Legal Claim Identification: Information Extraction with Hierarchically Labeled Data

Mihai Surdeanu, Ramesh Nallapati and Christopher Manning

Stanford University
{mihais,nmramesh,manning}@cs.stanford.edu

Abstract

This paper introduces a novel Information Extraction problem, where only parts of documents have relevance and linguistic annotations are available only for these segments. The data is hierarchical: the top layer marks the relevant text segments and the bottom layer annotates domain-specific entity mentions, but only in the segments marked as relevant in the top layer. We investigate this problem in the legal domain, where we extract the text corresponding to litigation claims and entity mentions such as patents and laws in each claim. Because entity mentions are not labeled outside claims in training data, a top-down approach that extracts claims first and entity mentions next seems the most natural. However, we show that other models are superior. Using a simple semi-supervised approach we implement a bottom-up Conditional Random Field model; we also implement a joint hierarchical CRF using a combination of pseudo-likelihood and Gibbs sampling. We show that both these models significantly outperform the top-down approach.

1. Introduction

Most state-of-the-art supervised Information Extraction (IE) approaches can be classified in two classes: *flat* extractors, which segment text into relevant regions, e.g., named entity mentions (Sang and Meulder, 2003) or elements of seminar announcements (Freitag, 1998), or *deep* extractors, which construct complex domain-specific semantic representations of content, e.g., the scenarios proposed by the Message Understanding Conference (MUC)[1] or the events and relations promoted by the Automatic Content Extraction (ACE) evaluations[2]. While the latter class of approaches are closer to true natural language understanding, such systems have not yet achieved commercial acceptance due to their relatively poor performance.

In this paper we argue that representations of intermediate complexity are more attractive for practical applications. Motivated by a real-world IE domain, we propose a novel IE task composed of two subtasks or layers: in the first layer we extract text segments relevant to the given domain and in the second layer we extract important entities[3] from these segments. Figure 1 shows a hypothetical example with such annotations. An important observation is that, for practicality, we implement a hierarchical annotation process, i.e., entities are annotated only inside regions of interest. This essentially yields an asymmetric task: while the top layer is fully annotated, the bottom layer has only partial annotations, i.e., many entities outside relevant regions are left unlabeled.

There are many domains where such a framework is useful. For example, somebody interested in the 2008 Olympic Games may want to extract only the relevant passages and corresponding entities from articles about Beijing, e.g., players, venues, dates, etc. Technology-savvy blog readers may be interested only in blog passages related to technology and entities such as gadget names and prices. In this paper, we focus on a third domain: Intellectual Property (IP) litigation, where we extract the text corresponding to litigation claims from pleading documents and the relevant entities inside each claim, e.g., patents and laws (see Figure 2 for an example). This task is motivated by several immediate applications: case summarization, semi-structured search inside claim texts, structured search over claim entities, visualization of the inter-party relations, e.g., who infringes whose patent.

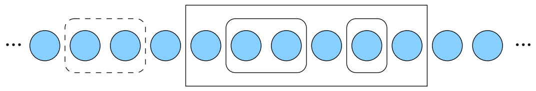

Figure 1: An example of text with hierarchical annotations. Individual words are circles, relevant text regions are rectangles, and the embedded entity mentions are rectangles with rounded corners. Entity mentions also occur outside of regions of interest and are represented here with dashed lines, i.e., they are unlabeled.

The contribution of this paper are two fold:

- We introduce a novel IE task motivated by a real-world application. We evaluate the constructed systems on a legal domain using data from actual case documents. The data is noisy: it comes from PDF documents converted automatically to text or from scanned documents converted to text using an Optical Character Recognition (OCR) system.
- Although the hierarchical nature of the task seems to impose a top-down approach, we show that other less intuitive models are preferable. Using a simple semi-supervised approach that addresses the missing labels in the entity layer we implement a bottom-up Conditional Random Field (CRF) (Lafferty et al., 2001) model. We also implement a joint hierarchical CRF model that extracts the two layers jointly using a combination of pseudo-likelihood and Gibbs sampling. We show that both these models outperform the top-down approach significantly.

The paper is organized as follows. Section 2 describes the IE task with a focus on the legal domain. Section 3 introduces the proposed models. Section 4 shows the results

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

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

[3] Throughout the paper we will use "entities" to stand for "entity mentions", for brevity.

...
31. On February 20, 2007, the USPTO duly and legally issued United States Pa\tent No. 7,179,046 B2 ("the '046 patent"), also entitled "Fan array fan section in air-handling
8

