Information Extraction with Weak Supervision

Keywords information networks, natural language processing, information extraction, latent variable models

Relevance to BAA As information networks get larger and more complex, acquiring explicit supervision for the training of information extraction systems becomes extremely expensive. Both the representation of the information in the knowledge base as well as the extraction process itself must be hand-designed. In this proposal we present a framework towards automating the training of information extraction systems with minimal supervision.

Background Natural language processing can largely be decomposed into two separate but closely related tasks: natural language understanding (NLU) and natural language generation (NLG). We argue that the two tasks are complementary, and capitalize on their duality by proposing a method to train a NLU system without direct supervision. More specifically, we utilize the latent variable model framework to train an information extraction system using the performance of a generative model of text as learning signal. Intuitively, we want an information extraction system to extract information that best explains the given text. We focus on the task of summarizing a table of data, which we refer to as Data-to-Text (D2T).

Why is D2T important? Unlike tasks such as translation or summarization where one would need to first understand a source passage then generate a target passage, D2T isolates the task of NLU from NLG by introducing an intermediate representation in the form of structured data. This makes the task similar to semantic parsing, program induction, and instruction following, except for the lack of compositionality in D2T's intermediate representation. Whereas the other tasks assume an executable intermediate representation, D2T has a latent representation that takes the form of alignments from words in the summary to their respective referent data. This allows D2T to serve as a useful benchmark for conditional text generation because of the reduced complexity. In addition to establishing a convenient representation, D2T allows us to assume relatively safely that the text is generated conditioned only on functions of the data table. This assumption is central to the task as it allows practitioners to define heuristics both for providing weak supervision to an information extraction system as well as evaluating a summary's fidelity to the underlying data.

Rotowire We provide a concrete example from a D2T dataset: In this proposal we focus on the recently proposed the Rotowire dataset [5]. Rotowire contains summaries of basketball games aligned with the respective box scores of those games. We consider a box score to be a collection or set of facts, where a fact is a tuple of an entity, relation type, and value. For example, consider the collection: a collection that consists of a single fact, (entity = Jeremy Lin, type = POINTS, value = 19), and a simple statement "Jeremy Lin scored 19 points". The relation type is the label in the box score, for example POINTS, REBOUNDS, etc. In order to automatically evaluate this summary, thanks to our assumption the summary was generated from only the data, we are able to use an information extraction system (IE) to extract entity-value pairs from text, predict the type of the relation between the entity and value, and compare resulting tuple (entity, type, value) to the given collection of facts.

Our proposal Our goal is to maximize the coverage of the information extraction system, which we define as the number of words contained within segments of text that are correctly aligned to

data, while minimizing the amount of supervision needed. We propose to learn an information extraction (IE) system as the approximate posterior distribution over alignments from segments of text to their generating data. We also propose to view the given data as incomplete and learn boolean-valued functions of the data as a step towards representation learning.

Problem Definition Rotowire consists of aligned box score data and basketball game summaries $\{(\mathbf{r}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{r}^{(2)}, \mathbf{y}^{(2)}) \dots\}$. For brevity, we refer to a single data and summary as $\mathbf{r}, \mathbf{y}$, omitting the superscript. Each data $\mathbf{r} = \{r_1, \dots, r_N\}$ is a set of N records, each of which has an entity, type, and value $r_i = (e_i, t_i, v_i)$. We refer collectively to all the entities, types, and values in a given data $\mathbf{r}$ as $\mathbf{e}, \mathbf{t}, \mathbf{v}$ respectively. Each summary $\mathbf{y} = \{y_1, \dots, y_T\}$ is a sequence of tokens that makes up the text of the game description.

For generation, the goal is to learn a conditional model $p(\mathbf{y} \mid \mathbf{r})$ of the text given the data. This is simple to evaluate, as we can use the log-likelihood of a given summary under our model as a measure of performance. In Wiseman et al. [5] model takes the form of a conditional language model that can copy values directly from records in $\mathbf{r}$. Subsequent work in Puduppully et al. [3] decomposed the distribution $p(\mathbf{y} \mid \mathbf{r}) = \sum_{\mathbf{c}} p(\mathbf{y} \mid \mathbf{c}) p(\mathbf{c} \mid \mathbf{r})$ by introducing a content plan $\mathbf{c}$, which is a sequence of records drawn from $\mathbf{r}$. This was also previously implemented in prior work [2], which modelled the text generation process through a hierarchical hidden semi-Markov model.

