Learning to Infer Entities, Properties and their Relations from Clinical Conversations

Nan Du, Mingqiu Wang, Linh Tran, Gang Li, and Izhak Shafran

{dunan, mingqiuwang, tranlm, leebird, izhak}@google.com

Google Inc.

Abstract

Recently we proposed the Span Attribute Tagging (SAT) Model (Du et al., 2019) to infer clinical entities (e.g., symptoms) and their properties (e.g., duration). It tackles the challenge of large label space and limited training data using a hierarchical two-stage approach that identifies the span of interest in a tagging step and assigns labels to the span in a classification step.

We extend the SAT model to jointly infer not only entities and their properties but also relations between them. Most relation extraction models restrict inferring relations between tokens within a few neighboring sentences, mainly to avoid high computational complexity. In contrast, our proposed Relation-SAT (R-SAT) model is computationally efficient and can infer relations over the entire conversation, spanning an average duration of 10 minutes.

We evaluate our model on a corpus of clinical conversations. When the entities are given, the R-SAT outperforms baselines in identifying relations between symptoms and their properties by about 32% (0.82 vs 0.62 F-score) and by about 50% (0.60 vs 0.41 F-score) on medications and their properties. On the more difficult task of jointly inferring entities and relations, the R-SAT model achieves a performance of 0.34 and 0.45 for symptoms and medications respectively, which is significantly better than 0.18 and 0.35 for the baseline model. The contributions of different components of the model are quantified using ablation analysis.

1 Introduction

The widespread adoption of Electronic Health Records by clinics across United States has placed a disproportionately heavy burden on clinical providers, causing burnouts among them (Wachter and Goldsmith, 2018; Xu, 2018; Arndt et al., 2017). There has been considerable interest, both in academia and industry, to automate aspects of documentation so that providers can spend more time with their patients. One such approach aims to generate clinical notes directly from the doctor-patient conversations (Patel et al., 2018; Finley et al., 2018a,b). The success of such an approach hinges on extracting relevant information reliably and accurately from clinical conversations.

In this paper, we investigate the tasks of jointly inferring entities, specifically, symptoms (Sx), medications (Rx), their properties and relations between them from clinical conversations. These tasks are defined in Section 2. The key contributions of the work reported here include: (i) a novel model architecture for jointly inferring entities and their relations, whose parameters are learned using the multi-task learning paradigm (Section 4), (ii) comprehensive empirical evaluation of our model on a corpus of clinical conversations (Section 6), and (iii) understanding the model performance using ablation study and human error analysis (Section 6.7). Since clinical conversations include domain specific knowledge, we also investigate the benefit of augmenting the input feature representation with knowledge graph embedding. Finally, we summarize our conclusions and contributions in Section 7.

2 Task Definitions

For the purpose of defining the tasks, consider the snippet of a clinical conversation in Table 1.

2.1 The Symptom Task (Sx)

This task consists of extracting the tuples (*symType, propType, propContent*).

The "pain" in the example in Table 1 is annotated as *symType* (sym/msk/pain), where msk stands for musculo-skeletal system. We have

DR: How often do you have pain in your arms?
PT: It hurts every morning.
DR: Are you taking anything for it?
PT: I've been taking Ibuprofen. Twice a day.

Table 1: An example to illustrate entities, properties and their relations. Entities – (sym/msk/pain: *pain*) & (meds/name: *Ibuprofen*); Properties – (symprop/freq: *every morning*) & (medsprop/freq: *twice a day*); Relations: (sym/msk/pain, symprop/freq, *every morning*), (Ibuprofen, medsprop/freq, *twice a day*).

pre-defined 186 categories for symptom types, curated by a team of practising physicians and scribes, based on how they appear in clinical notes. We deliberately abstained from the more exhaustive symptom labels such as UMLS and ICD codes (Bodenreider, 2004) in favor of this smaller set since our training data is limited.

The properties associated with the symptoms, *propType*, fall into four categories: symprop/severity, symprop/duration, symprop/location, and symprop/frequency. The *propContent* denotes the content associated with the property. In the running example, "every morning" is the content associated with the property type *symprop/frequency*.

Not all the symptoms mentioned in the course of clinical conversations are experienced by the patients. We explicitly infer the status of a symptom as experienced or not. This secondary task extracts the pair: (*symType, symStatus*).