systems." Huntair is the owner by assignment of all right, title and interest in and to the '046 patent. A copy of the '046 patent is attached to the Complaint as Exhibit A.
$\{^{\texttt{ClaimBegin}}$[**FIRST COUNTERCLAIM**]$_{\texttt{ClaimNumber}}$ [**INFRINGEMENT**]$_{\texttt{ClaimType}}$ OF [**U.S. PATENT NO. 7,137,775 B2**]$_{\texttt{Patent}}$ 32. Huntair repeats and realleges paragraphs 26-31 as though fully set forth\ herein.
33. Upon information and belief, Plaintiff [**is and continues to be directly infringing,**
contributorily infringing, and/or inducing infringement]$_{\texttt{ClaimType}}$ of the [**'775 patent**]$_{\texttt{Patent}}$ by, among other things, making, using, offering to sell, selling and/or
importing, without authority or license from
Plaintiff, fan arrays in this district and elsewhere in the United States, which embody, incorporate, or otherwise practice one or more claims of the [**'775 patent**]$_{\texttt{Patent}}$.
34. Upon information and belief, in its bid to obtain a contract to install an array of\
fans at facilities owned by Amcol in Chicago, Illi
nois, Plaintiff offered to utilize a fan system
that contains, embodies, and employs the invention described and claimed in the [**'775 patent**]$_{\texttt{Patent}}$.
35. Plaintiff's conduct constitutes infringement, as provided by [**35 U.S.C. $ 271**]$_{\texttt{Law}}$, of
one or more claims of the [**'775 patent**]$_{\texttt{Patent}}$.
36. As a result of this infringement, Huntair has been damaged and deprived of the
gains and profits to which it is entitled. Furthermore, Huntair will continue to be damaged unless
this Court enjoins Plaintiff's infringing conduct.$^{\texttt{ClaimEnd}}\}$
$\{^{\texttt{ClaimBegin}}$[**SECOND COUNTERCLAIM**]$_{\texttt{ClaimNumber}}$ [**INFRINGEMENT**]$_{\texttt{ClaimType}}$ [**OF U.S. PATENT NO. 7,179,046 B2**]$_{\texttt{Patent}}$ 37. Huntair repeats and realleges paragraphs 26-31 as though fully set forth herein.
...

Figure 2: A representative example of an annotated pleading document from an IP litigation case. Claim boundaries are marked with $\{^{\texttt{ClaimBegin}}$ and $^{\texttt{ClaimEnd}}\}$. Claim entities are in bold face and delimited by squared parentheses, e.g., [...]$_{\texttt{Patent}}$. Party names are not annotated because they are available in the case meta data.

of our empirical evaluation. Section 5 summarizes related work and Section 6 concludes the paper.

2. Problem Description

We start this section with a description of the IP litigation domain, as a concrete instance of the proposed IE task. Figure 2 shows an example annotated document from this domain. Other than adding the annotation labels and using bold face for entity mentions we preserved the format of the original document. The figure illustrates several of the issues that plague this data: incorrect pagination, e.g., new paragraphs created in the middle of sentences, missing or extraneous characters, e.g., *"Pa\tent"*, broken words, e.g., *"Illi nois"*, etc.

The domain has two layers of annotations. In the top layer we annotate the claim text regions, shown between $\{^{\texttt{ClaimBegin}}$ and $^{\texttt{ClaimEnd}}\}$ in the figure. The claim segments contain all the text that is vital to understand the claim (e.g., who infringes which patent) but no extraneous material (e.g., background information about the parties involved in the case or the relief sought). Ideally, these are separated sections in a pleading document, but in practice, it is common that this information be mixed. This makes the processing of pleading documents a non-trivial process, and is further motivation for an automated extraction system. The bottom layer annotates important entities inside claims:

`Patent` (P) – contains references to patent numbers, such as *"United States Patent No. 6,190,044"* or *" '044 patent"*.

`Law` (L) – marks references to both federal and state laws, including sections and sub-sections, e.g., *"35 U.S.C. $ 281, 283, 284, and 285"* or *"California 7 Business & Professions Code $ 17200, et seq."*. Here the $ sign is a typical error of our pre-processing system, which often fails to recognize the section mark symbol (§).

`ClaimNumber` (N) – annotates the numbered header that usually marks the beginning of the claim, e.g., *"First cause of action"*, *"Second claim for relief"*. These headers uniquely identify a claim, but they are often missing.

`ClaimType` (T) – identifies the type of the parent claim. It is typically instantiated by verbal phrases or verb nominalizations (see figure). These are obviously not entity mentions; they are more reminiscent of ACE event anchors. However, for brevity, we will refer to all these four segment types as "entities" throughout the paper.

From this domain definition we drew several important observations that drove the design of our IE models. First, because the relevant text segments (e.g., claims) are likely to cover several sentences or paragraphs, the extractors in the top layer must model the text at a granularity larger than individual words. As a proof of concept we ran a state-of-the-art Conditional Random Field (CRF) sequential tagger trained at word level for the task of extracting the claim regions. The performance was very low: approximately 5 F_1 points.[4] Based on this observation, we design our extractors for the top layer to use sentences as the atomic elements. Second, although entities can occur both inside and

[4] We detail our evaluation metrics in Section 4.

outside relevant text regions, during training entity tags are only available for sentences that are tagged as belonging to a segment of interest (e.g., claim). This was done because typically the entities of interest in the given domain are the ones mentioned inside relevant text regions (e.g., we are only interested in the infringed patents) and focusing on this content saves significant annotation effort[5]. This indicates that the most natural approach for this task follows a top-down architecture: first extract claim segments, and then extract the relevant entities from these claims. And finally, entities occur outside relevant text regions as well and it is reasonable to assume that they occur in stylistically similar text (after all, it is written by the same person) and some of the context is shared (Figure 2 shows that some of the claim patents are mentioned outside as well). Hence there is potential benefit in modeling the entities outside claims as well. This motivates our semi-supervised model introduced in the next section.

