Objectives for Information Extraction

Justin T. Chiu

June 30, 2019

Abstract

Many recent information extraction systems predict the relationship between an entity and value given the positions of their mentions in the text. This requires requires words to be annotated as mentions. Human annotation at the word level does not scale as the size of the text and the number of labels increases, as annotators must read every word. Automatic methods allow annotation to scale, 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, for each word, either chooses a triple from a knowledge base to explain or chooses to explain nothing.

1 Problem Statement

In relation extraction we extract facts from a passage of text. The goal of relation extraction is to convert facts expressed in natural language into a form amenable to computation. Relation extraction uses a relational representation: facts detail how values are related to entities. The challenge is to not only extract facts from text, but also determine where those facts are mentioned.

As sentences may contain many entity and value mentions, and therefore a large number of facts, we identify facts at a scale that minimizes ambiguity by casting the mention identification problem as classifying whether each word is a mention. We first identify the position of value mentions, predict who (the entity) and what (the relation type) the value mention is discussing, then aggregate those decisions.

Once we have identified where facts are mentioned, we must aggregate those decisions to to offer .

We assume access to a knowledge base (KB) that is aligned to sequences of text that discuss subsets of the information in the KB.

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.

In this work we build a relation extraction system with minimal supervision. 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.

The problem description is as follows: given a text $x = x_0, \ldots, x_I$ we model the facts $r = \{(e_j, t_j, v_j)\}_{j=0}^{J}$ 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. The set of facts r is our knowledge base (KB), and each individual fact r_j is a record. For brevity, let $e = \{e_j\}_{j=0}^{J}, t = \{t_j\}_{j=0}^{J}, v = \{v_j\}_{j=0}^{J}$ be the list of the entities, types, and values of the records in r respectively.

Given all entities e and types t, we reduce the construction of r to predicting, for every $j \in 0, \ldots, J$, the value v_j corresponding to the entity e_j and type t_j. We propose a model for this distribution $p(v \mid x, e, t)$.

2 Model

We define a graphical model that jointly models word and KB level extraction. The model first extracts information at the word level, then aggregates its predictions at the sequence level:

The word level extraction process has three steps. For each index $i \in 0, \ldots, I$

1. Value mention identification: Given a sequence of words x, we identify whether each word is a value mention with $p(m \mid x) = \prod_i p(m_i \mid x)$. Each $m_i \in \{0, 1\}$. Not every word in a mention must be identified; it suffices to find at least 1 word in a value mention.

2. Alignment: Each value mention is then aligned to a record in the knowledge base with $p(a \mid x) = \prod_i p(a_i \mid x, e, t)$, We align the word y_i by choosing who (the entity) and what (the relation type) generate the possible value mention at index i. In particular, $a_i = j$ denotes the alignment to the record r_j with $a_i \in 0, \ldots, J$. We assume that each value mention aligns to a single record.

3. Translation: All value mentions are translated into a representation from the KB schema with $p(z \mid x) = \prod_i p(z_i \mid x)$, with $z \in \mathcal{V}$.

We choose to locally normalize the word level distributions as our goal is to extract information from text, not condition on an existing KB.

Finally, in case there is disagreement on values at the word level, we aggregate the word level information at the sequence level in order to give a single distribution over the record values for x.

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

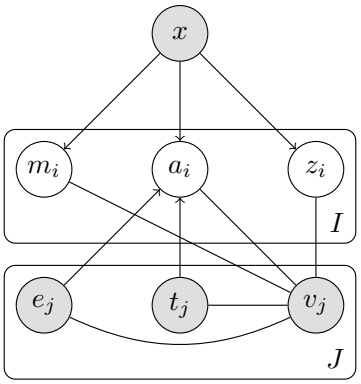

Figure 1: Our model predicts word-level values and alignments then aggregates those choices over all indices i to predict values at the KB level.

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

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

2.1 Parameterization

Our model has four steps: mention identification, mention alignment, mention translation, and aggregation. We parameterize the conditional distributions of each step 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 entities and types 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 decompose the alignment distribution into a distribution over entities $p(\epsilon_i \mid x)$ and types $p(\tau_i \mid x)$.

$$\begin{aligned} p(a_i \mid x) &= p(\epsilon_i \mid x) p(\tau_i \mid x) \\ p(\epsilon_i \mid x) &\propto \exp(E(e_{\epsilon_i})^T W_e \mathbf{h}_i) \\ p(\tau_i \mid x) &\propto \exp(E(\tau_{a_i})^T W_t \mathbf{h}_i) \end{aligned}$$

with $W_e \in \mathbb{R}^{d' \times d}, W_t \in \mathbb{R}^{d' \times d}$.

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 v1: In this version, we have a potential at each index where if an index is a value mention it votes for its value translation, otherwise it backs off to a prior.

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

with

$$\psi(v_j, z_i, a_i, m_i, e, t) = \begin{cases} E(v_j)^T W_v [E(e), E(t)], & m_i = 0 \\ 1(v_j = z_i, a_i = j), & m_i = 1. \end{cases}$$

The main issue with this parameterization is that if a sequence is very long, the sum of the priors will outweigh a single mention.

5. Aggregation v2:

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

with

$$\psi(v_j, z_i, a_i, m_i, e, t) = 1(v_j = z_i, a_i = j, m_i = 1)$$

6. Aggregation v3:

$$\begin{aligned} p(v_j \mid z, a, m = 0, e, t) &= p(v_j \mid e, t) \\ p(v_j \mid e, t) &\propto \exp(E(v_j)^T W_v [E(e), E(t)]) \\ p(v_j \mid z, a, m, e, t) &\propto \prod \exp(\psi(v_j, z_i, a_i, m_i, e, t)) \\ \psi(v_j, z_i, a_i, m_i, e, t) &= 1(v_j = z_i, a_i = j, m_i = 1) \end{aligned}$$

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. Marginalizing over just z has computational complexity $O(|\mathcal{V}|^I)$, which is exponential in the length of the text.

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

3.1 Inference network

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

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

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

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

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

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.

2. The alignment model $q(a_i \mid v, x, e, t)$ 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.

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 $x \in \mathcal{V}$ we have $q(z = x)$ is high, biasing the translation model towards transliteration at the start of training.

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 $p(z, a, m \mid x, e, t)$ to approximate the posterior of a conditional model of text given a complete KB $q(x, z, a, m \mid e, t, v)$. 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 $p(v \mid z, a, m, x, e, t)$ to approximate the posterior of a full generative model of text and the values of KB $q(x, v \mid e, t)$. b) train $p(v \mid z, a, m, x, e, t)$ using the following lower bound:

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

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, evaluation does not include the final distribution over all record values. As we assumed that the KB contained a superset of the facts contained in a sequence of text, we are evaluate whether the model can discover the subset of facts that are expressed in the text. We therefore perform extraction by using the marginal distributions $q(z), q(a), q(c)$ to value mentions as well as entities and types, giving us facts.

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.