The Physics of Text: Ontological Realism in Information Extraction

Stuart Russell
UC Berkeley
russell@berkeley.edu

Ole Torp Lassen
Roskilde University
otl@propercontext.org

Justin Uang
Palantir Technologies
justin.uang@gmail.com

Wei Wang
UPMC, Paris
benwei.wang@outlook.com

Abstract

We propose an approach to extracting information from text based on the hypothesis that text sometimes describes the world. The hypothesis is embodied in a generative probability model that describes (1) possible worlds and the facts they might contain, (2) how an author chooses facts to express, and (3) how those facts are expressed in text. Given text, information extraction is done by computing a posterior over the worlds that might have generated it. As a by-product, this unsupervised learning process discovers new relations and their textual expressions, extracts new facts, disambiguates instances of polysemous expressions, and resolves entity references. The probability model also explains and improves on Brin's bootstrapping heuristic, which underlies many open information extraction systems. Preliminary results on a small corpus of New York Times text suggest that the approach is effective.

1 Introduction

The purpose of information extraction (IE) is to produce both general knowledge structures and specific facts that will support inference, problem solving, and question answering. The primary difficulties include (1) the huge variety and ambiguity of linguistic expressions of underlying content; (2) the problem of resolving multiple entity references within and across documents; (3) the complexity of the underlying information itself: its ontology, temporal and causal structure, provenance, etc. Section 2 describes the major approaches that have been taken and their shortcomings.

Recent developments in probabilistic modeling and inference make it possible to revisit a Bayesian approach to IE championed by Charniak and Goldman (1992), among others. The approach is based on what one might call *ontologically realistic* generative models: that is, probability models that describe, in a very general sense, both the ways that the real world might be and the ways that world might be described in text. [1] Such models explain why this text is on the page in the same way that physical theories explain laboratory measurements—that is, by reference to an underlying reality. Perhaps surprisingly, commonly used generative models of language make no such reference.

A simple, initial model (Section 3) posits a world of facts (binary relations between entities) that are expressed using arbitrary dependency paths connecting named-entity mentions. Using the machinery of a probabilistic programming language such as BLOG (Milch and Russell, 2010) (augmented with a new form of proposal distribution for split–merge MCMC (Wang and Russell, 2015)) and a small, preprocessed corpus of New York Times sentences, preliminary results (Section 4) indicate that the approach is surprisingly effective in discovering relations, lexicons, and facts in an unsupervised fashion. A key advantage of this vertically integrated generative approach, compared to more classical bottom-up pipelines with deterministic stages, is that no hard decisions are made and all available context is ap-

[1] The word "realistic" in this context refers to the philosophical position of *realism*, usually ascribed to the Scottish School of Common Sense, which asserts that there is a real world and it is the subject of scientific theories and factual discourse.

Proceedings of AKBC 2016, pages 51–56,
San Diego, California, June 12-17, 2016.

plied to reduce uncertainty at every level, resulting in much higher accuracy (Pasula et al., 2003; Singh et al., 2013). Parsing, entity resolution, event recognition, and extraction emerge from a single, vertically integrated inference process—no special algorithms are needed.

2 Background and related work

The classical approach to IE involves a multi-stage pipeline: text goes in one end; each level processes small sets of input elements to produce larger elements; the output is usually a partially filled "template" describing a complex event. Intermediate levels include complex words, semantic elements (noun phrases, verb phrases), and elementary facts with unresolved entity references. Each pipeline stage requires manually created rules, specific to a particular domain, to recognize patterns among elements; or, learned classifiers can be used, but they require supervised training. The pipeline architecture has two major drawbacks: (1) decisions made using only local information are often wrong, and (2) errors propagate upwards leading to low overall accuracy.

Brin (1998) proposed a more scalable and automated approach called *bootstrapping*. Given a seed fact for a relation, such as $Author(CharlesDickens, GreatExpectations)$, bootstrapping aims to find all authorship facts and all textual patterns for expressing such facts. It alternates two steps:

1. Find sentences with known author–book pairs, e.g., "Charles Dickens wrote Great Expectations," and extract the pattern, "x **wrote** y."