3. Models

In the following subsections, we will describe several architectures that model this problem, starting with the simplest first. All the architectures use Conditional Random Fields (Lafferty et al., 2001) as a fundamental building block. We model both layers using first-order CRF taggers, using the Begin (B) – Inside (I) – Outside (O) notation to mark relevant segments in both layers, i.e., 'B' is assigned to elements (sentences or words, depending on the layer) that begin a relevant segment, 'I' is assigned to other elements inside the segment, and 'O' labels elements outside any relevant snippet.

In the top layer, the claim tag for each sentence s is represented by a discrete random variable C_s, and it takes values from the set $\{B, I, O\}$. We also denote the sequence of claim tags in a given document d by the vector $\mathbf{C}_d$. In the entity layer, $E_i \in \{\{B, I\} \times \{N, T, P, L\}\} \cup \{O\}$ represents the entity tag for the word at i^{th} position in a sentence. In other words, each word can be in the beginning ('B') or inside('I') of one of the four entity types or just be a non-entity (captured by the 'O' tag). We also represent the sequence of entity tags in a given sentence s by $\mathbf{E}_s$. $\mathbf{X}_d$ denotes the entire document text while $\mathbf{X}_s$ represents the text in sentence s, and X_i represents the i^{th} word in that sentence. We will use lower case letters to denote the values assumed by random variables (*e.g.*: c, $\mathbf{e}$, and x for a claim, an entity sequence and a textual token respectively). In addition, we use bold faced notation to represent sequences and regular faces to represent singleton tokens (*e.g.*: $\mathbf{C}$ for claim tag sequence and C for a singleton claim tag). We will omit subscripts where it is clear from the context.

3.1. Top-Down CRF

The top-down CRF is a simple architecture that closely mirrors the annotation process. In this approach, we train two independent CRFs which we call Claim CRF and Entity CRF. The Claim CRF operates on the whole document and considers each sentence as the smallest unit. It models the probability of claim tags sequence $\mathbf{C}_d$ for the document d conditioned only on text $\mathbf{X}_d = \mathbf{x}_d$, represented as $P(\mathbf{C}_d|\mathbf{x}_d)$.

The Entity CRF operates at the sentence level and considers each word as its smallest unit. For each sentence s, the Entity CRF models $P(\mathbf{E}_s|\mathbf{x}_s, c_s)$, the probability of its entity tag sequence $\mathbf{E}_s$ conditioned on the sentence text $\mathbf{x}_s$ as well as the corresponding claim tag $C_s = c_s$. The Entity CRF trains only from data inside claims because there is no labeled data available for entities outside claims.

At inference time, we first run the Viterbi algorithm for inference on the Claim CRF to generate the predicted claim sequence $\mathbf{c}_d^{(p)}$ for the whole document d. Then, we run inference for Entity CRF on each sentence s labeled as 'B' or 'I' by the Claim CRF, conditioned on the text $\mathbf{x}_d$, to output its predicted entity tag sequence $\mathbf{e}_s^{(p)}$.

The top-down model can be visualized from Figure 3, which displays a generic representation of all models discussed in this paper. The broken arrows from claims to entities in the figure correspond to this model and represent flow of information from claims to entities.

The probabilities modeled by the Claim CRF and the Entity CRF, and the inference order are summarized in row 1 of Table 1.

3.2. Bottom-up CRF

In the previous approach, the Claim CRF is ignorant of the underlying entities in the next layer. It is conceivable that the performance of the top layer Claim CRF could be improved by transmitting to it the entity information in each sentence, e.g., it is more probable to see references to patent numbers or statutes inside claim texts.

As a natural first approach, we use a bottom-up architecture as follows: for each sentence s, the Entity CRF models the probability of the entity sequence $\mathbf{E}_s$ conditioned only on the observed text sequence $\mathbf{x}_s$, given by $P(\mathbf{E}_s|\mathbf{x}_s)$. The Claim CRF, on the other hand, models for each document d, $P(\mathbf{C}_d|\mathbf{x}_d, \mathbf{e}_d)$, the probability of the claim sequence $\mathbf{C}_d$ conditioned on the entire observed document text $\mathbf{x}_d$ and the entity tag sequence of the entire document $\mathbf{e}_d$.

At inference time, we first run inference on the entity sequence using the Entity CRF to produce predicted entity tag sequence $\mathbf{e}_d^{(p)}$ and then run inference on the Claim CRF conditioned on these entity tags, to generate the predicted claim tag sequence $\mathbf{c}_d^{(p)}$. As a post-processing step, we remove the entity tags $\mathbf{e}^{(p)}$ that are outside the claims to output the final entity tags $\mathbf{e}_d^{(\text{constraints})}$. This additional cleaning up process for entities is necessitated in the bottom-up approach to satisfy the problem constraints that entities occur only inside claims.[6] The exact models for claims and entities for this architecture, and the inference order are displayed in row 2 of Table1.

This model will result in inferior performance owing to the missing entity labels outside claims. To elaborate, since the Entity CRF in this bottom-up architecture is oblivious

[5] A latent assumption is that most of the text is outside claims. This is why there are significant savings in not marking entities outside claims.

[6] Recall that in the top-down approach, the Entity CRF was conditioned on the claim tags, so it would learn to label entities only inside claims.

