Relation Extraction

Justin T. Chiu

July 12, 2019

Abstract

Recent relation extraction systems predict the relationship between an entity and value given the positions of their mentions in the text. This requires words to be annotated as mentions. The cost of obtaining human annotations for each word scales linearly with the size of the text. Automatic annotation methods allow the annotation process to scale sublinearly in human effort, but may introduce noise due to incorrect annotations. In order to train a probabilistic information extraction model without mention annotations, we specify a model that, identifies important words and uses them to explain a triple from a knowledge base.

1 Problem Statement

Relation extraction aims to extract facts from a passage of text. Extraction systems convert facts expressed in natural language into a form amenable to computation. Facts consist of three components: Entities, relation types, and values. The challenge is to not only extract facts from text, but also justify the extractions by determining where those facts are mentioned.

A *mention* is a surface realization of an abstract object in text. In relation extraction we justify extractions by identifying fact mentions. As text is noisy, the realization of a fact may be difficult to locate. We focus on locating fact mentions at the word level by identifying individual words as value mentions, rather than entity or type mentions.

The problem description is as follows, focusing on the domain of basketball summaries: Given a written summary of a basketball game $x = x_1, \ldots, x_I$ of length I, we model the aligned box score $\{(e_j, t_j, v_j)\}_{J=1}^{J}$ consisting of entities e_j, relation types t_j, and all values $v_j \in \mathcal{V}$. The set of facts is our knowledge base (KB), which contains J facts.

Let $D = \{(e_j, t_j)\}_{j=1}^{J}$ and $v = \{v_j\}_{j=1}^{J}$. The KB (D, v) can be viewed as a data table where d defines a flattened representation of the rows and columns, the *cells* of the table, and v gives the *values* of the cells. Our goal is to locate and extract facts from x.

Modeling only the KB (d, v) given the text x is not sufficient, as our goal is to locate fact mentions. In fact, we assume the KB contains many more facts than those mentioned in the text. This fits many scenarios in real world applications: we may have many entity and type pairs in our data table, but a summary may discuss only a small, salient subset of players and statistics. We therefore propose a model that first identifies words as value mentions, aligns those mentions to an entity and relation type in order to obtain a fact, and then aggregates word level decisions to resolve conflicts.

2 Model

We define a graphical model that performs extraction with justification.

2.1 Notation

Consider the following example: Let our KB consist of the data table and values

$$D = \left\{ \begin{array}{c} (e_1 = \text{John Doe}, t_1 = \text{Points}), \\ (e_2 = \text{John Doe}, t_2 = \text{Rebounds}), \\ (e_3 = \text{John Doe}, t_3 = \text{First name}), \\ (e_4 = \text{John Doe}, t_4 = \text{Last name}), \\ \vdots \end{array} \right\}, v = \left\{ \begin{array}{c} v_1 = 8, \\ v_2 = 12, \\ v_3 = \text{John}, \\ v_4 = \text{Doe}, \\ \vdots \end{array} \right\}$$

aligned to the summary $x =$ John Doe scored eight points. We introduce the following latent variables:

- $m = m_1, \ldots, m_I$ where each $m_i \in \{0, 1\}$ indicates whether word x_i is part of a value mention. In our example, we have $m_4 = 1$ as $x_4 =$ eight mentions the value $v_1 = 8$. Individual words may be part of a multiword value mention. We aim to identify at least one word in a multiword mention.

- $a = a_1, \ldots, a_I$ where each $a_i \in \{1, \ldots, J\}$ gives the index of the cell of the data table D_{a_i} whose value v_{a_i} is mentioned at index i. Again examining $x_4 =$ eight, we would have $a_4 = 1$, as x_4 is aligned to the the cell D_1 whose value $v_1 = 8$ is mentioned.

- $z = z_1, \ldots, z_I$ where each $z_i \in \mathcal{V}$ translates a value mention into a value. In the case of $x_4 =$ eight, the mention must be translated to the value $z_4 = 8$. This is the canonical representation of a value as determined by the schema of the KB. For example, the KB may only store numbers in numerical form, as opposed to alphabetical.

2.2 Model

We proceed to detail the model in two stages. We first give the word-level process, which makes a series of choices for each word x_i in the text $x = x_1, \ldots, x_I$, then the sequence-level process which aggregates all of the word-level decisions.