2.2 The Medication Task (Rx)

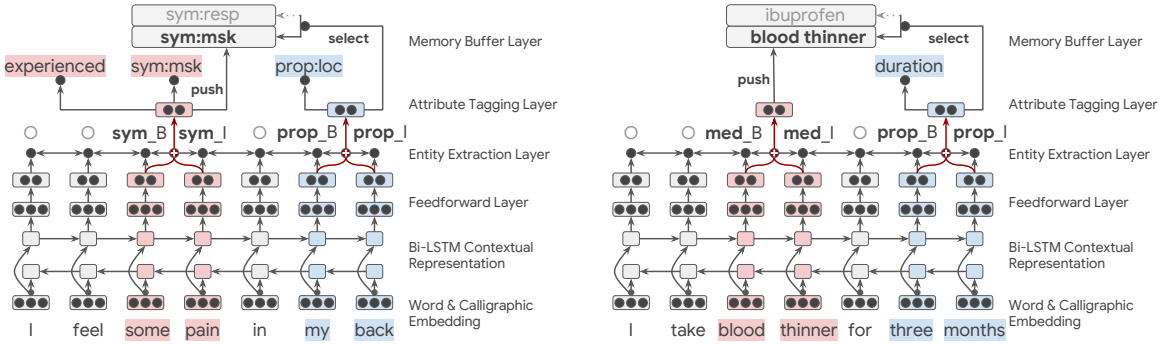
This task consists of extracting tuples of the form: (*medContent, propType, propContent*).

While symptoms can be categorized into a closed set, the set of medications is very large and continually updated. Moreover, in conversations, we would like to extract indirect references such as "pain medications" as *medContent*. We define three types of properties: medsprop/dosage, medsprop/duration and medsprop/frequency. In the running example,"twice a day" is the *propContent* of the type *medsprop/frequency* associated with the *medContent* "ibuprofen".

3 Previous Work

Relation extraction is a long studied problem in the NLP domain and include tasks such as the ACE (Doddington et al., 2004), the SemEval (Hendrickx et al., 2010), the i2b2/VA Task (Uzuner et al., 2011a), and the BioNLP Shared Task (Kim et al., 2013). Many early algorithms such as DIPRE algorithm by Brin (1998) and SNOWBALL algorithm by Agichtein and Gravano (2000) relied on regular expressions and rules (Fundel et al., 2007; Peng et al., 2014). Subsequent work exploited syntactic dependencies of the input sentences. Features from the dependency parse tree were used in maximum entropy models (Kambhatla, 2004) and neural network models (Snow et al., 2005). Kernels were defined over tree structures (Zelenko et al., 2003; Culotta and Sorensen, 2004; Qian et al., 2008). More efficient methods were investigated including shortest dependency path (Bunescu and Mooney, 2005) and sub-sequence kernels (Mooney and Bunescu, 2006). Recent work on deep learning models investigated convolutional neural networks (Liu et al., 2013), graph convolutional neural networks over pruned trees (Zhang et al., 2018), recursive matrix-vector projections (Socher et al., 2012) and Long Short Term Memory (LSTM) networks (Miwa and Bansal, 2016). Other more recent approaches include two-level reinforcement learning models (Takanobu et al., 2019), two layers of attention-based capsule network models (Zhang et al., 2019), and self-attention with transformers (Verga et al., 2018). In particular, (Miwa and Sasaki, 2014; Katiyar and Cardie, 2016; Zhang et al., 2017; Zheng et al., 2017; Verga et al., 2018; Takanobu et al., 2019) also seek to jointly learn the entities and relations among them together. A large fraction of the past work focused on relations within a single sentences. The dependency tree based approaches have been extended across sentences by linking the root notes of adjacent sentences (Gupta et al., 2019). Coreference resolution is a similar task which requires finding all mentions of the same entity in the text (Martschat and Strube, 2015; Clark and Manning, 2016; Lee et al., 2017).

In the medical domain, the BioNLP shared task deals with gene interactions and is very different from our domain (Kim et al., 2013). The *i2b2/va challenge* is closer to our domain of clinical notes, however, that task is defined on a small corpus of written discharge summaries (Uzuner et al., 2011b). Written domain benefits from cues such as the section headings which are unavailable in clinical conversations. For a wider survey of extracting clinical information from written clinical documents, see (Liu et al., 2012).



(a) Model architecture for the symptoms task.

(b) Model architecture for the medications task.

Figure 1: Variants of the R-SAT model architecture. The entity spans ("some pain", "blood thinner") are identified in a tagging step, which are pushed into a memory buffer along with their latent representation and in a subsequent step the property spans ("my back", "three months") selects the most related entity from the buffer.

4 Model

Our application requires performing multiple inferences simultaneously, that of identifying symptoms, medications, their properties and relations between them. For this purpose, we adopt the well-suited multitask learning framework and develop a model architecture, illustrated in Figure 1, that utilizes our limited annotated corpus efficiently.

4.1 Input Encoder Layer

Let $\mathbf{x}$ be an input sequence. We compute the contextual representation at the k-th step using a bidirectional LSTM, $\mathbf{h}'_\mathbf{k} = [\vec{\mathbf{h}}(\mathbf{x}_{\leq \mathbf{k}}|\vec{\Theta}_{LSTM}), \overleftarrow{\mathbf{h}}(\mathbf{x}_{\geq \mathbf{k}}|\overleftarrow{\Theta}_{LSTM})]$, which is fed into a two-layer fully connected feed-forward network. For simplicity, we drop the index k from the rest. The final features are represented as $\mathbf{h}'' = MLP(\mathbf{h}'|\Theta_{FF})$. In our task, we found that the LSTM-based encoder performs better that the transformer-based encoder (Vaswani et al., 2017; Chen et al., 2018).