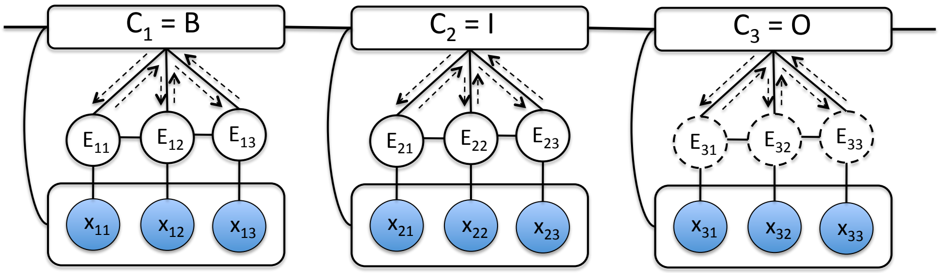

Figure 3: Generic graphical representation of all the models discussed in this paper: the top nodes represent the claim layer, the middle layer represents the entity layer and the bottom layer is text. Each node in the entity layer corresponds to a word, while each node in the claim layer corresponds to a sentence. The text nodes are darkly shaded because they are observed. The broken entity nodes in the third sentence, labeled in the top layer as outside claim ('O'), indicate that, outside claim mentions, entities are unlabeled at training time and ignored at test time. The edges correspond to the dependencies captured by the model (we removed some non-essential edges to prevent clutter). There are three types of edges between claims and entities: (a) the broken arrows from claims to entities represent the top-down pipelined system, (b) the broken arrows from entities to claims represent the two bottom-up pipelined systems and (c) the solid undirected edges represent the joint hierarchical model.

	Architecture	Claim Model	Entity Model(s)	Order of Inference
1	Top-down	$P(\mathbf{C}_d \mid \mathbf{x}_d)$	$P(\mathbf{E}_s \mid \mathbf{x}_s, c_s)$	$\mathbf{c}^{(p)} \rightarrow \mathbf{e}^{(p)}$
2	Bottom-up	$P(\mathbf{C}_d \mid \mathbf{e}_d, \mathbf{x}_d)$	$P(\mathbf{E}_s \mid \mathbf{x}_s)$	$\mathbf{e}^{(p)} \rightarrow \mathbf{c}^{(p)} \rightarrow \mathbf{e}^{(\text{constraints})}$
3	Semi-sup. Bottom-up	$P(\mathbf{C}_d \mid \mathbf{e}_d^{(\text{semi})}, \mathbf{x}_d)$	$P(\mathbf{E}_s \mid \mathbf{x}_s)$	$\mathbf{e}^{(p)} \rightarrow \mathbf{c}^{(p)} \rightarrow \mathbf{e}^{(\text{constraints})}$
4	Semi-sup. Joint Hierarchical	$P(\mathbf{C}_d \mid \mathbf{e}_d^{(\text{semi})}, \mathbf{x}_d)$	$P(\mathbf{E}_s \mid \mathbf{x}_s)$, $P(\mathbf{E}_s^{(\text{semi})} \mid \mathbf{x}_s, c_s)$	$\mathbf{e}^{(p)} \leftrightarrow \mathbf{c}^{(p)} \rightarrow \mathbf{e}^{(\text{constraints})}$

Table 1: Various architectures and their corresponding models.

to the claim information, at inference time, it is free to assign entity tags $\mathbf{e}^{(p)}$ in any sentence irrespective of its claim tag. Furthermore, since the Claim CRF is conditioned on labeled entities at training time, and since there are no labeled entities outside claims in training data, it learns that sentences that contain entities are very likely to be claims. Hence, performing inference on the Claim CRF conditioning on $\mathbf{e}^{(p)}$ may result in a large number of false positives for claims. In the next subsection, we will present a modified bottom-up architecture that will address the problem of missing labeled data in the entity layer.

3.3. Semi-supervised Bottom-up CRF

The bottom-up approach is problematic because the hierarchical nature of labeling generates partial entity labels in the annotated data, which may inject an unreasonable bias in the Claim CRF. If entity labels were available outside claims, the Claim CRF conditioned on entities would learn the true correlation between the presence of entities and claim segments. Hence, in this approach, we first train the Entity CRF only on sentences labeled as claims, and run it on the entire training set to generate predicted labels $\mathbf{e}^{(p)}$. We augment the labeled entities $\mathbf{e}$ from inside claims with $\mathbf{e}^{(p)}$ outside the claims to generate our semi-supervised labeled entity sequence $\mathbf{e}_d^{(\text{semi})}$. We use this data to condition the Claim CRF at training time.

Thus, the only difference between the semi-supervised bottom-up approach and the bottom-up approach is that the Claim CRF trains on semi-supervised entity labels $\mathbf{e}^{(\text{semi})}$ instead of only gold entity labels $\mathbf{e}$ as shown in row 3 of Table 1. Both these models are represented in Figure 3 by the broken arrows pointing upwards, symbolizing the pipelined information flow from entities to claims.

Since this model uses entities both inside and outside claims, it can be expected to capture the true correlation between entities and claims better than the standard bottom-up approach. An additional boost in performance may be expected also because the Claim CRF, training on predicted entities, can learn additional contextual and stylistic features of entities from outside the claims. Note that the standard bottom-up CRF presented above did not have this advantage.

