Information Extraction with Distant 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 after a choice is made. Information networks provide a graphical representation of information and how it propagates through a network. I 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. I propose to fill in the nodes of a knowledge graph by leveraging a generative model of corpora. Many recent works have taken an orthogonal direction known as link prediction, which fills in the missing *edges* of a knowledge graph [1]. Through link prediction one uses the relationships specified by the edges to reason about the facts contained in a node conditioned on the facts of its neighbours. However, we generally also have access to large amounts of unlabeled and unstructured data in the form of natural language corpora. I aim to leverage natural language corpora by learning an information extraction system that is able to extract facts from text to fill in the nodes of a knowledge graph given only the assumption that the text was generated from the information represented in the knowledge graph.

In this proposal I present a method towards automating the training of information extraction systems with unlabeled corpora by recasting the information extraction problem as a conditional generative modeling problem. We propose to perform information extraction by first modeling the generative process of writing the text given the data, then inverting that process in a probabilistically principled manner through posterior inference.

Background The goal of information extraction is to produce structured representations of information given unstructured text. Figure 2 is an example of a datum and text pair which may be used to train an information extraction system, inspired by the Rotowire dataset [8]. A typical approach to an information extraction system is the following pipeline:

1. First segment the text into mentions and values, i.e. michael jordan and all numerical values.
2. Then align the mentions to a entity, a node in a knowledge graph. The node of interest is labelled ‘michael jordan’.
3. Finally identify the relationships between segments. The relationships between these segments entail facts, which are below the summary in Figure 2.

Given the large cost of obtaining labels to train an extraction system, a model that can perform well with less supervision is appealing. For example, consider the case where Figure 1 is missing all values except for Michael Jordan’s name. In this scenario, where there is an availability of text but a lack of facts, learning to write the text is much easier than training an extraction system since the lack of facts results in a lack of labels for the extraction system. A model that creates text as well as facts is called a generative model. A generative model is defined by its generative process, which is a recipe for how the dataset is created. A

Figure 1: A simplified example of a summary generated from a set of facts about an entity. A full dataset consists of many unaligned sets of facts and sentences.

Entity	Type	Value	Summary
michael jordan	NAME	michael jordan	michael jordan had 22 points and 12 rebounds as well as four blocks
michael jordan	POINTS	22	
michael jordan	REBOUNDS	12	
michael jordan	BLOCKS	4	

generative model that includes latent or unobserved variables in its generative process is a latent variable model (LVM). The benefit of using a LVM rather than directly modeling facts given text is two-fold:

1. A LVM can learn in the semi-supervised or unsupervised setting, where only a few labels or no labels are provided respectively. This implies we can train a model with missing or noisy facts.
2. LVMs provide a principled approach to inverting their generative process, allowing one to reason about unobserved quantities earlier in the process given observations later in the process.

In this proposal, I focus on the class of LVMs known as hidden semi-Markov models (HSMMs), used in Liang et al. [5] as a conditional generative model for the task of aligning segments of text to nodes in a knowledge graph without supervision. Liang et al. [5] rely on the insight that a conditional generative model of text provides signal for learning the alignments: if the likelihood of a segment of text improves when moving from one alignment choice to a another, then the new alignment is more likely to be correct given a suitably strong likelihood model. The same intuition can be used to formulate a LVM for a semi-supervised information extraction system 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. We build on the formulation in Liang et al. [5] by parameterizing our LVM with neural networks, drastically increasing the generative model's capacity.

Problem Setup 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, I 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.

The Rotowire dataset [8] is an example of such a dataset. Rotowire contains summaries of basketball games $\mathbf{y}$ aligned with the respective box scores $\mathbf{r}$ of those games. Consider the datum in Figure 2 that consists of three records and the statement $\mathbf{y} =$ "michael jordan had 22 points and 12 rebounds as well as four blocks". For this example, the process of information extraction is to infer the values $\mathbf{v}$ of the records given the entities $\mathbf{e}$, types $\mathbf{t}$, and the text $\mathbf{y}$.

Proposal We propose to verify the efficacy of the LVM framework in the semi-supervised information extraction setting, with the following goals:

1. By formulating a LVM for generating text conditioned on data, obtain an information extraction system through posterior inference.
2. Demonstrate strong extractive performance with minimal labels.
3. Move towards a model for knowledge graph completion that captures the full joint distribution.

