Relation Extraction

Justin T. Chiu

August 19, 2019

Abstract

TODO

1 Problem Statement

Relation extraction is the task of jointly locating and converting information conveyed in natural language into a structured representation. By removing the noise inherent in natural language, extraction systems produce information in a representation that is amenable to computation. We focus on representations that can be decomposed into keys and values.

We produce an extraction system that justifies its decisions. We use these justifications or 'rationales' to bias the model away from relying on spurious correlations that may hurt generalization.

However, deciding what constitutes a rationale and obtaining supervision for that definition is expensive. For every corresponding definition of a rationale, supervision must be obtained by having annotators consider many examples. This prevents one from iterating quickly in the space of rationale definitions. We therefore limit our attention to models that are able to produce rationales without supervision, or with very limited supervision.

The problem description is as follows: Given a textual summary $x = x_1, \dots, x_I$ of length I, we model the aligned knowledge base (KB) $\{(k_j, v_j)\}_{j=1}^J$ consisting of entity e_j and type t_j tuples which we refer to as keys $k = \{(e_j, t_j)\}_{j=1}^J$, and values $v = \{v_j\}_{j=1}^J, v_j \in \mathcal{V}$. The KB (k, v) contains J facts.

Our goal is to locate and extract facts from x by modeling the values v given the text x and the keys k.

Consider the following example: Let our KB consist of information about a basketball game,

$$k = \left\{ \begin{array}{c} (\text{John Doe}, \text{POINTS}), \\ (\text{John Doe}, \text{REBOUNDS}), \\ (\text{John Doe}, \text{FIRST\_NAME}), \\ (\text{John Doe}, \text{LAST\_NAME}), \\ \vdots \end{array} \right\}, v = \left\{ \begin{array}{c} 8, \\ 12, \\ \texttt{`John'}, \\ \texttt{`Doe'}, \\ \vdots \end{array} \right\}$$

aligned to the summary $x =$ `'John Doe scored eight points'`. As an example extraction, we identify the value mention `'eight'` as the trigger for the fact (k_1, v_1). The fact (k_2, v_2) is not expressed in the text.

1.1 Model

In order to perform extraction with justification 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. We aim to identify at least one word in a multiword mention.
- $a = a_1, \ldots, a_I$ where each $a_i \in \{1, \ldots, J\} \cup \{\texttt{None}\}$ indicates that word x_i mentions the fact (k_{a_i}, v_{a_i}).
- $z = z_1, \ldots, z_I$ where each $z_i \in \mathcal{V} \cup \{\texttt{None}\}$ translates the value mention containing word x_i into the canonical representation of the value as determined by the schema of the KB. For example, the KB may only store numbers in numerical form, as opposed to alphabetical.

Our model specifies the joint distribution over the values v and latent variables z, a, m given the text x and keys k:

$$\begin{aligned}
&p(v, z, a, m \mid x, k)\\
&= p(v \mid z, a, m, x, k) p(z, a, m \mid x, k)\\
&= \left(\prod_{j=1}^{J} \underbrace{p(v_j \mid z, a, m, x, k)}_{\text{Aggregation}}\right) \underbrace{\left(\prod_{i=1}^{I} \underbrace{p(z_i \mid m_i, x)}_{\text{Translation}} \underbrace{p(a_i \mid m_i, x, k)}_{\text{Alignment}} \underbrace{p(m_i \mid x)}_{\text{Identification}}\right)}_{\text{Extraction}}
\end{aligned} \tag{1}$$

The extraction model has three components. The model must identify whether word x_i is a mention using the *identification* model $p(m_i \mid x)$, align word x_i to a key k_{a_i} via the *alignment* model $p(a_i \mid m_i, x, k)$, and translate the word x_i into a value z_i with the *translation* model $p(z_i \mid m_i, x)$.

This model assumes that each word x_i can explain at most one fact in the KB. Additionally, we make the simplifying assumption that a value mention's translation is unambiguous and independent from the alignment to a key.

It is possible for there to be conflicts in the word-level extraction model if two words are mentions with the same alignment but different values. The aggregation model $p(v_j \mid z, a, m, x, k)$ is necessary to resolve these conflicts, and uses the per-word latent variables z_i, a_i, m_i to predict the values of the KB.

We give the parameterizations of all conditional distributions in the following section.

1.2 Parameterization

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

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

$$p(m_i \mid x) \propto \exp(W_m \mathbf{h}_i)_{m_i}, \tag{2}$$

with $W_m \in \mathbb{R}^{2 \times d}$.

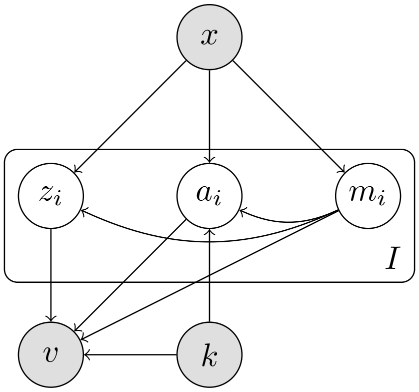

Figure 1: Our model predicts word-level values and alignments then aggregates those choices over all indices i to make a single decision for each value. Each word has the following latent variables: the mention $m_i \in \{0, 1\}$ indicates whether word x_i is a value mention, the alignment a_i gives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