We divide the information extraction task into three subtasks. Before outlining the tasks, we propose two measures of performance through which to evaluate an unsupervised information extraction system. The first is how well the information extracted from a summary allows a generative model to reconstruct the summary measured by the likelihood of the summary given the extracted information. We refer to this as reconstruction. The second is coverage, which we define as the number of words contained in segments that are aligned to a record in $\mathbf{r}$. (Do I need to argue why these are useful? And also how the tasks aim at increasing them by weakening assumptions or constraining model flexibility compared to previous work)

The first task (ALIGN) is to align segments of text to the records that generated them. This is similar to learning a content plan $\mathbf{c} \mid \mathbf{r}$ as in Puduppully et al. [3], however we are interested in the **posterior** distribution $p(\mathbf{c} \mid \mathbf{r}, \mathbf{y})$ of the content plan $\mathbf{c}$ after observing the text $\mathbf{y}$. Liang et al. [2] utilize the fact that they define a model in which posterior inference is tractable, however tractability does not hold once the latent distribution becomes autoregressive. In Wiseman et al. [5] and subsequently in Puduppully et al. [3] this was accomplished by separately training a classifier to predict the type t of an entity e and value v pair within a sentence. The entity and value are extracted heuristically by checking exact string matches within $\mathbf{e}$ and $\mathbf{v}$, and the supervision over t is obtained through the following function [5]: $\text{findType}(\hat{e}, \hat{v}) = \{r.t : x \in \mathbf{r}, r.e = \hat{e}, x.r = \hat{v}\}$. However, this limits alignments exclusively to entities and values explicitly in $\mathbf{r}$. We would like to align whole segments of text in order to increase the coverage of our information extraction system.

The second task (VALUES) is to reconstruct values v in the table $\mathbf{r}$. This is implemented on top of task (ALIGN). In particular, we want to learn $p(\mathbf{v} \mid \mathbf{y}, \mathbf{c}, \mathbf{e}, \mathbf{t})$, the distribution over all values given the summary, the content plan, all entities, and all types.

The third task (FUNCTIONS) is the most ambitious. In order to demonstrate the flexibility of the framework, we propose a method to further learn functions of $\mathbf{r}$ in an unsupervised manner. (TODO)

Model We begin by defining a model for (ALIGN) and proceed to (VALUES) and (FUNCTIONS).

Our generative model factors into the likelihood and prior: $p(\mathbf{y}, \mathbf{c} \mid \mathbf{r}) = p(\mathbf{y} \mid \mathbf{c}) p(\mathbf{c} \mid \mathbf{r})$. Our likelihood $p(\mathbf{y} \mid \mathbf{c})$ is given by a conditional neural language model with a copy mechanism as in

Gülçehre et al. [1], Wiseman et al. [5] in addition to monotonic attention [4, 6]. The prior $p(\mathbf{c} \mid \mathbf{r})$ is an autoregressive model over records parameterized with an LSTM. As we are primarily interested in posterior inference, the performance of the prior is not the most important aspect of the model. In fact, we will see in the next section that the prior serves to regularize our approximate posterior. (This is the simplest baseline aside from HSMM, can include $p(\mathbf{y}, \mathbf{c} \mid \mathbf{r}) = \prod_t p(y_t \mid \mathbf{y}_{<t}, c_t) p(c_t \mid \mathbf{c}_{<t}, \mathbf{y}_{<t}, \mathbf{r})$ if necessary)

Our initial IE model for (ALIGN) is given by $q(\mathbf{c} \mid \mathbf{y}, \mathbf{r})$, which includes both a segmentation of the summary as well as the alignments. Note that we denote the distribution using q since under the generative model $p(\mathbf{c} \mid \mathbf{y}, \mathbf{r})$ is well-defined as the posterior distribution of alignments given a summary and records. Initially we parameterize the approximate posterior $q(\mathbf{c} \mid \mathbf{y}, \mathbf{r}) = \prod_t q(c_i \mid \mathbf{y}, \mathbf{r})$ as a fully factored distribution over alignments. (Structured attention for pairwise potentials later, since only has node potentials for now. This is motivated by coverage) Each $q(c_i \mid \mathbf{y}, \mathbf{r})$ is parameterized by the output of a BLSTM run at the sentence level. (TODO: values, functions)