Extending a Standard Tagging Model In a typical tagging model, the contextual representation of the encoder $\mathbf{h}''$ is fed into a conditional random field (CRF) layer to predict the BIO-style tags (Collobert et al., 2011; Huang et al., 2015; Ma and Hovy, 2016; Chiu and Nichols, 2016; Lample et al., 2016; Peters et al., 2017; Yang et al., 2017; Changpinyo et al., 2018). Such a model can be extended to predict the relations. For example, in the utterance, "I feel some pain in my back", we could setup the tagger to predict the association between the symptom (*sym/msk*), and its property (*symprop/loc*) using a cross-product space where "my back" is tagged with *sym/msk+symprop/loc* so that the relation prediction problem is reformulated as a standard sequence labeling task. Although this would be a viable option for tasks where the tag set is small (e.g., place, organization, etc.), the cross-product space in our Sx task is unfortunately large (e.g., 186 Sx labels $\times$ 3 Sx property types, and 186 Sx labels $\times$ 3 Sx status types).

4.2 Span Extraction Layer

We propose an alternative formulation that tackles the problem in a hierarchical manner. We first identify the span of interest using ***generic*** tags with BIO notation, namely, (*sym_B*, *sym_I*) for symptoms and (*symprop_B*, *symprop_I*) for their properties, as in Figure 1(a). Likewise, (*med_B*, *med_I*) for medications and (*medsprop_B*, *medsprop_I*) for their properties as shown in Figure 1(b). This corresponds to highlighting, for example, "some pain" and "my back" as spans of interest.

Given the latent representations, $\mathbf{h} = (\boldsymbol{h}''_\mathbf{1}, \cdots, \boldsymbol{h}''_\mathbf{N})$, and the target tag sequence $\mathbf{y}^e = (y_1, \cdots, y_N)$ (e.g., *sym_B, sym_I, O, symprop_B, symprop_I*), we use the negative log-likelihood $-\log P(\mathbf{y}^e|\mathbf{h})$ under CRF as the loss of identifying spans of interest $-S(\mathbf{y}^e, \mathbf{h}) + \log \sum_{\mathbf{y}'} \exp(S(\mathbf{y}', \mathbf{h}))$, where $S(\mathbf{y}, \mathbf{h}) = \sum_{i=0}^{N} \mathbf{A}_{y_i, y_{i+1}} + \sum_{i=0}^{N} P(\mathbf{h}_i, \mathbf{y}_i)$ measures the compatibility between a sequence $\mathbf{y}$ and $\mathbf{h}$. The first component estimates the accumulated cost of transition between two neighboring tags using a learned transition matrix $\mathbf{A}$. $P(\mathbf{h}_i, \mathbf{y}_i)$ is computed via the inner product $\mathbf{h}_i^\top \mathbf{y}_i$ where

y_i belongs to any sequence of tags $\mathbf{y}$ that can be decoded from $\mathbf{h}$. During training, the $\log P(\mathbf{y}^e|\mathbf{h})$ is estimated using forward-backward algorithm and during inference, the most probable sequence is computed using the Viterbi algorithm.

4.3 Attribute Tagging Layer

Using the latent representation of the highlighted span, we can predict one or more ***attributes*** of the span. In Figure 1(a), we can predict two attributes associated with "some pain": *sym/msk* as the symptom label and *symStatus/experienced* as its status. Similarly, in Figure 1(b), the span property span "three months" has the predicted property type *medsprop/duration*. Therefore, by forming semantic abstractions for each highlighted text span, we decompose a single complex tagging task in a large label space into correlated but simpler sub-tasks, which are likely to generalize better when the training data is limited.

Given the spans, either from the inferred or the ground truth sequence $\mathbf{y}^*$, a richer representation of the contexts can be used to predict attributes than otherwise possible. A contextual representation is computed from the starting i and ending j index of each span.

$$\boldsymbol{h}^s_{ij} = \text{Aggregate}(\boldsymbol{h}_k|\boldsymbol{h}_k \in \mathbf{h}, i \leq k < j) \quad (1)$$

where $\text{Aggregate}(\cdot)$ is the pooling function, implemented as mean, sum or attention-weighted sum of the latent states of the input encoder. The kth attributes associated with the span are modeled using $P(y^k_{attr}|\boldsymbol{h}^s_{ij})$. For example, while prediction symptom labels s_x and their associated status s_t, the target attributes are $y^0_{attr} := y^{s_x}$ and $y^1_{attr} := y^{s_t}$. For predicting medication entities r_x and their properties p_r, each span only has one attribute. Since each attribute comes from a pre-defined ontology, the multi-nomial distribution $P(y^k_{attr}|\boldsymbol{h}^s_{ij})$ can be modeled as $\text{Softmax}(\boldsymbol{h}^s_{ij}|\Theta^k)$ for each attribute.