2. Find sentences matching known patterns, e.g., "JK Rowling **wrote** Harry Potter," and extract the corresponding fact, $Author(JKRowling, HarryPotter)$.

Bootstrapping is effective but not perfect: for example, it finds *correlated facts* such as "JK Rowling made millions from Harry Potter" and concludes (optimistically, perhaps) that "x **made millions from** y" describes authorship; moreover, because of *polysemy* in "wrote", it finds "JK Rowling wrote Neville Longbottom out of the movie" and concludes an incorrect fact, $Author(JKRowling, NevilleLongbottom)$. Despite these issues, bootstrapping is the core of modern "open" IE systems, such as CMU's Never-Ending Language Learning (Mitchell et al., 2015). Section 3 describes an IE method grounded in probability theory that (1) generates bootstrapping inferences as a natural consequence, (2) explains why and when bootstrapping works, and (3) avoids the difficulties mentioned above.

As in other areas of NLP, recent work on IE has adopted statistical models for text. *Discriminative* models such as conditional random fields (CRFs) are trainable, bottom-up classifiers usable for the early stages of a pipeline approach; they require less manual labor than rule-based methods, although they do require supervised data. *Generative* models describe a stochastic process whose *output* is text; given some actual text, an inference algorithm can reconstruct the underlying hidden variables that would explain the observed text. Several other groups (see, e.g., Rink and Harabagiu (2011) or Yao *et al*. (2011) and many variants cited by Grycner et al. (2014)) are developing generative models for IE and relation discovery, but their models generate text from a descriptive model of text, rather than from a model of an underlying real world. That is, the hidden variables include the *dictionaries* describing the words that each relation uses to express itself and the *types* for each of its arguments, but not the *facts* that explain why the text is there. Thus, they cannot truly reconstruct such a world from text. The difficulty can be illustrated very simply: if one generates a very large sample of authorship sentences from such a model, one will find sentences claiming that every person has written every book. In real text, on the other hand, sentences are statistically coupled by an important latent variable, namely the real world. Thus, the statistics of corpora are completely different in the two cases. Because this difference is so important in understanding the model we propose, we return to it in more detail in Section 3.1.

3 An elementary generative model for declarative text

Here is a naive but ontologically realistic explanation for declarative text: the world contains facts; people choose to report some of those facts; they

choose a way to report each fact; the text is the collection of sentences that results. More precisely, the model assumes only that the world contains

• some unknown number N of objects, $x_1, \ldots, x_N$;

• some unknown number K of binary relations, $R_1, \ldots, R_K$;

• a collection of facts $R_k(x_i, x_j)$; each of the N^2 potential facts for R_k holds with probability σ_k, which is an unknown *sparsity parameter* between 0 and 1.

Here, N and K use a broad prior such as a discrete log-normal; σ_k has a beta prior with a small mean. In this naive, declarative world, a theory of pragmatics predicts what will be reported (Gordon and Durme, 2013). The initial theory is trivial: when writing sentence s_t, the author chooses a fact $R_k(x_i, x_j)$ at random from the set of true facts and reports it.

Then the semantic–syntactic model describes the text of sentence s_t given fact $R_k(x_i, x_j)$:

• a "verb" v is sampled from the relation R_k's dictionary D_k, an unknown categorical distribution over dependency paths that is drawn from a Dirichlet prior;

• the arguments x_i, x_j are mentioned verbatim as named entities (later versions have generative models for named-entity mentions);

• the final sentence is given by $s_t = x_i \; v \; x_j$.

This model can be written in roughly 10 lines of BLOG code. When supplied with suitable text as evidence, it automatically produces an improved and more robust version of bootstrapping, handles polysemy, needs no seed facts, and discovers new relations and their common forms of expression without requiring hand-engineered features measuring similarity of dependency paths.

