Objectives for Information Extraction

Justin T. Chiu

June 25, 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 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 Introduction

In relation extraction we extract facts from a passage of text. The length of the text varies: depending on the dataset we may be interested extracting all facts expressed in a sentence or paragraph. The datasets we use do not consider the case where all facts from a large corpus are to be extracted, as supervision would be very expensive. All sequences of text are bounded in length.

Given a text $y = [y_0, \ldots, y_T]$ the goal is to predict all facts $r = \{(e_i, t_i, v_i)\}_{i \in I}$ expressed in that text. We refer to the set of facts r as a knowledge base (KB), and each fact r_i is referred to as a record. Each record r_i consists of an entity, type, and value triple. Let e, t, v be the entities, types, and values from the KB r in aggregate.

We assume supervision only at the proposition level of a KB. We know what the relationship is between entities and values (although this assumption can be relaxed), but we do not know where in text records are realized. 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 relax this assumption and do not use any annotations at the level of text.

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

y	$\triangleq$	A vector of words y_t
r	$\triangleq$	The knowledge base, an indexed set of records each consisting of entities, types, and values.
e	$\triangleq$	The list of all entities in a KB by index.
t	$\triangleq$	The list of all types in a KB by index.
v	$\triangleq$	The list of all values in a KB by index.

2 Model

Our goal is to produce a set of record triples given text. As we assume that only the KB and text are observed, we can either produce this set directly from the text or identify realizations of records at a more granular level and then aggregate our choices. We choose to pursue the latter since identifying facts at a granular level presents a step towards compositional language understanding, i.e. shallow parsing. Although entity or value mentions may be multi-word expressions, identifying realizations at the word level is no less expressive as we can aggregate our word level choices into sequence level ones.

Inspired by work in language grounding Liang et al. (2009), we focus on identifying value mentions in text. Our model first identifies words that are value mentions, translates the mention into a canonical value, then aligns those words to a record in the KB by predicting what entity and the relation type associated with the value mention.

Our information extraction model is denoted $q(r \mid y)$, a distribution over a KB r given some text y. Our word level model is given by the following: For each word y_t, we have

1. Value mention detector $q(c_t \mid y)$: We classify whether word y_t is a value mention. Each $c_t \sim \text{Bern}(f(y))$, where a value of 1 indicates that y_t is a value mention.

2. Translation $q(v_t \mid y)$: We translate the word y_t into a value v_t. The $v_t \sim \text{Cat}(g(y))$ is the canonical value from the KB schema associated with y_t.

3. Alignment $q(a_t \mid y)$: We align the word y_t by classifying who (the entity) and what (the relation type) are being talked about. In particular, $a_t \sim \text{Cat}(h(y))$ denotes the alignment to the record r_{a_t} given by the index $i = a_t$.

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

1. Aggregation $q(r_i \mid a, v, c)$: In order to parameterize a distribution over the KB r, we must then aggregate all of the word-level information into a distribution over facts at the sequence level.

3 Three perspectives on training

We can either train q directly on the conditional task or train it to mimic the posterior of a suitable generative model.

Axes of objectives:

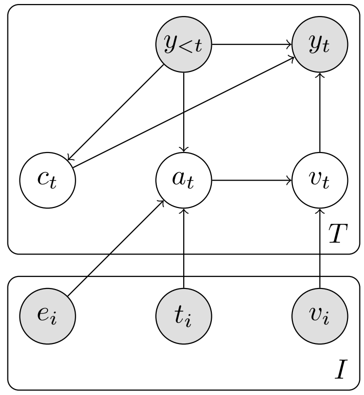

Figure 1: The generative model which produces words given a knowledge base.

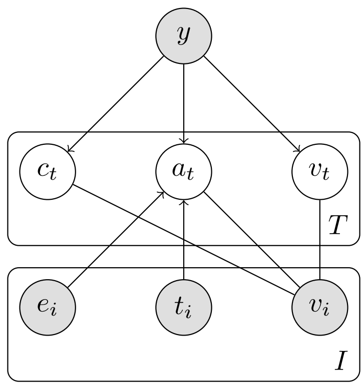

Figure 2: The inference network which predicts word-level values and alignments.

1. Proposal distribution: Learned or uniform
2. Probabilistic interpretation: Marginal likelihood or KL
3. Probabilistic interpretation 2: Approximate posterior of generative model or learn directly.

3.1 Marginal loss

3.2 KL

3.3 Approximating the posterior of a generative model

Why would we do this? The main motivation is semi-supervised information extraction, where values are missing.

$$\arg\max_p \sum_{y'} \sum_{x'} p^*(y', x') \log \frac{p(y', x')}{p^*(y', x')} = \arg\max_p \sum_{y'} \sum_{x'} p^*(y', x') \log \sum_z p(y', z, x') \tag{1}$$

References

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 91–99, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.