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

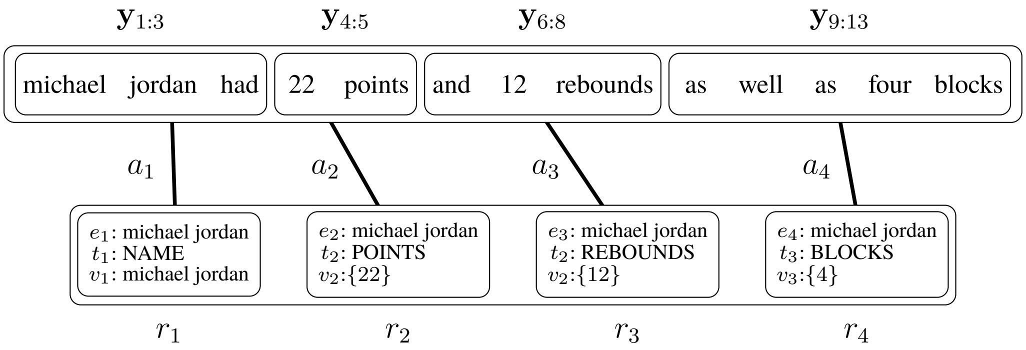

Figure 2: An example of our proposed information extraction procedure with an inferred segmentation and values, which the model gathers evidence for from the text. The values that must be inferred are in curly braces.

settings. Our proposed LVM is a conditional generative model that specifies the relationship between data, specifically the entities and types, and text. We denote this model `Values`. Our model takes the form of a hidden semi-Markov model (HSMM), where the model learns to generate the values, record alignments, and finally the text.

Our model assumes observed entities and relation types $\{\mathbf{e}, \mathbf{t}\}$, as well as latent values and alignments $\{\mathbf{v}, \mathbf{a}\}$. `Values` is given by the following generative process:

1. Prior Value Choice: $p(\mathbf{v} \mid \mathbf{e}, \mathbf{t})$. For each entity and type pair in our datum of records, predict a value. For example, given 'michael jordan' and POINTS, we predict 19. Each value is predicted independently.
2. Prior Record Choice: $p(\mathbf{a} \mid \mathbf{v}, \mathbf{e}, \mathbf{t})$. Conditioned on our choices of values as well as the given entities and records, in other words a completed data with no missing values, choose a sequence of records $\mathbf{a} = \{a_1, \ldots, a_T\}$ to describe with a Markov model.
3. Word Choice: $p(\mathbf{y} \mid \mathbf{a}, \mathbf{v}, \mathbf{e}, \mathbf{t})$. For each alignment a_i, choose a sequence of words $\mathbf{y}_i = \{y_{i1}, \ldots, y_{iJ}\}$ to describe the record indicated by the alignment. With the HSMM formulation, we have that within each segment aligned to a record, the words are modeled autoregressively.

An information extraction system is obtained by inverting this process to obtain the **posterior** distribution over alignments and values:

$$p(\mathbf{a}, \mathbf{v} \mid \mathbf{y}, \mathbf{e}, \mathbf{t}) = \frac{p(\mathbf{y}, \mathbf{a}, \mathbf{v} \mid \mathbf{e}, \mathbf{t})}{p(\mathbf{y} \mid \mathbf{e}, \mathbf{t})} = \frac{p(\mathbf{y}, \mathbf{a}, \mathbf{v} \mid \mathbf{e}, \mathbf{t})}{\sum_{\mathbf{a}, \mathbf{v}} p(\mathbf{y}, \mathbf{a}, \mathbf{v} \mid \mathbf{e}, \mathbf{t})}.$$

Although the HSMM formulation allows the summation over alignments to be carried out efficiently, the sum over value assignments is intractable. Instead one must resort to variational inference, where an approximation of the posterior distribution $q(\mathbf{a}, \mathbf{v} \mid \mathbf{y}, \mathbf{e}, \mathbf{t})$ is learned with a separate model. The conditional generative model and the approximate posterior are trained jointly by maximizing a lower bound on the log marginal likelihood of $\mathbf{y}$ with gradient-based methods. The resulting approximate posterior $q(\mathbf{a}, \mathbf{v} \mid \mathbf{y}, \mathbf{e}, \mathbf{t})$ 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.