4.4 Memory Buffer Layer

One of the critical components of our model is the memory buffer. Most previous models on joint inference of entities and relations consider all spans of entities and properties. This has the computational complexity of $\mathcal{O}(n^4)$ in the length of the input n, and makes it infeasible for application such as ours where the input could often be 1k words or more. We circumvent this problem using a memory buffer to cache all inferred candidate spans and test their relationship with inferred property spans. Note, unlike methods that cascade two such stages, our model is trained end-to-end jointly with multi-task learning.

The memory buffer saves different entries for symptom and medication tasks, as illustrated in Figure 1. At each occurrence of a symptom (medication) entity span, we push $\boldsymbol{m}^k = \text{Aggregate}(\{\boldsymbol{h}^{\boldsymbol{s}}_{\boldsymbol{ij}}, \boldsymbol{e}^{\boldsymbol{s}}\})$ into the k-th position of the memory buffer. For the symptom task, $\boldsymbol{e}^{\boldsymbol{s}}$ is the learned word embedding of one of the labels in the closed label set. In the medication case, $\boldsymbol{e}^{\boldsymbol{s}}$ is the Aggregate of learned word embedding of the verbatim sub-sequence corresponding to the medication entity.

4.5 Relation Inference Layer

Each span of inferred property in the conversation is compared against each entry in the buffer. A property entity span is represented as $\boldsymbol{y}^p = \text{Aggregate}(\{\boldsymbol{h}^{\boldsymbol{p}}_{\boldsymbol{ij}}, \boldsymbol{e}^p\})$ where $\boldsymbol{e}^p$ is the Aggregate of word embedding corresponding to the span. The multi-nomial likelihood is computed using a bilinear weight matrix $\boldsymbol{W}$. The most likely entry (k) is picked from the memory stack $\boldsymbol{M} = (\boldsymbol{m}^1, ..., \boldsymbol{m}^K)$ by maximizing the likelihood.

$$\begin{aligned} \hat{k} &= \arg\max_k P(k|\boldsymbol{y}^p) \\ &= \arg\max_k \text{Softmax}(\boldsymbol{y}^{p\top}\boldsymbol{W}\boldsymbol{M}) \end{aligned} \quad (2)$$

Remarks The computation cost of inferring relation between a property span and all the entities in the input is proportional to the memory buffer size. On our corpus, for Sx task, the mean and standard deviation per conversation was 22 and 15 respectively, and for Rx task, it was 32 and 23 respectively. Hence, the set of candidate entities considered is substantially smaller than all potential entities $O(n^2)$ in the input sequence.

The small size of the memory buffer also has an impact on rate of learning. In each training step, rather than updating all embedding, we only update a smaller number of embedding, those associated with the entries in the memory buffer. This makes the learning fast and efficient.

4.6 An End-to-end Learning Paradigm

We train the model end-to-end by minimizing the following loss function for each conversation:

$$\begin{aligned}\mathcal{L} = & - \alpha \log P(\mathbf{y}^e|\mathbf{h}) \\ & - \sum_{\{y^k_{attr} \in S\}} \log P(y^i_{attr}|\mathbf{h}) \\ & - \sum_{\{y^j_{pos} \in P\}} \log P(y^j_{pos}|\mathbf{h}) \end{aligned} \quad (3)$$

where $\mathbf{y}^e$ is the target sequence (sym_B, sym_I, $prop\_B$, $prop\_I$), $\{y^k_{attr}\}$ is the set of attribute labels for each highlighted span, $\{y^j_{pos}\}$ is the list of buffer slot indices and α is a relative weight.

During training, we are simultaneously attempting to detect the location of tags as well as classify the tags. Initially our model for locating the tags is unlikely to be reliable, and so we adopt a curriculum learning paradigm. Specifically, we provide the classification stage the reference location of the tag from the training data with probability p, and the inferred location of the tag with probability $1 - p$. We start the joint multi-task training by setting this probability to 1 and decrease it as training progresses (Bengio et al., 2015).

Since our model consists of span extraction and attribute tagging layers followed by relation extraction, we refer to our model as **Relational Span-Attribute Tagging Model** (R-SAT). One advantage of our model is that the computational complexity of joint inference is $\mathcal{O}(n)$ which is linear in the length of the conversation n. This is substantially cheaper than other previous work on joint relation prediction models where the computational complexity is $\mathcal{O}(N^4)$ (Lee et al., 2017).

