2 = 0.6$ and $\rho_3 = 0.3$ respectively.

Hard vs Soft Constraints Table 4 compares the performance of BKC-HARD system with that of BKC system. First 3 rows in this table show the performance of both systems for the individual categories (TEST, TRE and PROB). Fourth row shows the overall score of both systems. BKC system outperformed BKC-HARD system on all the categories by statistically significant differences at $p = 0.05$ according to Bootstrap Resampling Test (Koehn, 2004). For the OVERALL category, BKC system improved over BKC-HARD system by $(86.1 - 85.1 = )1.0$ F1 points.

4.2.2 Comparing with state-of-the-art baseline

In 2010 i2b2/VA shared task, majority of top systems were CRF-based models. So, we decided to use CRF as our baseline. Table 2 compares the performance of 4 systems: B, BK, BC and BKC. As pointed out before, our BK system uses all the knowledge-based features and is very similar to the
Table 4: Soft constraints (BKC) consistently perform much better than hard constraints (BKC-HARD).

<table>
<thead>
<tr>
<th></th>
<th>BKC-HARD</th>
<th>BKC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>84.7</td>
<td>85.8</td>
</tr>
<tr>
<td>TRE</td>
<td>84.7</td>
<td>85.7</td>
</tr>
<tr>
<td>PROB</td>
<td>85.6</td>
<td>86.7</td>
</tr>
<tr>
<td>OVERALL</td>
<td>85.1</td>
<td>86.1</td>
</tr>
</tbody>
</table>

Figure 4: This figure shows the effect of training data size on performance of concept recognition.

top-performing systems in i2b2 challenge. We see from Table 2 that BKC system consistently performed the best for individual as well as overall categories\(^2\). This result is statistically significant at \(p = 0.05\) according to Bootstrap Resampling Test (Koehn, 2004). It is also to be noted that BC system performed significantly better than B system for all the categories. Thus, the constraints are helpful even in the absence of knowledge-based features. Since we report results on publicly available datasets, the future works would be able to compare their results with ours.

4.2.3 Effect of training data size

In Figure 4, we report the overall F1-score on a part of the development set as we vary the size of training data from 40 documents to 130 documents. We notice that the performance increases steadily as more and more training data is provided. This suggests that if we could train on full training data as was made available during challenge, the final scores could be much higher. We also notice from the figure that BKC system consistently performs better than state-of-the-art BK system as we vary the size of training data. This shows that the joint inference procedure designed by us is very robust.

5 Discussion and Related Work

Joint inference approaches which incorporate declarative knowledge in statistical models have been widely used in last few years to solve IE tasks. Some of the representative models for joint inference include posterior regularization (PR) (Ganchev et al., 2010), generalized expectations (GE) (Mann and McCallum, 2007; Mann and McCallum, 2008), constraint-driven learning (CoDL) (Chang et al., 2007), methods based on integer programs (Roth and Yih, 2004), gibbs sampling (Finkel et al., 2005) and recently the methods that are based on dual-decomposition (Reichart and Barzilay, 2012). Among these approaches, PR, GE and CoDL were proposed for semi-supervised setting. However, in this paper, we are considering a fully supervised scenario.

Roth and Yih (2004) suggested the use of integer programs to model joint inference in a fully supervised setting. Their approach is most closely related to ours. However, they used only hard constraints in their inference formulation. Chang et al. (Chang et al., 2012) recently used soft constraints in Constrained Conditional Models. However, unlike us, they performed approximate inference using beam search. In this paper, we showed that it is possible to do exact inference efficiently even while using soft constraints.

Conclusion

This paper presented a global inference strategy (using IQP) for concept recognition which allows us to model structural knowledge of the clinical domain as soft constraints in the optimization framework. Our results showed that soft constraints are much more effective than hard constraints.

Acknowledgments

This research was supported by Grant HHS 90TR0003/01 and by the IARPA FUSE Program via Department of Interior National Business Center contract number D11PC2015. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the HHS, IARPA, DoI/NBC or the US government.

\(^2\)Please note that the results reported in Table 2 can not be directly compared with those reported in the challenge because we had only a fraction of the original training and testing data.
References