3.4. Semi-supervised Joint Hierarchical CRF

The pipelined approaches discussed thus far model only one-way flow of information from one layer to the other. It is reasonable to assume that there is potential benefit in modeling both the layers jointly: the Entity CRF could recognize the relevant entities better, knowing whether it is inside or outside a claim, while the Claim CRF could tag the claims better, knowing what type of entities are more likely to occur inside claims than outside.

The new model therefore estimates the joint probability of both $\mathbf{C}_d$ and $\mathbf{E}_d$, conditioned on the observed document text sequence $\mathbf{x}_d$. The graphical representation of this new model is shown in Figure 3 as solid undirected edges between claims and entities. The model is hierarchical by definition because the top layer of claims is at sentence level while the bottom layer is at word token level.

Although this model is more attractive than the pipelined models, exact learning is practically infeasible [7]. Hence, in

[7] The complexity of inference is $O((|L_1| \times |L_2|)^2 n)$, where L_1 is the label set for the top layer and L_2 is the label set for the bottom layer and n is the length of the sequence.

this paper, we use a variant of pseudo-likelihood for training (Besag, 1975). Pseudo-likelihood is known to be a consistent estimator of true likelihood and is known to work well in cases where local features are strong (Parise and Welling, 2005; Toutanova et al., 2003). In this method, the joint likelihood of all the variables in a model is approximated by the product of the probability of each variable, conditioned on all other variables. In our model we apply the pseudo-likelihood only between the two layers as shown below:

$$P(\mathbf{C}, \mathbf{E}|\mathbf{x}) \approx P(\mathbf{C}|\mathbf{E}, \mathbf{x})P(\mathbf{E}|\mathbf{C}, \mathbf{x}) \tag{1}$$

This approximation makes learning efficient because each conditional probability in the right hand side of Eqn. 1 reduces to two conditional CRFs: $P(\mathbf{C}|\mathbf{E}, \mathbf{x})$ is the Claim CRF conditioned on entities while $P(\mathbf{E}|\mathbf{C}, \mathbf{x})$ is the Entity CRF conditioned on claims, both of which can be estimated using exact methods for CRFs.

Similar to the semi-supervised bottom-up approach, we train the Claim CRF $P(\mathbf{C}|\mathbf{E}, \mathbf{x})$ conditioned on semi-supervised entity labels $\mathbf{e}^{(\text{semi})}$ as shown in row 4 of Table 1. The symmetric nature of the joint model leaves us no choice but to train the Entity CRF also on $\mathbf{e}^{(\text{semi})}$, as shown in the same row of Table 1. We also list an unconditioned Entity CRF $P(\mathbf{E}|\mathbf{x})$ as an additional model used in this architecture because it is required to generate $\mathbf{e}^{(\text{semi})}$ at training and $\mathbf{e}^{(p)}$ at testing time.

Since exact inference is computationally expensive as well, we use Gibbs sampling (Andrieu et al., 2003) to perform approximate inference, since it has many interesting parallels with pseudo-likelihood. Like pseudo-likelihood, Gibbs sampling deals with local probability of each variable, conditioned on all other variables.[8] In this approach, we sample each variable in turn from its probability conditioned on its latest assignments of its neighbors. This iterative process, when run long enough is guaranteed to converge to the true posterior.

In our case, since we have a two tier hierarchy, in each iteration, we successively sample all the variables in one layer then move to the other layer. Also, since we need best variable assignments rather than true posterior, we use *simulated annealing* with Gibbs sampling, using a linear *cooling schedule*, as proposed in (Finkel et al., 2005).

4. Experimental Results

We start this section by describing the experimental settings, we continue with a description of the feature set used in both subtasks, and we conclude with a discussion of the experimental results.

4.1. Data

The corpus used in this paper contains 90 pleading documents from actual IP litigation cases. The documents are either PDF documents converted to text (for newer cases) or scanned documents converted to text using an OCR system (for older cases). A significant amount of noise was introduced in the data by this process. The corpus was preprocessed using an in-house tokenizer and sentence boundary detector. The sentence boundary was adapted to the pagination of this corpus, e.g., it introduces sentence breaks at two consecutive new line characters even if no punctuation mark exists. The resulting tokenized text was part-of-speech (POS) tagged using the Stanford POS tagger[9]. Lastly, the corpus was annotated by an IP litigation expert, who followed strict annotation guidelines designed by a multi-disciplinary group of experts from both Law and Computer Science. Table 2 summarizes the corpus statistics.

This corpus was randomly split into a training partition (70%) and a testing partition (30%). We were careful not to have documents from the same case in both training and testing.[10] This yielded a training corpus of 64 documents and a testing set of 26 documents.

4.2. Evaluation Metrics

As evaluation metrics we used the standard precision, recall, and F_1 scores coupled with a strict-match criterion in the spirit of the CoNLL evaluations (Sang and Meulder, 2003). In other words, an extracted segment is considered correct if it matches exactly the tokens in the corresponding annotation and it has the correct label.

4.3. Features

For Entity CRF we used a modified version of the Stanford Named Entity Recognition (NER) software[11] (Finkel et al., 2005). We used its default feature set consisting of: (a) word, (b) part of speech (POS) tag, and (c) word-shape, where the word shape captures the case of the alpha characters in the word, collapses sequences of the same type, but maintains punctuation. These features are extracted from the current word and its immediate context, i.e., the previous and following word. We extended this feature set with only one new feature: the claim tag of the current sentence c_s (for the top-down and joint approaches).