Given a probability model that exhibits bootstrapping behavior, one can ask *why it works*: why is bootstrapping a reasonable inference in many cases? The answer is *sparsity*: given just the sentences "Charles Dickens **wrote** Great Expectations" and "Charles Dickens **authored** Great Expectations," one can show by simple calculation that the posterior odds that **wrote** and **authored** refer to the same relation are approximately proportional to $1/\sigma_k$. Intuitively, if they are not the same relation, the second sentence requires the reader to believe that a new

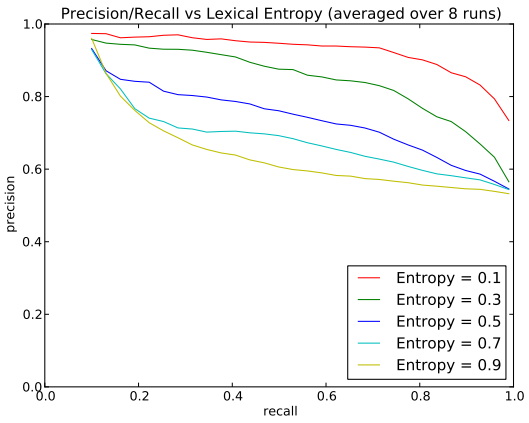

Figure 1: Measuring the ability of inference to extract accurate facts under polysemy: precision–recall curves for different levels of lexical entropy.

fact holds in the world, which is an unlikely coincidence if σ_k is small. And common sense suggests that nearly all relations are very sparse: for example, the relation matrix $Author(person, book)$ has a sparsity of approximately 1 in 7 billion. (In practice sparsity must be adjusted for the fact that authors are much more likely to be mentioned in text than the average human, so the effective sparsity is probably closer to 1 in ten thousand.) Inferences of this type are not only strong but also frequent, due to the "birthday paradox" phenomenon: given, say, a million facts for a relation expressible by two distinct patterns, a bootstrapping opportunity (when the two patterns are used with the same pair of arguments) occurs with overwhelming probability after about 2000 sentences. Each new bootstrapping opportunity of this kind connects two previously disjoint subgraphs of sentences; one expects that probabilistic analysis of graphs should become a key technique in the mathematics of information extraction.

One can also study properties of the mathematical model via simulation. For example, one can generate worlds whose relational dictionaries exhibit different levels of polysemy, generate text from those worlds, and measure the ability of inference to extract true facts. Let $d_{k,i}$ be the probability assigned by dictionary D_k to word (or dependency path) w_i, and define the *lexical entropy* of a collection of dictionaries $D_{k_1}, \ldots, D_{k_m}$ to be the frequency-weighted average over words w_i of the entropy

of the categorical distribution $[d_{k_1,i}, \ldots, d_{k_m,i}]/Z$. Then identical dictionaries (yielding completely ambiguous text) have lexical entropy 1 and disjoint dictionaries (yielding unambiguous text) have lexical entropy 0. The results shown in Figure **??** show that with lexical entropy 0.9, i.e., almost identical dictionaries that generate text that is impenetrable to mere humans, it is possible to achieve 90% precision at 10% recall—i.e., to reliably identify relations in some cases. Thus, the approach should be highly robust to polysemy in practice.

Like the original bootstrapping algorithm, it is susceptible to correlated facts, because it assumes that each relation samples its facts independently. Independence fails with relations such as marriage and divorce, since the latter implies the former; when the data contain "A got married to B" and "A just divorced B", bootstrapping assumes that "x got married to y" and "x just divorced y" mean the same thing. This is unfortunate. A similar problem arises with pairs of relations where one is a more specific version of the other, for example "x is a professor at y" and "x is employed by y". A more sophisticated model of reality fixes this problem; e.g., one allows a relation R_k to be either *de novo* or *highly correlated* (according to some unknown, but not small, correlation factor $\rho_{k,l}$) with some other relation R_l.

