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. Information networks are also dynamic; over time one method for representing information may become inadequate. Currently, both the representation of the information in a 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.

Introduction Natural language processing contains two separate but closely related subfields: natural language understanding (NLU) and natural language generation (NLG). Recent approaches to information extraction use only the NLU perspective and frame extraction as a classification problem. We argue that the two perspectives, NLU and NLG, are complementary and capitalize on their duality by proposing a method to train a NLU system without direct supervision. More specifically, we train an information extraction system by using the performance of a deep generative model as signal. By training in this fashion we obtain an information extraction system that extracts facts that best explain the given text.

Recent work using neural network-based systems cast information extraction as a supervised problem [2, 8]. Although approaches do incorporate structure into internal representations, for example to integrate multiple sources of information [6], the final output still uses strong supervision during training. Finding strong supervision is a difficult task, and as datasets get larger we must approach information extraction from a different perspective. Latent variable models (LVMs) are one method for alleviating the need for supervision. LVMs do not require manual labels, as they instead treat quantities of interest as latent random variables and deal with them in a probabilistically principled fashion. We will leverage recent advances in deep generative modeling in order to scale and extend their model while still training without explicit supervision.

Neural models for language generation have seen much progress in recent years, with state of the art performance in both language modeling and translation [**?**] (MoS + Transformer). By integrating the powerful language modeling capabilities of neural networks with the inductive biases of graphical models through LVMs, we can leverage the LVM framework to derive a principled method for training an information extraction system with less supervision. Namely, we turn to an efficient technique for training hard attention that relies on variational inference [1]. This technique is applicable to sequential LVMs such as hidden Markov models and hidden semi-Markov models (HSMMs). HSMMs have been applied to small scale datasets for text generation [7] as well as without integrating neural networks [4]. We will scale sequential LVMs to large datasets through variational inference and use the signal from our generative model in order to train an information extraction system.

Background We consider datasets consisting of aligned data and text $\left\{(\mathbf{r}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{r}^{(2)}, \mathbf{y}^{(2)}) \ldots\right\}$. For brevity, we refer to a single datum and text as $\mathbf{r}, \mathbf{y}$, omitting the superscript. Each datum $\mathbf{r} = \{r_1, \ldots, r_N\}$ is a set of N records, each of which has an entity, type, and value $r_i = (e_i, t_i, v_i)$. The datum $\mathbf{r}$ is a flattened representation of an information network. Whereas an information network may also be represented as a hypergraph, $\mathbf{r}$ is a list of relations or records. We refer collectively to all the entities, types, and values in a given datum $\mathbf{r}$ as $\mathbf{e}, \mathbf{t}, \mathbf{v}$ respectively. Each text $\mathbf{y} = \{y_1, \ldots, y_T\}$ is a sequence of tokens.

Let random variables $\mathbf{z}$ be unobserved or latent, $\mathbf{y}$ observed, and $\mathbf{x}$ taken as conditioning and thus not modelled. For information extraction we are interested in distributions $p(\mathbf{z} \mid \mathbf{y}, \mathbf{x})$, where $\mathbf{z}$ and $\mathbf{x}$ may correspond to various quantities depending on the task but $\mathbf{y}$ is always the text.

As a concrete example, we use the Rotowire dataset [8]. Rotowire contains summaries of basketball games $\mathbf{y}$ aligned with the respective box scores $\mathbf{r}$ of those games. Consider a datum that consists of a single record, $\mathbf{r} = \{(e_1 = \text{Jeremy Lin}, t_1 = \text{POINTS}, v_1 = 19)\}$, and a simple statement $\mathbf{y} =$"Jeremy Lin scored 19 points". In a simple case, the process of information extraction may be to infer any subset of $\mathbf{r}$, which in this case will be our latent $\mathbf{z}$, given the remaining elements in $\mathbf{r}$ which corresponds to $\mathbf{x}$, as well as the text $\mathbf{y}$. For example, we may want to infer the type of the relation t given the entity Jeremy Lin, the value 19, as well as the text $\mathbf{y}$. In this case, we would have $\mathbf{z} = \{t_1\}$ and $\mathbf{x} = \{e_1, v_1\}$. In an alternative task, we may want to infer the value v_1 as well as the type t_1 given $\mathbf{y}$ and e_1, therefore $\mathbf{z} = \{t_1, v_1\}$ and $\mathbf{x} = \{e_1\}$.

Note that we are not constrained to setting $\mathbf{z}$ to subsets of $\mathbf{r}$. We also consider the case where $\mathbf{z}$ includes alignments from individual words y_t to records r_i. We denote the alignments $\mathbf{a} = \{a_1, \ldots, a_T\}$, where each a_t is associated with y_t and selects a record r_i such that $a_t = i$.

Proposal We propose to demonstrate the efficacy of the LVM framework in the weakly supervised information extraction setting. We do so by estimating the distribution over alignments and values given the text, entities, and types. This is denoted by $p(\mathbf{z} \mid \mathbf{y}, \mathbf{x})$, where $\mathbf{z} = \{\mathbf{a}, \mathbf{v}\}$ and $\mathbf{x} = \{\mathbf{e}, \mathbf{t}\}$. In the previous section, we defined $\mathbf{a}$ to be a latent variable that represents the alignments from words to records, while $\mathbf{v}$ corresponds to all the values in a datum of records and $\mathbf{e}, \mathbf{t}$ the entities and types respectively. This distribution $p(\mathbf{z} \mid \mathbf{y}, \mathbf{x})$ is the IE system, which infers the alignments and values given the text, entities, and types.