I will evaluate the approach on the TAC KBP 2015 slot filling, TACRED [10], and ROTOWIRE [8] datasets in order to compare to previous work. The goal will be to demonstrate competitive performance on extraction metrics such as precision, recall, and F1, while using as little supervision as possible by ignoring subsets of the given information. For example, with the model `Values`, we can allow the model to only learn from subsets of the given values (or none at all) in ROTOWIRE, such as only the home team's statistics, at training time. Given success in that goal, the next step would be to extend the model to capture more of the joint distribution with the aim of boosting sample and label efficiency. Possible extensions in this direction include coreference resolution [2], learning the types of relations in a semi-supervised manner, leveraging discourse structure to improve the conditional generative model [7], explicitly modeling nuisance variables such as author style [3], and incorporating multi-hop reasoning in order to leverage relationships between entities [1, 6].

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. Intro to information networks and KG
         A. Information networks and decision making
         B. Specify knowledge graphs as the information networks we are interested in
         C. Brief outline of knowledge graphs, ie nodes are entities, contain sets of facts, edges specify relationships between facts (can include self-loops)
      ii. Argument that KGs cannot be populated by hand. (Brief outline of methods for populating, no in-depth descriptions provided in this proposal)
         A. Link prediction
         B. Multi-hop link prediction
         C. Corpora-based
      iii. Proposal: train information extraction systems by recasting as a generative modeling problem.
   (b) Background
      i. We focus on information extraction, which is the task of producing structured representations of text.
      ii. Define information extraction
         A. The goal is to produce or fill in structured representations of information from a given unstructured text.
         B. 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.
      iii. Generative model
      iv. If we have more text than labels, or at best noisy incomplete labels from heuristic methods...
      v. In this case, it is easier to do the inverse problem: learn to generate text from facts.
      vi. We learn to generate text, which is the inverse of information extraction, with a generative model. We then use an algorithm called belief propagation to invert the generative process of the model to obtain an information extraction system.
      vii. Benefits
         A. stuff
   (c) Recent advances in neural LVMs
      i. Semi-supervised LVMs Kingma et al. [4]?
      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. [5] for the task of aligning segments of text to records in a knowledge base without supervision.
      v. As in Liang et al. [5], 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, which we will serve as a starting point for the proposed research.

    (d) Define generative model: HSMM as in [5], 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) Training and Inference

        i. As we are dealing with large state spaces, 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.

    (h) Experiments, evaluation, and expectation

        i. Evaluate on Rotowire.

        ii. We evaluate `Values` using the precision, recall, and F1 score on the task of predicting the values associated with entities, otherwise known as slot-filling.

        iii. What would success look like?

            A. Competitive to supervised methods when supervision is available

            B. But able to be applied when supervision is not available

            C. Able to leverage lots of unlabeled data during training, and success would see a marked improvement over purely supervised methods as well as the purely supervised version of this model.

        D. Would provide explanations for the answers (ie segmentations).

    iv. Also on ACE?

(i) Future work

   i. Incorporate more structure into the generative model, for example entity tracking or coreference resolution Haghighi and Klein [2].
   ii. Model more structure in the data, for example the edges between nodes in the knowledge graph Chen et al. [1].
   iii. 'Multi-hop' reasoning, where we try to compose relationships to infer new ones, i.e. through unification Chen et al. [1], Rocktäschel and Riedel [6].

(j) 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] 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`.

[2] Aria Haghighi and Dan Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 385–393, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL `http://dl.acm.org/citation.cfm?id=1857999.1858060`.

[3] Wei-Ning Hsu, Yu Zhang, and James R. Glass. Learning latent representations for speech generation and transformation. *CoRR*, abs/1704.04222, 2017. URL `http://arxiv.org/abs/1704.04222`.

[4] 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`.

[5] 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`.

[6] Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. *CoRR*, abs/1705.11040, 2017. URL `http://arxiv.org/abs/1705.11040`.

[7] Christina Sauper and Regina Barzilay. Automatically generating wikipedia articles: A structure-aware approach. 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 208–216, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9. URL `http://dl.acm.org/citation.cfm?id=1687878.1687909`.

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

[9] Sam Wiseman, Stuart Shieber, and Alexander Rush. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/D18-1356`.

[10] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45. Association for Computational Linguistics, 2017. doi: 10.18653/v1/D17-1004. URL `http://aclweb.org/anthology/D17-1004`.