3.1 Comparison to other generative models for relational text

In Section 2, we claimed that an ontologically realistic model is fundamentally distinct from other generative models for relational text that have appeared in the literature. Here we go into more detail on this point, using as an example the Rel-LDA model in Figure 1 of Yao et al. (2011). (We choose Rel-LDA because of its superficial similarity to our model, not because of any particular failings.) In Rel-LDA, the generative process is as follows:

• For each relation $r \in \{1, \ldots, R\}$, multinomials $p_{r,1}$, $p_{r,v}$, $p_{r,2}$ are drawn from Dirichlet priors, representing distributions over "words" for the first argument, "verb", and second argument, respectively.

• For each document, a multinomial q over R values is drawn from a Dirichlet prior, where $q(r)$ indicates the probability that any given sentence is generated by relation r.

• For each sentence, a relation indicator r is drawn from the multinomial, and "Words" for the first argument, verb, and second argument are drawn independently from $p_{r,1}$, $p_{r,v}$, $p_{r,2}$, respectively.

Subsequent elaborations of this model add features for the "verb" dependency path and types for the arguments. Given an actual document containing subject-verb-object sentence triples, inference with this model discovers the underlying relations by clustering the sentences.

The difference between this model and an ontologically realistic model is that Rel-LDA posits no underlying world. A trained Rel-LDA model describes what text for a given relation "looks like", but lacks the distinction between world and text and the statistical coupling that a latent world introduces. For example, a model trained on text that lists facts about books purchased on Amazon might learn that Stephen King is the most common first argument for the author relation. If we include a new sentence in the corpus, "Vladimir Vapnik wrote The Nature of Statistical Learning Theory", and then ask the model, "Who wrote The Nature of Statistical Learning Theory?", the answer will be "Stephen King".

Another important consequence of the absence of a latent world is that the Rel-LDA model does not assign high probability to bootstrapping inferences. We speculate that this leads to a learning process that requires many more sentences to reliably discover a set of relations and dictionaries, but we have yet to carry out this experiment. For now, we report only a very preliminary and anecdotal set of results with our model.

4 Preliminary experiments

The model was applied to a small subset (8500 sentences) of an NYT corpus (Yao et al., 2011). The subset was chosen heuristically to have a relatively small number of distinct entity mentions, so as to increase the number of bootstrapping opportunities. The sentences contain two named entities connected by a grammatical dependency path, matching our trivial grammatical model. The inference process discovers (1) roughly 200 relations that underlie the text; (2) the dictionaries describing how each relation is expressed by dependency paths; and (3) the facts belonging to each relation.

We ran smart–dumb/dumb–smart split–merge MCMC (Wang and Russell, 2015) for about 10 minutes and inspected the most likely sampled world. We found that relation 46, which we might call "subsidiary of", emerged with the following highly probable dependency paths in its dictionary:

```
[appos|->unit->prep->of->|pobj]
[appos|-> part->prep->of-> pobj]
[nn| <-unit->prep->of->[pobj]
[partmod|-> own->prep->by->[pobj]
[rcmod|-> own->prep->by-> |pobj]
[appos|-> subsidiary->prep->of-> |pobj]
[rcmod|-> part->prep->of-> |pobj]
[rcmod|-> unit->prep->of ->|pobj]
[poss|<- parent-> |appos]
[appos|-> division->prep->of-> |pobj]
[pobj|<- of<-prep<-office->appos
     ->part->prep->of-> |pobj]
[pobj|<- of<-prep<-unit->appos
     ->part->prep->of-> |pobj]
[nn|<- division->prep->of-> |pobj]
[appos|-> unit-> |nn]
[nsubjpass|<-own->prep->by-> |pobj]
[nn|<- office->prep->of-> |pobj]
```

We also found 60 facts such as $rel_{46}(BBDO\ \ Worldwide, Omnicom\ \ Group)$ and $rel_{46}(Fox, News\ \ Corporation)$. Manual verification of all facts for the most common 20 relations shows roughly 95% precision, with most errors arising from the limitations of the preprocessor and named entity extractor. Because of the strength and frequency of bootstrapping inferences, it seems likely (although this remains to be verified) that the relation discovery process is more accurate than in other approaches and requires less data. Extensions to the generative model to include entity attributes and a more complete semantic grammar (see Section 5) will automatically resolve entity references and avoid the need for pre-parsing or named-entity recognition.

