Objectives for Information Extraction

Justin T. Chiu

June 28, 2019

Abstract

Many recent information extraction systems predict the relationship between an entity and value given their locations in the text. This requires observed locations of mentions, which requires annotations at the word level. Any form of annotation at the word level does not scale as the size of the text and the number of labels increases, and even more so if there is ambiguity. In order to train a probabilistic information extraction model without any supervision at the level of text, we specify a model that, for each word, either chooses a triple from a knowledge base to explain or chooses to explain nothing.

1 Problem Setup

In relation extraction (RE) we extract facts from a passage of text. The goal of RE is to convert facts expressed in natural language into a form amenable to computation. RE focuses on a relational representation: facts are extracted into a form that details how values are related to entities. This relational representation is useful for performing computation. Rather than dealing with noisy text, we may perform automatic deduction by composing facts in relational form.

In this work we build a relation extraction system with minimal supervision. We assume supervision only at the proposition level of a KB, with no annotation given at the level of text. In order to perform extraction, we must first identify where facts are. As sentences may contain many entities and values, and therefore possibly a large number of facts, we focus on identifying facts at the word level. In particular, we focus on identifying the location of value mentions and predicting who (the entity) and what (the relation type) the value mention is discussing.

Note on related work: Except for Zeng et al. (2018), prior work has either assumed that the locations of entities and values are given as input features or that the locations of entities and values are observed at training time.

We are primarily concerned with the scenario where we have an overcomplete KB schema with respect to a specific passage of text. This fits many scenarios in real world applications: we may have thousands of entities of interest if our KB was pulled from an external source such as Freebase, but the particular document we wish to analyze only discusses tens of entities, only a few of which are present in our KB.

Table 1: Notation for relation extraction.

x	$\triangleq$	A sequence of words $x_i, i \in \mathcal{I}$ corresponding to a sentence or document.
$\mathcal{X}$	$\triangleq$	The vocabulary.
$\mathcal{I}$	$\triangleq$	The index set for the text: $\mathcal{I} = \{0, 1, 2, \ldots\} \subset \mathbb{N}$.
r	$\triangleq$	The knowledge base, an indexed set of records each consisting of entities, types, and values. Each $r_j = (e_j, t_j, v_j)$.
$\mathcal{J}$	$\triangleq$	The index set for the KB: $\mathcal{J} = \{0, 1, 2, \ldots\} \subset \mathbb{N}$.
e	$\triangleq$	The list of all entities in a KB by index. Each $e_j \in \mathcal{E}$.
t	$\triangleq$	The list of all types in a KB by index. Each $t_j \in \mathcal{T}$.
v	$\triangleq$	The list of all values in a KB by index. Each $v_j \in \mathcal{V}$.
$f_*(x)$	$\triangleq$	A learned function whose output is of shape $\lvert\mathcal{I}\rvert \times D$, where $\lvert\mathcal{I}\rvert$ is the length of the word sequence and D is the context-dependent.
c_i	$\triangleq$	A word level RV that indicates whether a word is content or not. We have $c_i \in \{0, 1\}$.
a_i	$\triangleq$	A word level RV that indicates an the index of the aligned for word x_i. We have $a_i \in J$.
z_i	$\triangleq$	A word level RV that indicates the canonical value word x_i is translated to, such that $z_i \in \mathcal{V}$

The problem description is as follows: given a text $x = [x_0, \ldots, x_{|\mathcal{I}|}], x_i \in \mathcal{X}$, we must model the facts $r = \{(e_j, t_j, v_j)\}_{j \in \mathcal{J}}$ expressed in that text with respect to a schema that details all entities $e_j \in \mathcal{E}$, relation types $t_j \in \mathcal{T}$, and all values $v_j \in \mathcal{V}$. We assume that the schema of the KB $(\mathcal{E}, \mathcal{T}, \mathcal{V})$ is known at all times, and that the schema covers all facts of interest.

We refer to the set of facts r as a knowledge base (KB), and each fact r_j is referred to as a record. Each record r_j consists of an entity, type, and value triple. Let e, t, v be the projection of the entities, types, and values from the KB r in aggregate.

Given all entities e and types t, we reduce the construction of r to predicting, for every $j \in \mathcal{J}$, the value v_j corresponding to the entity e_j and type t_j. We propose to model $p(v \mid x, e, t)$, the distribution over values of corresponding entities and types given the text.

2 Model

We define a (mixed) directed graphical model that jointly models word and KB level extraction. The model first predicts the word level information, then aggregates its predictions at the KB level.

We introduce the following random variables for each index $i \in [I]$:

1. c_i indicates whether a word is a value mention. We have $c_i \in \{0, 1\}$.
2. a_i indicates the index of the record aligned to word x_i. We have $a_i \in J$.
3. z_i indicates the canonical value word x_i is a mention of, such that $z_i \in \mathcal{V}$.

Let f_* be arbitrary learned functions that maps each x_i in x to a point on a simplex. In all cases, f_* takes the form of a BLSTM over the sequence of words x with a linear transformation applied to the output at each time step i. The dimensionality of the simplex can be inferred from the domain of the distribution parameterized by f_*. The model $p(v \mid x)$ has the following generative story. For each index $i \in \mathcal{I}$, we have

1. Value mention $p(c \mid x) = \prod_i p(c_i \mid x)$: We choose whether word x_i is a value mention or not. Each $c_i \sim \text{Bern}(f_c(x)_i)$, where a value of 1 indicates that x_i is a value mention.
2. Alignment $p(a \mid x) = \prod_i p(a_i \mid x, e, t)$: We align the word y_i by choosing who (the entity) and what (the relation type) are being talked about. In particular, $a_i \sim \text{Cat}(f_a(x, e, t)_i)$ denotes the alignment to the record r_{a_i} given by the index $j = a_i$.
3. Translation $p(z \mid x) = \prod_i p(z_i \mid x)$: We translate the word x_i into a value z_i. The $z_i \sim \text{Cat}(f_z(y)_i)$ is the canonical value from the KB schema associated with x_i.

