Information Extraction with Weak Supervision

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

Relevance to Army BAA: II. A. c. iii. (3) Information Networks In order to provide decision makers with the information necessary to make informed choices as well as predict the effects of those choices, we must have an efficient representation of relevant information and a predictive model for the resultant state of the world given a choice. Information networks provide a graphical representation of information and how it propagates through a network. We focus on *knowledge graphs*, information networks where each node contains a set of facts about an entity and an edge describes how the facts in one node influence the facts in another. Knowledge graphs provide an intuitive and queryable representation of knowledge. A decision maker may query the relevant nodes to gain situational awareness, and when simulating a decision that alters the information in one node the graph can easily propagate those changes by virtue of its representation.

Given that a knowledge graph must represent an extremely large number of facts and relationships, it is infeasible to specify these completely by hand. Many recent works have focused on learning to fill in the missing edges of a knowledge graph by recognizing patterns in fully-labeled subgraphs in order to predict whether there should be an edge between two nodes in an unlabeled subgraph [2]. By filling in missing edges, we are able to use the relationships specified by the edges in order to reason about the facts contained in a node conditioned on the facts of its neighbours.

We propose an orthogonal approach which instead relies on jointly modeling text as well as the nodes of the information network. We aim to leverage copious amounts of natural language text by learning an information extraction system that is able to extract facts from text to fill in the nodes of a knowledge graph. In this proposal we present a method towards automating the training of information extraction systems with unlabeled corpora by recasting the information extraction problem as a generative modeling problem. Specifically, we plan to use the performance of a deep generative model of text as signal for learning an extraction system.

(Include exemplar knowledge graph diagram, and show that we focus on filling nodes paired with text) (Diagram highlighting the difference between modeling edges without text for KG completion vs only modeling nodes jointly with text. Highlight complementary nature and emphasize combination as future work)

Introduction The goal of information extraction is to produce structured representations of information given unstructured text. We use summaries of basketball games as a running example. In the context of a basketball game summary, an information extraction system would infer all the statistics associated with a player given the summary. A typical approach to an information extraction system is the following pipeline, where each stage has a model that is trained independently from the other stages: first segment the text into entities and values, then extract named entities and possibly perform coreference resolution by predicting whether segments refer to the same entity, and finally relation extraction where we identify the relationships between the extracted entities and values. We can then use the extracted relationships along with the associated entities and values to populate the nodes of a knowledge graph, where each node would correspond to a player and contain their associated statistics. See Figure 1 for an example of this process. This is orthogonal and complementary to using existing nodes in a knowledge graph to fill in missing ones. As we are interested in utilizing large amounts of unlabeled corpora for the training of an information extraction system, we restrict our focus to learning the values of each node independently given

the text. Neural networks, in combination with latent variable models (LVMs), provide a method for training such an information extraction pipeline jointly and end-to-end.

The progression from a pipelined system into one that is trained jointly has precedent in the field of natural language processing; it recently occurred in machine translation. Previously, statistical machine translation utilized a highly pipelined approach, where each stage utilized a model that was trained independently of the others. The approach was unified with an end-to-end neural model in Bahdanau et al. [1]. We recently recast Bahdanau et al. [1]'s model in the LVM framework in Deng et al. [3] which resulted in performance gains as well as improved sample complexity. This progression parallels our proposal: we wish to specify an information extraction system that is learned end-to-end rather than in a pipeline, and we use the LVM framework to do so. LVMs provide a principled way of specifying either semi-supervised or unsupervised models [5], and have been shown to have better sample complexity than discriminative models [3, 7].

Although LVMs have recently been proposed in the context of knowledge graph completion [2, 8], they either do not utilize text or do not have a generative model of text. We argue that the generative model we specify should be as close to the data generating process as possible. In particular, Chen et al. [2] does not utilize text at all and Qu et al. [8] do not incorporate generative model of the text. We propose to explicitly model how a text is created given a knowledge graph using a conditional generative model with latent variables.

In this proposal, we focus on the class of LVMs known as hidden semi-Markov models (HSMMs), used in Liang et al. [6] as a generative model for the task of aligning segments of text to nodes in a knowledge graph without supervision. As in Liang et al. [6], we are interested in learning a generative model of text so that we can minimize the amount of supervision necessary for training an information extraction system. The performance of their generative model of text provides signal for learning the alignments: if the likelihood of a segment of text improves when moving from a past alignment choice to a new one, then the new alignment is more likely to be correct given a suitably strong likelihood model. We use that same intuition to formulate a related LVM for weakly supervised information extraction which aims to model not just the alignments from segments of text to nodes in a knowledge graph but also the values in the nodes themselves. By parameterizing the generative HSMM with neural networks as in Wiseman [9], we hope to incorporate recent progress in parameterizing LVMs with neural networks so as to learn a more accurate information extraction system by using a more powerful generative model – taking care to ensure that the model can be trained with minimal supervision. (cite Deng et al. [3] for training strategies)

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, where each record $r_i = (e_i, t_i, v_i)$ is a tuple containing an entity, type, and value. The datum $\mathbf{r}$ is a knowledge base, or equivalently an information network without a representation of the causal effects between 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 each from a vocabulary V.