For Claim CRF, we used three feature groups: (a) sentence words, (b) number of new-line characters preceding the sentence (as an approximation of pagination), and (c) the entity tags in the sentence $\mathbf{e}_s$ (for the bottom-up and joint approaches). These features are extracted from the current sentence, the previous two and the following two sentences. Note that we did not tune any of these features in any manner.

4.4. Results and Discussion

Table 3 lists the overall results of the proposed architectures and of three oracle systems. Each oracle system trains only one layer and uses gold information in the other layer during both training and inference, e.g., the claim oracle is a bottom-up system that has access to gold entity labels. The difference between the two entity oracles is that one is fully supervised whereas the other one is semi-supervised, i.e.,

[8] This reduces to a logistic regression model of probability of each variable given its neighbors, in case of undirected exponential models such as ours.

[9] http://nlp.stanford.edu/software/tagger.shtml

[10] The 90 documents came from only 49 cases, so this was an important constraint.

[11] http://nlp.stanford.edu/software/CRF-NER.shtml

Documents	Sentences	Words	Claims	ClaimNumbers	ClaimTypes	Patents	Laws
90	25,250	548,402	362	319	579	1292	433

Table 2: Corpus statistics.

	Claims			Entities		
	Precision	Recall	F_1	Precision	Recall	F_1
Top-down	80.00	54.05	64.52	**86.42**	52.63	65.42
Bottom-up	60.65	50.81	55.29*	48.1	**60.47**	53.58*
Semi-supervised Bottom-up	**89.74**	**56.76**	**69.54***	85.34	56.65	**68.09***
Semi-supervised Joint Hierarchical	88.89	56.22	68.87*	86.16	55.69	67.65*
Claim Oracle	92.40	85.41	88.76	–	–	–
Entity Oracle	–	–	–	83.62	62.99	71.85
Semi-supervised Entity Oracle	–	–	–	85.25	61.77	71.64

Table 3: Overall scores of the proposed architectures and of several oracle models. Asterisks indicate that the difference between the corresponding score and the score of the top-down model is statistically significant. The results for the semi-supervised bottom-up and joint models are not significantly different. All significance tests are performed using two-tailed paired t-test at 95% confidence interval on 20 samples obtained using bootstrap resampling.

the latter trains on $\mathbf{E}^{(\text{semi})}$. We draw several observations from these results:

(a) The performance of the top-down model is reasonable, considering the difficulty of the task and the size and quality of the data. We attribute these results mainly to our hierarchical approach, where each layer models the text at different granularity (sentences or words).

(b) As expected, the first bottom-up approach performs quite badly. This is caused by the skewed entity distribution caused by the partial labeling of the training data, which confuses the claim classifier at inference time.

(c) The semi-supervised bottom-up system addresses this issue successfully. This is our best performing system. This proves that information propagated from the bottom layer improves the top layer significantly. Consequently, the entity layer improves as well, because $\mathbf{E}^{(\text{constraints})}$ (i.e., entities after deleting instances outside claim boundaries) are based on the predictions of the top layer.

(d) The joint model outperforms the top-down model significantly, but it does not perform better than the semi-supervised bottom-up approach. There are two potential causes for this behavior: first, the feedback from the claim model, which has low recall, may end up hurting the performance of the entity layer when computing $P(\mathbf{E}_s^{(\text{semi})}|\mathbf{x}_s, c_s)$; second, because the joint inference must use parallel labels between the two layers, the entity layer self trains on predicted entity labels for data outside of claims, and this may introduce more noise than signal. We can actually quantify the impact of these problems using the two entity oracles. The only difference between the two oracles is that the semi-supervised oracle self-trains its entity model: $P(\mathbf{E}_s^{(\text{semi})}|\mathbf{x}_s, c_s)$ versus $P(\mathbf{E}_s|\mathbf{x}_s, c_s)$. The oracle results indicate that self-training causes a performance drop of .2 F_1 points. Hence, the other .2 F_1 points in the difference between the bottom-up and joint models are caused by the feedback from the claim layer. We conjecture that both these problems are caused by insufficient training data. As more data becomes available, we expect that both self-training the entity layer and the feedback from the claim to the entity layer be successful.

(e) Nevertheless, the table indicates that the joint model improves the precision of the entity layer with respect to the semi-supervised bottom-up model. The entity precision of the joint hierarchical model is .8 points higher than that of the semi-supervised bottom-up model. This is caused again by the feedback from the claim layer to the entity layer. Event though the claim layer in the joint model has low recall, its precision is quite high. This provides precise feedback to the entity layer on where claim boundaries exist, which in turn enhances the precision of the entity layer.

(f) Despite its good performance, the claim oracle actually indicates how difficult this domain is: because gold entities are labeled only inside claims, one would expect this oracle to score close to 100 F_1 points, because any entity mention is a strong hint that the corresponding sentence belongs to a claim. The fact that the claim oracle scores only 88 F_1 points indicates that there is high ambiguity for the sentences not covered by entities.