The word-level process has three steps. For each index in time $i \in 1, \ldots, I$ we perform

1. Value mention identification: Given a sequence of words x, we independently choose whether each word is a value mention with
$$p(m \mid x) = \prod_i p(m_i \mid x) \tag{1}$$

2. Alignment: Each word x_i is independently aligned to a record in the knowledge base with
$$p(a \mid x, D) = \prod_i p(a_i \mid x, D) \tag{2}$$

We align the word x_i by choosing the cell whose value generates the possible value mention at index i. In particular, $a_i = j$ denotes the alignment to the cell d_j. We assume that each value mention aligns to a single record. Although we align every word to a cell, words that are not chosen to be mentions, i.e. x_i such that $m_i = 0$, will be ignored.

3. Translation: All words are translated into a value from the KB schema with

$$p(z \mid x) = \prod_i p(z_i \mid x) \tag{3}$$

with $z_i \in \mathcal{V}$. We would like this to capture synonyms of values not captured by the limited schema of the KB. Although every word is translated to a value, we ignore those that are not chosen as value mentions.

Finally, we aggregate the word-level choices at the sequence level in order to make a single choices for the values v given the summary x.

4. Aggregation: Given the word-level values z, alignments a, mentions m, and data table D we choose the sequence-level values v_j independently from

$$p(v \mid z, a, m, d) = \prod_j p(v_j \mid z, a, m, D_j) \tag{4}$$

We assume each of the word-level choices in each of z, a, m are made independently given the text x and table cells D. This gives us the following factorization of the relation extraction system:

$$\begin{aligned} p(v, z, a, m \mid x, D) &= p(v \mid z, a, m, x, d)p(z \mid x)p(a \mid x, D)p(m \mid x) \\ &= \left(\prod_{j=1}^{J} p(v_j \mid z, a, m, x, D)\right)\left(\prod_{i=1}^{I} p(z_i \mid x)p(a_i \mid x, D)p(m_i \mid x)\right) \end{aligned} \tag{5}$$

2.3 Parameterization

We parameterize the conditional distributions for mention identification, alignment, translation, and aggregation below.

Let $\mathbf{h}_i \in \mathbb{R}^d$ be a contextual embedding of the word x_i, and E an embedding function that maps the cells of the data table D to vectors in $\mathbb{R}^d$.

1. Identification: We use the contextual embedding to directly predict whether a word is part of a value mention.

$$p(m_i \mid x) \propto \exp(W_m \mathbf{h}_i), W_m \in \mathbb{R}^{2\times d}$$

2. Alignment: We use a bilinear function of the cell embeddings and contextual embeddings to parameterize the alignment distribution.

$$p(a_i \mid x, d) \propto \exp(E(D_{a_i})^T W_d \mathbf{h}_i)$$

with $W_d \in \mathbb{R}^{d\times d}$.

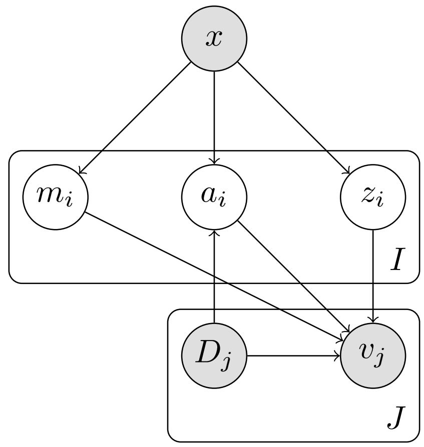

Figure 1: Our model predicts word-level values and alignments then aggregates those choices over all indices i to make a single decision for each value. Each word has the following latent variables: the mention $m_i \in \{0, 1\}$ indicates whether word x_i is a value mention, the alignment a_i gives the cell D_{a_i} that x_i aligns to, and the value z_i gives the canonical value that x_i translates to.