We proceed to detail how to specify a LVM, then provide a concrete example linking the above dataset description to our LVM formulation: Let 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 that can be specified as $p(\mathbf{z} \mid \mathbf{y}, \mathbf{x})$, 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 [10]. 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 its simplest incarnation, 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 value v_1 given the entity Jeremy Lin, the type POINTS, as well as the text $\mathbf{y}$. In this case, we would have $\mathbf{z} = \{v_1\}$ and $\mathbf{x} = \{e_1, t_1\}$. 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 $\mathbf{z} = \{v_1, t_1\}$ and $\mathbf{x} = \{e_1\}$.

(clarify) 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 $a_t = i$. (NEED TO FIX ALIGNMENTS AND T vs t, HMM vs HSMM)

Proposal We propose to verify the efficacy of the LVM framework in the weakly supervised information extraction setting, with the goal of demonstrating strong extractive performance with minimal labels. We present one instance of a LVM and outline how it can be used to obtain an information extraction model without direct supervision, then argue that the same approach can be applied in even more ambitious settings. Our first LVM is a conditional model that specifies the relationship between data, specifically the entities and types, and text. We denote this model `Values`. Similar to the models defined in Wiseman [9] and Liang et al. [6], our model takes the form of a hidden semi-Markov model (HSMM). The primary difference is that the other models simply assumed the records were complete and conditioned on them, whereas ours learns to generate the values. `Values` is given by the following generative process:

1. Value Choice: For each pair of entities and types in our datum of records, we predict a value. We assume that each record type constrains the values to be members of a finite set. Thus each record type is assigned a categorical distribution over its respective values, and the values are drawn independently from that respective distribution. Each $v_i \sim \text{Cat}(f_{t_i}(e_i))$ is drawn from a Categorical distribution, whose parameters $f_{t_i}(e_i)$ are output by a neural network f_{t_i} that is shared across record types and takes as input the entity e_i.

2. Record Choice: Conditioned on our choices of values as well as the given entities and records, we choose a sequence of records $\mathbf{a} = \{a_1, \ldots, a_I\}$ to describe, given by their index a_i. Note that each record choice is described by at least one token, hence we have $I \leq T$. The record choices are parameterized as a Markov model where each $a_t \sim \text{Cat}(f_\theta(a_{t-1}, \mathbf{v}, \mathbf{e}, \mathbf{t}))$, where f_θ is a neural network. (NEED TO FIX ALIGNMENTS AND T vs t, HSMM vs HMM)

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. The words are modelled by an autoregressive emission model within each segment that is aligned to the same record: $y_{ij} \sim \text{Cat}(f_\theta(\mathbf{y}_{i1:ij}, v_{a_i}, e_{a_i}, t_{a_i}))$, where f_θ is another neural network and $\mathbf{y}_{i1:ij}$ is all tokens y from indices $i1$ to ij.

The value and record choices correspond to prior distributions over values $p(\mathbf{v} \mid \mathbf{e}, \mathbf{t})$ and alignments $p(\mathbf{a} \mid \mathbf{v}, \mathbf{e}, \mathbf{t})$ respectively, while the word choice model gives us the likelihood of some text given our value and alignment choices $p(\mathbf{y} \mid \mathbf{a}, \mathbf{v}, \mathbf{e}, \mathbf{t})$. In this case, we have latent $\mathbf{z} = \{\mathbf{v}, \mathbf{a}\}$ and observed $\mathbf{x} = \{\mathbf{e}, \mathbf{t}\}$. We obtain an information extraction by using the **posterior** distribution over alignments and values:

$$p(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) = \frac{p(\mathbf{y}, \mathbf{z} \mid \mathbf{x})}{p(\mathbf{y} \mid \mathbf{x})} = \frac{p(\mathbf{y}, \mathbf{z} \mid \mathbf{x})}{\sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z} \mid \mathbf{x})}.$$