(g) The relatively low performance of the entity oracles indicates that entity recognition in the legal domain is a hard problem, even when the task is limited at analyzing the text inside claims. We analyze the behavior of our entity models later on this section.

In order to understand the relative importance of various features in the Claim CRF, we perform ablation experiments using the semi-supervised bottom-up architecture. This test involves removing one feature-type at a time and measuring the performance. The results of the test, displayed in Table 4 show that the model is heavily lexicalized – the F1 performance of the CRF drops to as low as 36.02 when words are removed as features. The test also demonstrates that the entities contribute about 5% points in F1, indicating the utility of joint and bottom-up architectures. Surprisingly, pagination does not carry a strong signal for claim identification, and we attribute it to the noisy features resulting from the OCR translation.

Table 5 lists the scores of our best model for each entity type. The table indicates that claim numbers and patents are recognized with acceptable performance, most likely due to their simple structure. In contrast, claim types have low performance. The explanation is that claim type mentions are often complex verbal or nominal phrases, which

	Precision	Recall	F_1
All features	89.74	56.76	69.54
– lexicalization	61.84	25.41	36.02
– pagination	88.33	57.3	69.51
– entities	80.00	54.05	64.52

Table 4: Ablation experiment for the Claim CRF using the semi-supervised bottom-up architecture.

	Precision	Recall	F_1
Claim Number	97.06	54.40	69.72
Claim Type	53.97	26.25	35.32
Law	71.57	36.32	48.18
Patent	94.93	80.94	87.38

Table 5: Results for the entity layer using the semi-supervised bottom-up architecture.

are hard to model using first-order CRFs at word level. We expect more successful models to use full syntax for this entity type. Somewhat surprisingly, mentions of laws are also recognized with low performance. The most common error for this type was caused by the document pre-processor. Law mentions typically include non-ASCII characters (e.g., §), which are mistakenly converted to punctuation marks by the text converters, and these are later seen as end-of-sentence markers by our sentence boundary detector. Since the entity tagger works at sentence level, it cannot recover entities split in different sentences. This is yet another example of a problem that a real-world IE system must address.

For completeness, we show the results of the ablation experiment for Entity CRF in Table 6. To avoid the complex inter-dependencies between the two layers,[12] in this experiment we used the top-down architecture. Similarly to Table 4, this experiment shows that our models are heavily lexicalized: removing lexical features caused a drop in the F_1 score of more than 11 points. The drop is not as high as the drop reported in Table 6 because some of the lexical information is captured by the POS tag and word shape features. The features with the second highest impact are the features extracted from the context surrounding the word to be classified: ignoring this context causes a drop of approximately 3 F_1 points. These observations are consistent with previous work on named entity recognition. What is different in our domain is that POS information does not help when combined with lexicalization: removing POS features yields a slight improvement in the F_1 score. This is caused by the fact that our data is significantly different from the data used to train the POS tagger, both in quality and in domain. Because of this, using the POS tagger in this corpus generates more noise than signal.

Lastly, Figure 4 shows the learning curves for our three best scoring approaches. The curves for Claim CRF show that the bottom-up and the joint systems behave similarly. On the other hand, the top-down approach scores consistently lower, when using more than 20% of the data. For smaller training corpora, the top-down approach performs better because the entity models are not strong enough to provide useful signal in the bottom-up or joint systems. Extrapolating from this observation, we expect that the joint approach will in turn start performing better than the bottom-up one with enough training data. The bottom part of Figure 4 shows a similar story. The differences between the learning curves for Entity CRF are not that large, but they are still statistically significant for the majority of the plot points and they lead to the same conclusions.

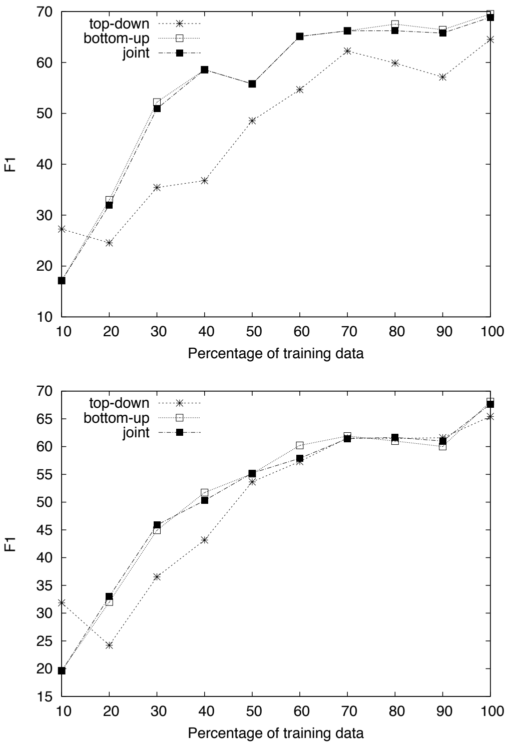

Figure 4: Learning curves of the best three models. The top chart plots the F_1 score of the Claim CRF. The bottom chart plots the F_1 score of the Entity CRF.

	Precision	Recall	F_1
All features	86.42	52.63	65.42
– lexicalization	71.12	43.61	54.07
– POS tags	89.63	51.96	65.79
– word shape	86.80	51.63	64.75
– context	86.32	49.04	62.55

Table 6: Ablation experiment for the Entity CRF using the top-down architecture. “context” indicates all features from the previous and following word. The other three experiments remove the corresponding feature group from all tokens (current, previous, and following word).