5 Knowledge Graph Features

Medical domain knowledge could be helpful in increasing the likelihood of symptoms when related medications is mentioned in a conversation, and vice versa. One such source is a knowledge graph (KG) whose embedding represent a low-dimensional projection that captures structural and semantic information of its nodes. Previous work has demonstrated that KG embedding can improve relation extraction in written domain (Han et al., 2018). We utilize an internally developed KG that contains about 14k medical nodes of 87 different types (e.g., medications, symptoms, treatments, etc.). The nodes are represented by 256 dimension embedding vectors, which were trained

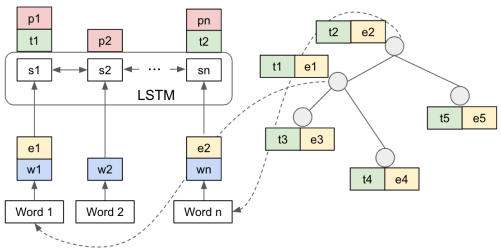


Figure 2: Illustration of how POS (p) features and knowledge graph (KG) are incorporated into our encoder. Dashed lines represent mappings from words (w) to KG nodes, which contain the embedding (e) and the type (t) information.

to minimize word2vec loss function on web documents (Mikolov et al., 2013). A given node may belong to multiple types and this is encoded as a sum of one-hot vectors. The input word sequences were mapped to KG nodes using an internal tool (Brown, 2013). For words that do not map to KG nodes, we use a learnable UNK vector of the same dimension as the KG embedding. In addition, we also represented linguistic information using part-of-speech (POS) tags as one-hot vector. The POS tags were inferred from the input sequence using an internal tool with 47 distinct tags (Andor et al., 2016). In our experiments, we find it most effective to concatenate word embedding with KG entities, and the encoder output with the embedding of POS tags and the KG entity types.

6 Experiments

We describe our corpus, evaluation metrics, the experimental setup, the evaluations of the proposed model and comparison with different baselines on both the symptom and medication tasks.

6.1 Corpus

Given the privacy-sensitive nature of clinical conversations, there aren't any publicly available corpora for this domain. Therefore, we utilize a private corpus consisting of 92K de-identified and manually transcribed audio recordings of clinical conversations, typically about 10 minutes long [IQR: 5-12 minutes] with mostly 2 participants (72.7%). Other participants when present included, for example, nurses and caregivers. The corresponding manual transcripts contained sequences that were on average 208 utterances or 1,459 words in length. We note that due to the casual conversational style of speech, an entity mentioned at the beginning can be related to a property mentioned at the end of the conversation. This

makes the problem of modeling relations much harder than previous work on extracting relations.

A subset of about 2,950 clinical conversations, related to primary care, were annotated by professional medical scribes. The ontology for labeling medication consisted of the type of medications (e.g., medications, supplements) and their properties (e.g., dosage, frequency, duration), and that for symptoms consisted of 186 symptom names and their properties. This resulted in 77K and 99K tags for the medication and symptom tasks, respectively. In all, there were 23k and 16k relationships between medications and symptoms and their properties, respectively. The conversations were divided into training (1,950), development (500) and test (500) sets.

In the case of medications, about 70% of the labels were about medications and the rest about their properties, of which 51% were dosage or quantity, and 40% were frequency. In the case of symptoms, 41% of the labels were about symptom names, another 41% about status, and the rest about properties, of which 39% were about frequency and 37% about body locations.

6.2 Pretraining

Since our labeled data is small, only about 3k, the input encoder of the model was pre-trained over the entire 92k conversations. For pre-training, given a short snippet of conversation, the model was tasked with predicting the next turn, similar to skip-thought (Kiros et al., 2015). Our models were trained using the Adam optimizer (Kingma and Ba, 2015) and the hyperparameters are described in the supplementary material.

6.3 Evaluation Metrics

As described in Section 2, our tasks consist of extracting tuples – (*symType, propType, propContent*) for symptoms task and (*medContent, propType, propContent*) for medications. The precision, recall and F1-scores are computed jointly over all the three elements and the content is treated as a list of tokens for evaluation purposes. To allow for partial content matches, we generalize the calculation of precision and recall such that

$$Precision = \frac{1}{|\mathcal{S}_{\hat{Y}}|} \sum_{i \in \mathcal{S}_{\hat{Y}}} \prod_{j=1}^{3} \mathcal{I}_{\hat{y}_i^j}(\hat{y}_i^j, y_i^j)$$

$$Recall = \frac{1}{|\mathcal{S}_{Y}|} \sum_{i \in \mathcal{S}_{Y}} \prod_{j=1}^{3} \mathcal{I}_{y_i^j}(\hat{y}_i^j, y_i^j)$$

where $\mathcal{S}_{\hat{Y}}$ denotes the set of predictions, $\mathcal{S}_Y$ denotes the set of ground truths, and $\mathcal{I}_{z_i^j}(\hat{x}_i^j, x_i^j) = |\hat{x}_i^j \cap x_i^j| / |z_i^j|$. We note that, as *symType* and *propType* are simply target classes, $\mathcal{I}$ reduces to a simple indicator function. Under the scenario that the content includes single elements, the entire calculation simplifies to the exact matching-based calculation of precision and recall over the set of predictions and ground truths. For the symptom task, we additionally evaluate the performance of predicting *symType* and *symStatus* by performing the exact matching-based calculation.