Although the HSMM formulation allows the summation (marginalization) over alignments to be carried out efficiently, we cannot marginalize over value assignments. We instead resort to variational inference as in Deng et al. [3], where we learn an approximation of the posterior distribution $q(\mathbf{z} \mid \mathbf{y}, \mathbf{x})$ with a separate model. Given that we are primarily interested in the values rather than the alignments, we can obtain an information extraction system over only values by marginalizing over alignments. By marginalizing over the alignment distribution, the model propagates uncertainty over alignments to uncertainty over values. We can train this model by maximizing a lower bound on the log marginal likelihood or evidence of $\mathbf{y}$, called the evidence lower bound (ELBO) which is given formally by the expression $\text{ELBO}_q \triangleq \mathbb{E}_{q(\mathbf{z}|\mathbf{y},\mathbf{z})} [\log p(\mathbf{y} \mid \mathbf{z}, \mathbf{x})] \leq \log \mathbb{E}_{p(\mathbf{z}|\mathbf{x})} [p(\mathbf{y} \mid \mathbf{z}, \mathbf{x})]$. This objective can be maximized with gradient-based methods. The resulting approximate posterior $q(\mathbf{z} \mid \mathbf{y}, \mathbf{x})$ can be used independently of the generative model as an information extraction system that gives a distribution over values in a table of records and alignments from text to records.

As we are interested in applying the LVM framework to information extraction rather than a single model, we proceed to motivate and outline an extension to `Values`. In addition to inferring values, ideally an information extraction system would also be able to infer new entities and relation types. We propose a possible route towards defining model that can learn to parameterize new relation types in an unsupervised manner using the same LVM framework as `Values`, which we denote `Types`. The motivation behind `Types` is that a segment of text may refer to multiple records at the same time. For example, in basketball games a 'triple-double' refers to more a player achieving a value of more than 10 in any three of the five categories: points, steals, rebounds, blocks, or assists. The goal of `Types` is to attempt to capture the latent relationship behind utterances such as 'triple-double' appearing in the text and the underlying records in an unsupervised manner. We plan to introduce a new step to the generative process of `Values` that allows the model to learn new records as boolean-valued functions of relations already defined in the data. By approximating these boolean functions with neural networks, we hope to find a model parameterization that admits a low variance Monte Carlo gradient estimator.

We will evaluate our initial approach on the Rotowire dataset, and extensions to our model that will include entity tracking and event resolution on the automatic content extraction (ACE) [4] and the Text Analysis Conference's Streaming Multimedia Knowledge Base Population (SM-KBP) datasets. We expect the variance of the gradient estimator to be an issue, in particular its effect on sample complexity. In previous work, we observed that gradient estimators based on exact inference resulted in better sample complexity than Monte Carlo gradient estimators [3], and we expect that controlling the variance of the gradient estimator with the inductive bias of neural architectures will be of paramount importance for the success of our proposed method.

Given admittance to the NDSEG Fellowship Program, I will evaluate the application of LVMs to the problem of information extraction. As a result of the digital age, the ubiquity of information networks as well as their enormous growth makes it clear that a method for training information extraction systems with minimal supervision is a necessity. I will push for scalable information extraction systems that require minimal supervision by recasting information extraction as a generative modeling problem.

Outline

1. Introduction
   (a) Relevance to BAA
      i. Importance of representation and correctness of knowledge in decision making, as well as the power of prediction.
      ii. Information networks are one way of representing knowledge as a graph as well as interactions?
      iii. What are information networks? Graphical representation of objects, their characteristics of interest, and their relationship to objects.
      iv. Recent work focuses on learning structure between nodes in order to fill parts of the graph that may be missing [2].
      v. We focus on the orthogonal approach of completing knowledge graphs using large amounts of unstructured text rather than the relationships between nodes.
      vi. Reduce information networks to knowledge bases by removing edges (kind of interesting as we can view knowledge bases as a mean field approximation to information networks where info networks must specify not only conditional distributions but also interventional distributions).
      vii. In this proposal we present a framework towards automating the training of text-centric information extraction systems with minimal supervision.
      viii. Shine some hope on interventional distribution representations? (This might be a little difficult)
   (b) Supervision in Information Extraction
      i. Define information extraction
         A. The goal is to produce structured representations of information from a given unstructured text.
         B. Ideally, but not necessarily computer-readable in addition to human-readable.
         C. A typical pipeline for information extraction includes text segmentation, named entity recognition, coreference resolution, relation extraction, and finally producing structured representations of the unlabeled text.
         D. This is knowledge-base completion.
         E. We are primarily interested in knowledge-base completion, as the other tasks such named entity recognition are typically part of a pipeline aimed at knowledge-base completion.
      ii. Argue for LVM approach to unify all parts of the pipeline and train end-to-end with minimal supervision.
      iii. We aim to unify these two approaches through recently developed techniques for training LVMs parameterized with neural networks.
   (c) End-to-end trends and LVMs
      i. Draw parallels to progress in translation, mainly in terms of structure?
         A. Statistical machine translation: pipelines
         B. End-to-end with Sutskever
         C. Deterministic structure via attention Bahdanau