Since we do not assume that we have supervision for $\mathbf{z}$, we cannot learn the information extraction system directly. Insead, we must learn a conditional model of the text and latent variables given the conditioning $p(\mathbf{y}, \mathbf{z} \mid \mathbf{x})$ since we observe $\mathbf{y}$. We will subsequently show how to use this conditional model to train the IE system. We define the conditional model with the following generative story:

1. Value Choice: For each pair of entities and types in our datum of records, we predict a value.
2. Record Choice: Conditioned on our choices of values as well as the given entities and records, we choose a sequence of records $\mathbf{a} = \{a_1, \ldots, a_I\}$, denoted by their indices, to describe.
3. Word Choice: For each record alignment a_i, we choose a sequence of words $\mathbf{y}_i = \{y_{i1}, \ldots, y_{iJ}\}$ to describe the record.

JUNK PAST THIS POINT.

Old Prop 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 $\{(\mathbf{r}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{r}^{(2)}, \mathbf{y}^{(2)}) \ldots\}$. For brevity, we refer to a single data and summary as $\mathbf{r}, \mathbf{y}$, omitting the superscript. Each data $\mathbf{r} = \{r_1, \ldots, r_N\}$ 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 $\mathbf{y} = \{y_1, \ldots, y_T\}$ 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. [8] the 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. [5] 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 [4], 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. [5], 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. [4] 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. [8] and subsequently in Puduppully et al. [5] 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 [8]: $\text{findType}(\hat{e}, \hat{v}) = \{r.t : x \in \mathbf{r}, r.e = \hat{e}, x.r = \hat{v}\}$. 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. [3], Wiseman et al. [8] in addition to monotonic attention [7, 9]. 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:

$$\begin{aligned}
\mathcal{L}_q &= \underbrace{\log p(\mathbf{y} \mid \mathbf{x})}_{\text{Evidence}} - \underbrace{D_{KL}[q(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) || p(\mathbf{z} \mid \mathbf{y}, \mathbf{x})]}_{\text{Posterior KL}} && (1) \\
&= \underbrace{\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z} \mid \mathbf{y}, \mathbf{x})} \left[\log p(\mathbf{y} \mid \mathbf{z}, \mathbf{x})\right]}_{\text{Reconstruction}} - \underbrace{D_{KL}[q(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) || p(\mathbf{z} \mid \mathbf{x})]}_{\text{Prior KL}}. && (2)
\end{aligned}$$

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)

Outline

1. Introduction
   (a) Relevance to BAA
      i. What are information networks? Characterized by large graphs and voluminous data.
      ii. Representation is key
      iii. Supervision is expensive.
      iv. Information networks are also dynamic; over time one method for representing information may become inadequate.
      v. Currently, both the representation of the information in a knowledge base as well as the extraction process itself must be hand-designed.
      vi. In this proposal we present a framework towards automating the training of information extraction systems with minimal supervision.
   (b) Decomposition of NLP into NLU and NLG
      i. Two separate but closely related subfields: natural language understanding (NLU) and natural language generation (NLG).
      ii. Recent approaches to information extraction use only the NLU perspective and frame extraction as a classification problem.
      iii. 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.
   (c) Supervision in NLU
      i. Recent work using neural network-based systems cast information extraction as a supervised problem [2, 8].
      ii. Even with structured representations [6], the final output is still supervised.
      iii. Again, supervision does not scale.
      iv. Maybe dynamism of information networks? Don't have an argument on hand, though.
      v. Latent variable models (LVMs) are one method for alleviating the need for supervision.
      vi. LVMs do not require manual labels, as they instead treat quantities of interest as latent random variables and deal with them in a probabilistically principled fashion.
   (d) Recent advances in NLG
      i. Deep generative models, namely LVMs with neural network components, have integrated the flexibility of neural networks with the inductive biases of graphical models.
      ii. Most importantly, efficient techniques for training hard attention that rely on variational inference [1]. This technique is applicable to sequential LVMs such as hidden Markov models and hidden semi-Markov models (HSMMs).
      iii. HSMMs have been applied to small scale datasets for text generation [7] as well as without integrating neural networks [4].
2. Background

  (a) Formal notation for elements of the dataset

  (b) Define the distribution we would like to learn: $p(z \mid y, x)$. z and x are placeholders and will change, but y is always the text.

  (c) Link to rotowire example (argument is that ACE is made up of ontonotes-like sentences, so all short-form)

3. Proposal

  (a) Define IE as the distribution we would like to learn

    i. Align: $p(c \mid \mathbf{y}, \mathbf{r})$

    ii. Values: $p(c, \mathbf{v} \mid \mathbf{y}, \mathbf{e}, \mathbf{t})$ (Just this one)

    iii. ??: $p()$

  (b) Define generative model: h-HSMM as in [4], but with neural components as in [7]. The generative story:

    i.

  (c) Training and Inference

    i. Maximize ELBO

  (d) Experiments

    i.

  (e) Conclusion

    i.

References

[1] Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M. Rush. Latent alignment and variational attention. *CoRR*, abs/1807.03756, 2018. URL `http://arxiv.org/abs/1807.03756`.

[2] Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. *CoRR*, abs/1504.06580, 2015. URL `http://arxiv.org/abs/1504.06580`.

[3] 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`.

[4] 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`.

[5] 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`.

[6] Dirk Weissenborn, Tomas Kocisky, and Chris Dyer. Dynamic integration of background knowledge in neural nlu systems. *CoRR*, abs/1706.02596, 2017. URL `http://arxiv.org/abs/1706.02596`.

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

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

[9] 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`.