We illustrate this evaluation metric with an example below: There are two symptoms in the reference and the model extracted one of them. In the extracted symptom, the model correctly identified one out of the two content words. So, we score the precision as $1/1(1 * 1 * (1/1)) = 1$ and recall as $1/2((0 * 0 * 0) + (1 * 1 * (1/2))) = 0.25$.

Prediction: [(sym/sob, prop/severity, [bad])]
Reference: [(sym/unk, prop/location, [arm]),
(sym/sob, prop/severity, [really, bad])]

6.4 Baselines

Symptom Task As a baseline for this task, we train an extension of the standard tagging model, described in Section 4.1. The label space for extracting the relations between symptoms and their properties is *186 symptoms* $\times$ *3 properties*, and for extracting symptoms and their status is *186 symptoms* $\times$ *3 status*. Using the BIO-scheme, that adds up to 2,233 labels in total. The baseline consists of a bidirectional LSTM-encoder followed by two feed-forward layers $[512, 256]$ and then a 2,233 dimension softmax layer. The label space is too large to include a CRF layer. The encoder was pre-trained in the same way as described in Section 6.2, the hyperparameters were selected according to Table 2, and the model parameters were trained using cross-entropy loss.

Medication Task For this task, we adopt a different baseline since the generic medication entity type (e.g., drug name, supplement name) does not provide any useful information unlike the 186 symptom entity labels (e.g., sym/msk/pain). Instead, we adopt the neural co-reference resolution approach which is better suited to this task (Lee et al., 2017). The encoder is the same as the baseline for symptom task and pre-trained in the same manner. Since the BIO labels contain only 9 elements in this case, the encoder output is fed into a CRF layer. Each candidate relation is represented by concatenating the latent states of the head tokens of the medication entity and the property. This representation is augmented with an embedding of the token-distance, which is fed to a softmax layer whose binary output encodes whether they are related or not. Note our R-SAT model does not take the advantage of this distance embedding.

6.5 Parameter Tuning

Table 2 shows the parameters that were selected after evaluating over a range on a development set. In all experiments, the $\mathrm{Aggregate}(\cdot)$ function is implemented as the mean function for its simplicity.

Parameter	Used	Range
Word emb	256	[128 – 512]
LSTM Cell	1024	[256 – 1024]
Enc/dec layers	1	[1 – 3]
Dropout	0.4	[0.0 – 0.5]
L2	1e-4	[1e-5 – 1e-2]
Std of VN	1e-3	[1e-4 – 0.2]
α	0.01	[1e-4 – 0.1]
Learning rate	1e-2	[1e-4 – 1e-1]

Table 2: Hyperparameters of our models for model reproducibility.

6.6 Results & Ablation Analysis

The performance of the proposed R-SAT model was compared with the baseline models, and the results are reported in Table 3.

Symptom Task The model was trained using multi-task learning for both tasks: (*symType, propType, propContent*) as well as (*symType, symStatus*). The performance was evaluated using all the elements of the tuple as described in Section 6.3. The baseline performs better on (*symType, symStatus*) compared to (*symType, propType, propContent*) possibly because there are more instances of the former in the training data than the latter. The R-SAT model performs significantly better than baselines on both tasks.

For understanding the contribution of different components of the model, we performed a series of ablation analysis by removing them one at a time. In extracting relations in *Sx + Property*, the KG embeddings along with POS tags contribute a relative gain of **13%** while the memory buffer brings a relative gain of **8%**. Neither of them impact *Sx + status*, and that is expected for memory buffer since the status is tagged on the same span as the contents of the memory buffer. Multi-task learning brings a relative improvement of **4%** on *Sx + Property*, and this may be because there are fewer instances of this relation in the training data, and jointly learning with *Sx + Status* helps to learn better representations. Note we have not checked other sequences for removing model components (e.g., removing Multi-tasking earlier or KG later).

Medication Task In the Rx case, we only have one task (*Rx + Property*), that is, predicting the relations between medications and their properties, e.g., ([ibuprofen], prop/dosage, [10 mg]). The baseline gives reasonable performance. Ablation analysis reveals that KG and POS features contribute about **4.6%** relative improvement, while the contextual span in memory buffer adds a substantial **43%** relative improvement. Since the medications are from an open set, we cannot run experiments without the buffer. Compared to symptoms task, the model performs better on medication task, and this may be due to lower variability in dosage.