D. Latent attention Deng et al. [3]

ii. Leverage modularity of both the neural network and LVM frameworks to incorporate more of the pipeline into a joint training framework.

iii. Qu et al. [8] Semi-supervised relation extraction (which is pretty much similar) How is what I want different? Capitalize on the progress of powerful generative models, can argue that their model is much weaker in terms of generative power, but the framework is similar (they perform coordinate ascent in a LVM).

(d) Recent advances in neural LVMs

i. Semi-supervised LVMs Kingma et al. [5]?

ii. Demonstrate that parameterization with a neural network does not affect computational complexity of inference.

iii. Then the same technique can be applied to model with more structure, as long as the graphical model itself permits tractable inference.

iv. In this proposal, we focus on the hidden semi-Markov model (HSMM), used in Liang et al. [6] for the task of aligning segments of text to records in a knowledge base without supervision.

v. As in Liang et al. [6], we are interested in learning a generative model of text so that we can minimize the amount of supervision necessary for training an information extraction system.

vi. Also that although worse sample complexity, using an approximate posterior with monte carlo sampling achieves comparable performance.

2. Background and Notation

(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)

(d) Clarify that the scope of posterior inference is very general.

3. Proposal

(a) Outline approach

i. Choose a subset of available data as conditioning, and thus it is not modelled.

ii. The joint distribution of the remaining variables, both observed and unobserved, will be modelled.

(b) Link back to motivation. We want to scale information extraction by requiring less supervision.

(c) We present one model as an example, then later demonstrate how the framework can handle extensions of the model.

(d) Define generative model: HSMM as in [6], and [9]. The generative story (a picture would be helpful):

i. Fill in values

ii. Choose alignments

iii. Choose words

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

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

(f) We either use the posterior distribution of the conditional model or learn an approximation of it.

(g) Argue that the segmental model encourages more coverage by adding pairwise dependencies between labelings [6]. We will check how much structure in the generative model aids information extraction.

(h) Training and Inference

i. As we are dealing with documents of significant length, we train with an approximate posterior in order to satisfy memory constraints.

ii. Highlight that the approx posterior is a SEPARATE model that can be used completely independently from generative model, i.e. we throw away generative model after training.

iii. We maximize a lower bound on the log marginal likelihood, called the evidence lower bound.

(i) Extension, `Types`

i. Introduce new step in generative process

ii. Learn a boolean function that composes predicates applied to existing records

iii. The search space is very large, so we must either constrain our model in a very clever way or obtain large amounts of data as any stochastic gradient estimator will have very high variance

(j) Extensions

i. learn new types as functions of existing ones

ii. learn a randomly initialized embedding of the type and a neural network directly to predict the value

iii. let the input distribution be

(k) Experiments, evaluation, and expectation

i. Evaluate on Rotowire? Highlight long-form text

ii. Also on ACE

iii. As corpora may be too large, we might need more hierarchy in the generative model

iv. and also since memory is linear in the length of the sequence, we may have to resort to approximate inference. We can optimize a lower bound on the marginal likelihood with an approximation of the posterior distribution [3].

v. Evaluation metrics: for slot-filling, we evaluate the

(l) Conclusion

i. Please accept!

ii. Recap: Minimal supervision IE systems so that they can scale to extracting information for large information networks from large bodies of text.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL `http://arxiv.org/abs/1409.0473`.

[2] Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. Variational knowledge graph reasoning. *CoRR*, abs/1803.06581, 2018. URL `http://arxiv.org/abs/1803.06581`.

[3] 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`.

[4] T. George Doddington, C. Alexis Mitchell, T. Mark Przybocki, N. Lance Ramshaw, C. Stephanie Strassel, and N. Ralph Weischede. The automatic content extraction (ACE) program–tasks, data, and evaluation. 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.78.8442&#38;rep=rep1&#38;type=pdf`.

[5] Diederik P. Kingma, Danilo Jimenez Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014. URL `http://arxiv.org/abs/1406.5298`.

[6] 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`.

[7] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 841–848, Cambridge, MA, USA, 2001. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2980539.2980648`.

[8] Meng Qu, Xiang Ren, Yu Zhang, and Jiawei Han. Overcoming limited supervision in relation extraction: A pattern-enhanced distributional representation approach. *CoRR*, abs/1711.03226, 2017. URL `http://arxiv.org/abs/1711.03226`.

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

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