3. Translation: We use the contextual embedding to translate a word into a value.

$$p(z_i \mid x) \propto \exp(W_z \mathbf{h}_i), W_z \in \mathbb{R}^{|\mathcal{V}| \times d}$$

4. Aggregation: If there exists an index that is a mention and is also aligned to r_j we allow it to vote on the value v_j, otherwise we ignore the text and use a prior distribution over values $p(v_j \mid d_j) \propto \exp(E(v_j)^T W_v[E(D_j)])$.

$$p(v_j \mid z, a, m, d) \propto \begin{cases} \prod \exp(\psi(v_j, z_i, a_i, m_i, d)), & \exists i, m_i = 1 \wedge a_i = j \\ \exp(E(v_j)^T W_v[E(d_j)]), & \text{otherwise} \end{cases}$$
$$\psi(v_j, z_i, a_i, m_i, d) = 1(v_j = z_i, a_i = j, m_i = 1)$$

3 Training and Inference

To train a latent variable model, we must marginalize over the unobserved RVs and maximize the likelihood of the observed. Ideally, we would optimize the following objective

$$\log p(v \mid x, d) = \log \sum_{z,a,m} p(v, z, a, m \mid x, d) \tag{6}$$

However, maximizing $\log p(v \mid x, d)$ directly is very expensive for this model as the summation over z, a, m is intractable. The summation over z, a, m has computational complexity $O((|\mathcal{V}| \cdot J \cdot 2)^I)$.

We therefore resort to approximate inference, specifically amortized variational inference.

3.1 Inference network

We introduce an inference network $q(z, a, m \mid v, x, d)$ and optimize the following lower bound on the marginal likelihood with respect to the parameters of both p and q:

$$\log p(v \mid x) \geq \mathbb{E}_{q(z,a,m|v,x,d)} \left[\log \frac{p(v, z, a, m \mid x, d)}{q(z, a, m \mid v, x, d)} \right] \tag{7}$$

We propose to parameterize $q(z, a, m \mid v, x, d)$ as follows. We decompose

$$\begin{aligned} q(z, a, m \mid v, x, d) &= q(z \mid a, v, x) q(a \mid v, x, d) q(m \mid v, x) \\ &= \prod_i q(z_i \mid a, v, x) q(a_i \mid v, x, d) q(m_i \mid v, x) \end{aligned} \tag{8}$$

The conditional distributions of our inference network are very similar to the relation extraction model, but they condition on the values v.

Let $\mathbf{h}_i \in \mathbb{R}^d$ be a contextual embedding of the word x_i. We use attention weights over records to get a weighted representation of the records of the KB for each index i:

$$\begin{aligned} \mathbf{g}_{r_j} &= [E(d_j), E(v_j)] \\ \alpha_j &\propto \exp(\mathbf{g}_{r_j}^T W_\alpha \mathbf{h}_i) \end{aligned}$$

The inference network is given by

1. The value mention model $q(m_i \mid v, x)$ has access to the values v from the KB, which it conditions on when detecting value mentions.

$$p(m_i \mid v, x) = W'_m \text{MLP}([\sum_j \alpha_j \cdot \mathbf{g}_{r_j}, \mathbf{h}_i]), W'_m \in \mathbb{R}^{2 \times d}$$

2. The alignment model $q(a_i \mid v, x, d)$ uses the attention weights to parameterize the alignment $p(a_i \mid x) = \alpha_{a_i}$.

3. The translation model $q(z_i \mid a, v, x) = 1(z_i = v_{a_i})$ conditions on the alignments a and ensures the chosen z is consistent with the alignments.

We use the score function gradient estimator to perform gradient ascent on the objective in Equation 8 with respect to both p and q. We utilize a leave-one-out baseline for variance reduction.

One concern is that the model may learn to never rely on the text for extraction, setting $m_i = 0$ at every index. We can avoid this by initializing $q(z)$ to ensure that for words x_i where there exists a $z_i \in \mathcal{V}$ such that x_i and z_i are a lexical match we assign high probability to the transliteration $q(z_i = x_i)$ via pretraining.

4 Evaluation

As we assumed that the KB contained a superset of the facts contained in a sequence of text, we are interested in evaluating whether the model can discover and locate the subset of facts that are expressed in the text.

We evaluate by determining whether the model can identify which facts in a KB are expressed in text. Given a ground truth set of facts extracted by a human annotator, we compute the precision and recall of the extractions by the model. We perform extraction by finding $\arg\max_{z_i,a_i,m_i} q(z_i), q(a_i), q(m_i)$ for each word x_i.

References