4.1 A note on evaluation

The traditional method of evaluation for IE systems relies on the ability to inspect the extracted knowledge base for correctness. As probability models become more sophisticated, this approach will fail, because (1) the knowledge is represented using internal symbols referring to relations and entities the system discovers for itself, that may not correspond to standard concepts in English, and (2) the meaning of a given internal symbol varies across possible worlds in the inference process. Instead, as with humans, the query interface must, inevitably, be via language itself. For example, one can ask whether the subjects of sentences 14 ("Obama roasts WH press") and 22 ("President seeks second term") are the same entity, or ask for x such that "x is President of South Sudan" is true.

5 Conclusions and further work

An ontologically realistic generative probabilistic model has many advantages for information extraction. Initial experiments show that the approach can discover relations and extract facts in an unsupervised fashion, performing bootstrapping and disambiguation inferences without special heuristics.

Of course, the initial model is vastly oversimplified. We need to add types, generative models for entity mentions, the ability to extract general knowledge (Clark et al., 2014), and a standard upper ontology including mereology, time, actions, and events. (There is no point in forcing a learning algorithm to rediscover these concepts.) A more sophisticated model of reporting bias is needed (Gordon and Durme, 2013). There should be a "backoff hierarchy" of weaker and more general semantic–syntactic models to cope with uninterpretable text; as learning proceeds, new and more specific grammatical forms are added and probability mass moves down the hierarchy. Finally, our purely symbolic semantics can be augmented by a vector-space model of word meaning; this should result in faster generalization via sharing among dictionaries.

Obviously there is much new ground to explore before the proposed approach can handle unrestricted text. Engaging in this exploration seems preferable to trying to extract information about the world from models that deny the world's existence.

Acknowledgments

This research was funded by a Chaire Blaise Pascal awarded to the first author, administered by the Fondation de l'École Normale Supérieure, and by a Senior Chaire d'excellence of the Agence Nationale de la Recherche. The Laboratoire d'Informatique de Paris VI and its Director, Patrick Gallinari, generously hosted the project and its authors.

References

Sergey Brin. 1998. Extracting patterns and relations from the World-Wide Web. In *WebDB Workshop at EDBT-98*.

E. Charniak and R. Goldman. 1992. A Bayesian model of plan recognition. *Artificial Intelligence*, 64:53–79.

Peter Clark, Niranjan Balasubramanian, Sumithra Bhakthavatsalam, Kevin Humphreys, Jesse Kinkead, Ashish Sabharwal, and Oyvind Tafjord. 2014. Automatic construction of inference-supporting knowledge bases. In *AKBC-14*.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *AKBC-13*.

Adam Grycner, Gerhard Weikum, Jay Pujara, James Foulds, and Lise Getoor. 2014. A unified probabilistic approach for semantic clustering of relational phrases. In *AKBC-14*.

Brian Milch and Stuart Russell. 2010. Extending Bayesian networks to the open-universe case. In *Heuristics, Probability and Causality: A Tribute to Judea Pearl*. College Publications.

T. Mitchell, W. Cohen, et al. 2015. Never-ending learning. In *Proc. AAAI-15*.

H. Pasula, B. Marthi, B. Milch, S. J. Russell, and I. Shpitser. 2003. Identity uncertainty and citation matching. In *NIPS-02*.

B. Rink and S. Harabagiu. 2011. A generative model for unsupervised discovery of relations and argument classes from clinical texts. In *EMNLP-11*.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *AKBC-13*.

Wei Wang and Stuart Russell. 2015. A smart-dumb/dumb-smart algorithm for efficient split-merge MCMC. In *UAI-15*.

L. Yao, A. Haghighi, S. Riedel, and A. McCallum. 2011. Structured relation discovery using generative models. In *EMNLP-11*.