Learning and Inference A latent variable model $p(\mathbf{y}, \mathbf{z} \mid \mathbf{x})$ models an observed $\mathbf{y}$ as well as an unobserved $\mathbf{z}$ conditioned on $\mathbf{x}$. When fitting such a model, we would like to maximize the evidence $p(\mathbf{y} \mid \mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z} \mid \mathbf{x})$ which marginalizes over the latent $\mathbf{z}$. Depending on the structure of $p(\mathbf{y}, \mathbf{z} \mid \mathbf{x})$, the marginalization procedure may be intractable to perform exactly. For example, this is the case with an autoregressive model for the latent variable $p(z_t \mid \mathbf{z}_{<t})$, where variable elimination's runtime would be exponential in the length $|\mathbf{z}|$. This also precludes tractable posterior inference, i.e. $p(\mathbf{z} \mid \mathbf{y}, \mathbf{x})$, since by Bayes' Rule we have $p(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) = p(\mathbf{y}, \mathbf{z} \mid \mathbf{x}) / p(\mathbf{y} \mid \mathbf{x})$ which requires evaluating the intractable sum in the evidence $p(\mathbf{y} \mid \mathbf{x})$. Therefore we resort to learning an approximation of the posterior through the variational principle: the best approximation within a family of distributions is the one with minimal KL-divergence to the model's posterior. The KL between the approximate posterior and true posterior is still intractable to minimize exactly, so we instead maximize the evidence lower bound ($\mathcal{L}_q$), which is the evidence minus the posterior KL:

$$\mathcal{L}_q = \underbrace{\log p(\mathbf{y} \mid \mathbf{x})}_{\text{Evidence}} - \underbrace{D_{KL}[q(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) || p(\mathbf{z} \mid \mathbf{y}, \mathbf{x})]}_{\text{Posterior KL}} \tag{1}$$

$$= \underbrace{\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z} \mid \mathbf{y}, \mathbf{x})} [\log p(\mathbf{y} \mid \mathbf{z}, \mathbf{x})]}_{\text{Reconstruction}} - \underbrace{D_{KL}[q(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) || p(\mathbf{z} \mid \mathbf{x})]}_{\text{Prior KL}}. \tag{2}$$

Were it not for local extrema, maximizing this quantity would maximize the evidence and minimize the posterior KL jointly. Notice that in (2), the objective we use for training both the generative model and IE system, all expectations are taken with respect to the IE system or approximate posterior $q(\mathbf{y} \mid \mathbf{z}, \mathbf{x})$. (TODO: training procedure via REINFORCE + control variate, posterior constraints for incorporating information from findType function)

For task (ALIGN), we have the fully observed summary $\mathbf{y}$, the unobserved content plan $\mathbf{z} = \mathbf{c}$, and all records as conditioning $\mathbf{x} = \mathbf{r}$. For task (VALUES), we again have the observed summary $\mathbf{y}$, but we pretend the values are unobserved $\mathbf{z} = \{\mathbf{c}, \mathbf{v}\}$, and use the rest of the records as conditioning $\mathbf{x} = \{\mathbf{e}, \mathbf{t}\}$. (TODO: functions)

References

[1] Çaglar Gülçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. *CoRR*, abs/1603.08148, 2016. URL `http://arxiv.org/abs/1603.08148`.

[2] Percy Liang, Michael I. Jordan, and Dan Klein. 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, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9. URL `http://dl.acm.org/citation.cfm?id=1687878.1687893`.

[3] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. *CoRR*, abs/1809.00582, 2018. URL `http://arxiv.org/abs/1809.00582`.

[4] Sam Wiseman. Learning neural templates for text generation. *CoRR*, abs/lol, 2018. URL `http://arxiv.org/abs/lol`.

[5] Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. Challenges in data-to-document generation. *CoRR*, abs/1707.08052, 2017.

[6] Lei Yu, Jan Buys, and Phil Blunsom. Online segment to segment neural transduction. *CoRR*, abs/1609.08194, 2016. URL `http://arxiv.org/abs/1609.08194`.