Finally, we aggregate the word level information at the sequence level:

4. Aggregation $p(v \mid z, a, c) = \prod_j p(v_j \mid z, a, c)$: Given the word level values z,alignments a, value mention decisions$c$, we choose the KB level value v_j from a conditional random

field. This choice is made independently from the other values in the KB given the word level choices. We parameterize

$$p(v_j \mid z, a, c, e, t) \propto \prod_i \exp(\psi(v_j, z_i, a_i, c_i, e, t)),$$

with

$$\psi(v_j, z_i, a_i, c_i, e, t) = \begin{cases} f_v(v_j, e, t), & c_i = 0 \\ 1(v_j = z_i, a_i = j), & c_i = 1. \end{cases}$$

We model the marginal distribution $p(z \mid x)$ rather than the full joint distribution $p(v, z \mid x)$ with an undirected model because we are interested in extracting the facts expressed in the text without the influence of the values in a KB.

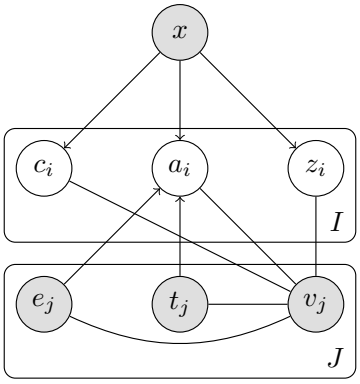

Figure 1: Our model predicts word-level values and alignments then aggregates those choices over all indices i to predict a value at the KB level.

This gives us the following factorization of the relation extraction system:

$$\begin{aligned} p(v \mid x, e, t) &= \sum_{z,a,c} p(v, z, a, c \mid x, e, t) \\ &= \sum_{z,a,c} p(v \mid z, a, c, x, e, t) \prod_i p(z_i, a_i, c_i \mid x, e, t) \\ &= \sum_{z,a,c} \prod_j p(v_j \mid z, a, c, x, e, t) \prod_i p(z_i \mid x) p(a_i \mid x, e, t) p(c_i \mid x) \end{aligned} \tag{1}$$

3 Training

To train a latent variable model, we must marginalize over the unobserved RVs and maximize the likelihood of the observed. However, maximizing $\log p(v \mid x)$ directly is very expensive for this model. The aggregation model alone with z unobserved is a Restricted Boltzmann Machine, where exact inference is $O(|\mathcal{V}|^{|\mathcal{I}|} + |\mathcal{V}|^{|\mathcal{J}|})$.

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

3.1 Inference network

Our first approach is to specify an inference network $q(z, a, c \mid v, x, e, t)$. We then optimize the following lower bound on the marginal likelihood with respect to the parameters of both p and q:

$$\log p(v \mid x) \geq \mathbb{E}_{q(z,a,c|v,x,e,t)} \left[\log \frac{p(v, z, a, c \mid x, e, t)}{q(z, a, c \mid v, x, e, t)} \right] \tag{2}$$

We propose to parameterize $q(z, a, c \mid v, x, e, t)$ as follows. We decompose

$$\begin{aligned} q(z, a, c \mid v, x, e, t) &= q(z \mid a, v, x) q(a \mid v, x, e, t) q(c \mid v, x) \\ &= \prod_i q(z_i \mid a, v, x) q(a_i \mid v, x, e, t) q(c_i \mid v, x) \end{aligned} \tag{3}$$

The conditional distributions of our inference network are very similar to the relation extraction model, but they condition on the values v. Let $g_*(\cdot)$ be a function that maps its arguments to a $|\mathcal{I}| \times D$ dimension tensor, where $g_*(\cdot)_i$ returns a point on the D dimensional simplex. At each index $i \in \mathcal{I}$, the conditional distributions are given by:

1. The value mention model $q(c_i \mid v, x) = \text{Bern}(g_c(v, x)_i)$ has access to the values v from the KB, which it conditions on when detecting value mentions.

2. The alignment model $q(a_i \mid v, x, e, t) = \text{Cat}(g_a(v, x, e, t))$ uses a contextual representation of each x_i and chooses a record. In contrast to $p(a \mid x, e, t)$, this model has access to values as well.

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.

3.2 Approximate the posterior of a generative model

Alternatively, we may decompose the training of our extraction system $p(v \mid x)$ into two stages: In the first stage we train $p(z, a, c \mid x, e, t)$ to approximate the posterior of a conditional model of text given a complete KB $q(x, z, a, c \mid e, t, v)$. This has the benefit of allowing us to exert control over where value mentions are detected through our design of the text model q.

In the second stage, we have two choices: a) train $p(v \mid z, a, c, x, e, t)$ to approximate the posterior of a full generative model of text and the values of KB $q(x, v \mid e, t)$. b) train $p(v \mid z, a, c, x, e, t)$ using the following lower bound:

$$\log p(v \mid x) \geq \mathbb{E}_{p(z,a,c|x,e,t)} \left[\log p(v \mid z, a, c, x, e, t) \right] \tag{4}$$

Ideally the bound in Eqn. 4 should not be looser than the one presented in Eqn. 2, as conditioning on the observed values of a KB should not reduce the entropy of a good alignment model.

4 Evaluation

Although we have a model over the values of all records, we are not interested in explaining every record. As we assumed that the KB contained a superset of the facts contained in a sequence of text, we are only interested in a subset of the records.

References

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.