Relation Only Prediction For teasing apart the strength and weakness of the model, we evaluated its performance when the entities and their properties were given, and the model was only required to decide whether a relation exists or not.

As a baseline, we compare our model with a most recently proposed model for document-level joint entity and relation extraction: BRAN, which achieved state-of-art performance for chemical-disease relation (Verga et al., 2018). When this model was originally used to test relations between all pairs of entities and properties in the entire conversation, it performed relatively poorly. Using the implementation released by the authors, the performance of BRAN was then optimized by restricting the distance between the pairs and by fine-tuning the threshold. The best results are re-

Model	Sx + Property	Sx + Status	Rx + Property
Baseline	0.18	0.44	0.35
R-SAT	**0.34**	**0.57**	**0.45**
w/o [KG]	0.30	0.56	0.43
w/o [KG, Context]	0.26	0.55	0.30
w/o [KG, Context, Buffer]	0.24	0.55	n/a
w/o [KG, Context, Buffer, Multi-task]	0.23	n/a	n/a
Human	0.51	0.78	0.52

Table 3: Comparison of the performance of the proposed R-SAT model with baselines and ablation analysis on different components (KG, Context, Buffer, Multi-task) where 'context' is the latent representation of the span $\boldsymbol{h}_{ij}^{s}$ in the memory buffer.

Model	*Sx + Property*	*Rx + Property*
BRAN	0.62	0.41
R-SAT	0.82	0.60

Table 4: Performance of the model when the entities and properties are given and it is only required to predict existence of relations.

ported in Table 4. Our proposed R-SAT model without any such constraints performs better than BRAN on both tasks by an absolute F1-score gain of about 0.20.

Interestingly, the performance of our model on *Sx + Property* jumps from 0.34 in the joint prediction task to 0.82 in the relation only prediction task. This reveals the primary weakness of the Sx model is in tagging the entities and the properties accurately. In contrast, the F1-score for *Rx + Property* is impacted less, and only moves up from 0.45 to 0.6.

The task of inferring whether a relation is present between a medication and its properties is more challenging than in the case of symptoms task. This is not entirely surprising since there is a higher correlation between symptom type and location (e.g., respiratory symptom being associated with nose) and relatively low correlation between dosage and medications (e.g., 400mg could be the dosage for several different medications).

6.7 Analysis

For understanding the inherent difficulty of extracting symptoms and medications and their properties from clinical conversations, we estimated human performance. A set of 500 conversations were annotated by 3 different scribes. We created a "voted" reference and compared the 3 annotations from each of the 3 scribes against them.

The F1-score of scribes were surprisingly low, with 0.51 for *Sx + Property* and 0.78 for *Sx + Status*. The model performance also finds extracting relation in *Sx + Property* to be more difficult than *Sx + Status* task. In summary, the model performance reaches 67% of human performance for *Sx + Property* and 73% for *Sx + Status*. The F1-score of scribes for *Rx + Property* is similar to that of *Sx + Property*. In this case, the model achieves about 85% of human performance. The human errors or inconsistencies in Sx and Rx annotations appear to be largely due to missed labels and not due to inconsistent spans for the same tags, or inconsistent tags for the same span.

While the majority of our relations in the reference annotations occurred within the same sentence, approximately 11.1% of relations occurred across 3 or more sentences. This typically occurred when the symptoms or medications are discussed over multiple dialog turns, as illustrated in Table 1. Among the relations correctly identified by the model, 10.6% were also across 3 or more sentences, which is very similar to the priors on the reference and seem to contain no bias. We notice that in certain cases, the model is able to link a property to an entity that is far away (100+ sentences) when a nearby mention of the same entity was missed by the model. Models that only examine relations in nearby sentences (2-3 sentences) would have missed the relation in such a scenario.

The majority of the errors result from our model missing the property span. Specifically, we see that 35% and 81% of the errors are due to model not detecting medications and symptoms property. For example, when i really have to, every three three months, which are rare mentions in informal language.

Our reference links each property to only one

entity. In certain cases, we notice that the model links the entity to an alternative mention or entity that is equally valid (Advil vs pain killer). So, our performance measure underestimates the actual model performance.

7 Conclusions

We propose a novel model to jointly infer entities and relations. The key components of the model are: a mechanism to highlight the spans of interest, classify them into entities, store the entities of interest in a memory buffer along with the latent representation of the context, and then infer relation between candidate property spans with the entities in the buffer. The components of the model are not tied to any domain. We have demonstrated applications in two different tasks. In the case of symptoms, the entities are categorized into 188 classes, while in the case of medications, the entities are an open set. The model is tailored for tasks where the training data is limited and the label space is large but can be partitioned into subsets. The two stage processing where the candidates are stored in a memory buffer allows us to perform the joint inference at a computational cost of $\mathcal{O}(n)$ in the length of the input n compared to methods that explore all spans of entities and properties at a computational cost of $\mathcal{O}(n^4)$. The model is trained end-to-end. We evaluate the performance on three related tasks, namely, extracting symptoms and their status, relations between symptoms and their properties, and relations between medications and their properties. Our model outperforms the baselines substantially, by about 32-50%. Through ablation analysis, we observe that the memory buffer and the KG features contribute significantly to this performance gain. By comparing human scribes against "voted" reference, we see that the task is inherently difficult, and the models achieve about 67-85% of human performance.