2. Alignment: We use a bilinear function of the cell embeddings and contextual embeddings to parameterize the alignment distribution. We align $a_i = \texttt{None}$ if a word is not a mention.

$$\begin{aligned} p(a_i \mid m_i = 1, x, k) &\propto \exp(E(k_{a_i})^T W_a \mathbf{h}_i) \\ p(a_i = \texttt{None} \mid m_i = 0, x, k) &= 1 \end{aligned} \tag{3}$$

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

3. Translation: We use the contextual embedding to translate a word into a value. If a word is chosen not to be a mention such that $m_i = 0$, we set $z_i = \texttt{None}$.

$$\begin{aligned} p(z_i \mid m_i = 1, x) &\propto \exp(W_z \mathbf{h}_i)_{z_i} \\ p(z_i = \texttt{None} \mid m_i = 0, x) &= 1 \end{aligned} \tag{4}$$

with $W_z \in \mathbb{R}^{|\mathcal{V}| \times d}$.

4. Aggregation: We choose the value associated with a particular key in the KB by first using the translated values of any mentions aligned to the corresponding key. If no mention is aligned to the key, we choose the value given only the key.

Formally, the conditional distribution is given by

$$p(v_j \mid z, a, m, k) \propto \begin{cases} \prod_i \exp(\psi(v_j, z_i, a_i, m_i)), & \exists i, m_i = 1 \wedge a_i = j \\ \exp(E(v_j)^T W_v [E(k_j)]), & \text{otherwise} \end{cases} \tag{5}$$

$$\psi(v_j, z_i, a_i, m_i) = \mathbf{1}(v_j = z_i, a_i = j, m_i = 1) \tag{6}$$

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

2 Training and Inference

We would like to optimize the marginal likelihood of the values v given the text x and keys k:

$$\log p(v \mid x, k) = \log \sum_{z,a,m} p(v, z, a, m \mid x, k) \tag{7}$$

However, maximizing $\log p(v \mid x, k)$ directly is very expensive as the summation over variables z, a, m is intractable; in particular, the summation has computational complexity $O((|\mathcal{V}| \cdot J)^I)$. We therefore resort to approximate inference, specifically amortized variational inference.

2.1 Inference Network

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

$$\log p(v \mid x, k) \geq \mathbb{E}_{q(z,a,m|v,x,k)} \left[\log \frac{p(v, z, a, m \mid x, k)}{q(z, a, m \mid v, x, k)} \right] \tag{8}$$

The joint distribution of $q(z, a, m \mid v, x, k)$ in the inference network is given by

$$q(z, a, m \mid v, x, k) = \prod_i \underbrace{q(z_i \mid a, v, x)}_{\text{Translation}} \underbrace{q(a_i \mid v, x, k)}_{\text{Alignment}} \underbrace{q(m_i \mid v, x)}_{\text{Identification}} \tag{9}$$

In the inference network, we use the contextual embedding $\mathbf{h}_i \in \mathbb{R}^d$ of the word x_i and extend the embedding function E to embed not only keys but also values in $\mathbb{R}^d$. We introduce the following attention weights over records in order to get a weighted representation of the KB for each word x_i:

$$\begin{aligned} \mathbf{g}_j &= [E(k_j); E(v_j)] \\ \alpha_{ij} &\propto \exp(\mathbf{g}_j^T W_\alpha \mathbf{h}_i) \end{aligned} \tag{10}$$

with $W_\alpha \in \mathbb{R}^{2d \times d}$.

We propose to parameterize the inference network $q(z, a, m \mid v, x, k)$ as follows:

1. Identification: To predict whether x_i is a mention, the inference network uses the weighted representation of the KB α along with the contextual representation $\mathbf{h}_i$

   $$p(m_i \mid v, x) \propto \exp\left(V_m \text{MLP} \left(\left[\sum_j \alpha_{ij} \cdot \mathbf{g}_j; \mathbf{h}_i \right] \right) \right)_{m_i} \tag{11}$$

   where MLP is a neural network and $V_m \in \mathbb{R}^{2 \times d}$.

2. Alignment: The alignment model uses the attention weights in Eqn. 10: $q(a_i = j \mid v, x, k) = \alpha_{ij}$.

3. Translation: The translation model $q(z_i \mid a_i, v, x) = \mathbf{1}(z_i = v_{a_i})$ conditions on the alignment a_i and ensures the translated value z_i is consistent with the alignment.

We optimize the objective in Eqn. 9 with respect to both the extraction model p and the inference network q via gradient ascent using the score function gradient estimator. We utilize a leave-one-out baseline for variance reduction.

(Under consideration, could hopefully remove pretraining by parameterizing translation model using local features such as embedding or conv, but previous experiments on this front were negative) As our model is highly unidentifiable, we bias our extraction model by pretraining $p(z_i \mid x)$ on lexical match.

3 Evaluation

We evaluate whether the model can discover and locate the subset of facts in a KB that are expressed in the text without observing the values of the KB. We compare the extractions of our model with a ground truth set of facts, reporting the micro-averaged precision, recall, and F1 score. The grouth truth facts consist of entities, types, values.

We perform extraction by finding

$$\arg\max_{z,a,m} p(z, a, m \mid x, k).$$

We obtain the fact (k_j, v_j) at word x_i if $m_i = 1$, $a_i = j$ and $z_i = v_j$.