5. Related Work

In the field of IE, most body of work –too large to be cited here– falls into one of the two classes described before: flat extractors or deep, semantic extractors. The middle ground has been addressed mainly by works that investigate the recognition of nested named entity mentions, which are common in the medical domain (Alex et al., 2007) and in corpora on languages other than English (Marquez et al., 2007). There are significant differences between our work and nested NER: (a) nested NER is non-hierarchical in the sense that all layers operate at token level, (b) there are no

[12] For example, in the bottom-up architecture the claim layer depends on the performance of the entity layer, and, in turn, the output constraints for the entity layer depend on the performance of the claim layer.

missing labels in any layer. (Alex et al., 2007) also use a combination of sequential CRF classifiers, but their joint approach focuses on joint representation rather than joint modeling.

The general idea of breaking documents into "zones" with consequences for further processing is not new, e.g., Teufel and Moens used document segmentation based on rhetorical structure for the summarization of scientific articles (Teufel and Moens, 2002). A paper that is closer to ours in terms of using pipelined or joint CRFs for natural language processing from multiple layers is that of (Sutton et al., 2007). In this work, the authors used a two layer factorial CRF to jointly model noun-phrase chunking and POS tagging, and demonstrated significant performance gains compared to a pipelined system of independently trained CRFs. For the same reasons as above, we argue that our problem is more complex than theirs. The work of (McDonald et al., 2007) uses a hierarchical CRF with different levels of granularity (documents and sentences) to model coarse to fine sentiments in a document, but their data is fully observed. Recent work of (Truyen et al., 2008) indeed proposes a hierarchical CRF that incorporates missing labels. They present detailed theoretical treatment of the model in a missing labels scenario, but they test their model only on fully observed data (e.g., joint POS tagging and syntactic chunking).

6. Conclusions

This paper introduces a novel Information Extraction problem, where only parts of documents have relevance and linguistic annotations are available only for these segments. The problem has several hierarchical properties. First, the data is annotated using a two-layer hierarchy: the top layer marks the relevant text segments and the bottom layer annotates domain-specific entity mentions only in these segments. Due to this approach, the data for the bottom layer is only partially labeled, i.e., entity mentions outside of the relevant text segments are not annotated. Second, the two layers are modeled at different granularity: the top layer using the sentence as the atomic element and the bottom layer using words.

We investigate this problem on a real-world application from the IP litigation domain. We introduce two models that outperform significantly the top-down cascaded approach. Using a simple semi-supervised approach for the entity layer we implement a bottom up model and then we extend it to a joint hierarchical CRF. We discuss the advantages and limitations of all approaches.

All in all, this work shows that complex IE systems can be built and trained using hierarchical, partially-labeled data. We believe that this reduces annotation efforts, which is an important constraint in the development of any supervised IE system. To further improve the performance of our system without increasing the annotation burden on the legal experts we plan to: (a) combine our approach with unsupervised topic segmentation algorithms (Allen, 2002), which will be used to enhance our claim extractor, and (b) combine our models with rule-based systems, e.g., we expect a rule-based patent mention extractor to perform well, and to provide hints about where claim information is concentrated. On the legal side of project, in future work we will extend our entity extraction model with other entity types of interest, e.g., product names, and our claim detection model with other types of claims, e.g, trade secret or trademark violation.

Acknowledgments

We thank the Stanford Intellectual Property Litigation Clearinghouse (IPLC) project[13] for their annotation effort and Jenny Finkel for the help with her CRF implementation.

7. References

B. Alex, B. Haddow, and C. Grover. 2007. Recognising nested named entities in biomedical text. In *Proc. of BioNLP 2007*.

J. Allen. 2002. Introduction to topic detection and tracking. *Topic Detection and Tracking: Event-Based Information Organization*.

C. Andrieu, N. De Freitas, A. Doucet, and M.I. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning*.

J. Besag. 1975. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195.

J. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems through Gibbs sampling. In *Proc. of ACL*.

D. Freitag. 1998. Machine learning for information extraction in informal domains. *Ph.D. thesis, Carnegie Mellon University*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*.

L. Marquez, L. Villarejo, M.A. Marti, and M. Taule. 2007. SemEval-2007 task 09: Multilevel semantic annotation of catalan and spanish. In *Proc. of SemEval-2007*.

R. McDonald, T. Neylon K. Hannan, M. Wells, and J. Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proc. of ACL*.

S. Parise and M. Welling. 2005. Learning in Markov Random Fields: an empirical study. In *Joint Statistical Meeting*.

E.F. Tjong Kim Sang and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*.

C. Sutton, A. McCallum, and K. Rohanimanesh. 2007. Dynamic Conditional Random Fields: Factorized probabilistic models for labeling and segmenting. *The Journal of Machine Learning Research*.

S. Teufel and M. Moens. 2002. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4).

K. Toutanova, D. Klein, C. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT-NAACL*.

T.T. Truyen, D.Q. Phung, H.H.Bui, and S. Venkatesh. 2008. Hierarchical Semi-Markov Conditional Random Fields for recursive sequential data. In *Proc. of NIPS*.

[13] http://www.law.stanford.edu/program/centers/iplc/