Acknowledgments

This work would not have been possible without the help of a number of colleagues, including Laurent El Shafey, Hagen Soltau, Yuhui Chen, Ashley Robson Domin, Lauren Keyes, Rayman Huang, Justin Stuart Paul, Mark Knichel, Jeff Carlson, Zoe Kendall, Mayank Mohta, Roberto Santana, Katherine Chou, Chris Co, Claire Cui, and Kyle Scholz.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the ACM Conference on Digital Libraries*, pages 85–94. ACM.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Brian G. Arndt, John W. Beasley, Michelle D. Watkinson, Jonathan L. Temte, Wen-Jan Tuan, Christine A. Sinsky, and Valerie J. Gilchrist. 2017. Tethered to the EHR: primary care physician workload assessment using EHR event log data and time-motion observations. *Annals of Family Medicine*, 15(5):419–26.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1171–1179, Cambridge, MA, USA. MIT Press.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database issue):D267–70.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *In WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT98*, pages 172–183.

Aaron Brown. 2013. Semantics and structured knowledge in practice at google. In *IEEE ICSC*. IEEE International conference on semantic computing.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

Soravit Changpinyo, Hexiang Hu, and Fei Sha. 2018. Multi-task learning for sequence tagging: An empirical study. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2965–2977. Association for Computational Linguistics.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *ACL*.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Kevin Clark and Christopher D Manning. 2016. Deep reinforcement learning for Mention-Ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program tasks, data, and evaluation. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*.

Nan Du, Kai Chen, Anjuli Kannan, Linh Tran, Yuhui Chen, and Izhak Shafran. 2019. Extracting symptoms and their status from clinical conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 915–925, Florence, Italy. Association for Computational Linguistics.

Greg P. Finley, Erik Edwards, Amanda Robinson, Najmeh Sadoughi, Mark Miller, David Suendermann-Oeft, and Nico Axtmann Michael Brenndoerfer. 2018a. An automated medical scribe for documenting clinical encounters. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Gregory Finley, Wael Salloum, Najmeh Sadoughi, Erik Edwards, Amanda Robinson, Nico Axtmann, Michael Brenndoerfer, Mark Miller, and David Suendermann-Oeft. 2018b. From dictations to clinical reports using machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 121–128. Association for Computational Linguistics.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. RelEx–relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, Bernt Andrassy, and Thomas A. Runkler. 2019. Neural relation extraction within and across sentence boundaries. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 33–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2016. Investigating LSTMs for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929, Berlin, Germany. Association for Computational Linguistics.

J D Kim, Y Wang, and Y Yasunori. 2013. The Genia event extraction shared task. *Proceedings of the BioNLP*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28*, pages 3294–3302.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *ACL*.

ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. 2013. Convolution neural network for relation extraction. In *International Conference on Advanced Data Mining and Applications*, pages 231–242. Springer.

Feifan Liu, Chunhua Weng, and Hong Yu. 2012. *Natural Language Processing, Electronic Health Records, and Clinical Research*, pages 293–310. Springer Science & Business Media.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074. Association for Computational Linguistics.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3(0):405–418.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Raymond J. Mooney and Razvan C. Bunescu. 2006. Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 171–178. MIT Press.

Pinal Patel, Disha Davey, Vishal Panchal, and Parth Pathak. 2018. Annotation of a large clinical entity corpus. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2033–2042. Association for Computational Linguistics.

Yifan Peng, Manabu Torii, Cathy H Wu, and K Vijay-Shanker. 2014. A generalizable NLP framework for fast development of pattern-based biomedical relation extraction systems. *BMC Bioinformatics*, 15(1):285.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765. Association for Computational Linguistics.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 697–704, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems*, pages 1297–1304.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011a. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J. Am. Med. Inform. Assoc.*, 18(5):552–556.

Ozlem Uzuner, Brett R South, Shuying Shen, and Scott L. DuVall. 2011b. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of American Medical Informatics Association*, 18(5):552–6.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 872–884. Association for Computational Linguistics.

Robert Wachter and Jeff Goldsmith. 2018. To combat physician burnout and improve care, fix the electronic health record. *Harvard Business Review*.

Rena Xu. 2018. The burnout crisis in american medicine. *The Atlantic*.

Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *ICLR)*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark. Association for Computational Linguistics.

Xinsong Zhang, Pengshuai Li, Weijia Jia, and Hai Zhao. 2019. Multi-labeled relation extraction with attentive capsule network. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.