Most of the world's *knowledge*, be it factual news, scholarly research, social networks, subjective opinions, or even fictional content, is now easily accessible as digitized text. Unfortunately, due to the unstructured nature of text, much of the useful content in these documents is hidden. There is a need for novel machine learning approaches that can extract meaningful, structured knowledge (such as graphs/networks and databases) from text collections. This structured knowledge facilitates tasks such as question answering (*e.g. facts for virtual agents such as Siri*), easier organization (*e.g. professional details and networks from public bios*), analysis of high-level patterns (*e.g. social networks from public and personal communication*), discovering actionable content (*e.g. business acquisition information from news*), and supporting decision making (*e.g. citation graphs and key results from research papers*). This is my over-arching goal: **designing machine learning algorithms to extract structured information from text**.

In order to perform *information extraction*, mentions of the entities of interest (i.e., *people*, *locations*, *organizations*) need to be identified in the text and disambiguated to real-world entities, followed by recognizing relations between these entities (such as *friendship*, *birthplace*, and *employment*) that are expressed in the text. These tasks pose a number of unique challenges for machine learning: (a) since most machine learning techniques require annotated datasets, human labelers need to read each sentence, annotate it with the entities and relations that appear in the sentence, and construct a final database of all the extracted facts; a nearly impossible task for any reasonably sized dataset, (b) machine learning models used in information extraction are large and densely connected due to the need to aggregate evidence across multiple documents, leading not only to intractable exact inference, but also to inefficient approximations, and (c) these complex models and approximate inference techniques make it difficult for algorithm designers to identify the cause for incorrect predictions, making information extraction models indecipherable and difficult to improve.

My particular research focus has been on addressing the following question:

> **How can I make it easy for users to design and train scalable machine learning algorithms that accurately extract structured information from large collections of text documents?**

There are three important aspects of this goal:

1. **Scalability** to support not only large datasets, but also deep and expressive models; my contributions have focused on novel models and algorithms that utilize parallelism and distributed computing to facilitate scalable learning.
2. **Interactivity** in machine learning to facilitate training with minimal supervision, for which I have introduced training algorithms to directly inject user intuition into the information extractors.
3. **Programmability** of machine learning to facilitate quick prototyping, and easy deployment; I investigated compact, expressive abstractions for many machine learning techniques, and provided efficient *black-box* implementations.

# 1 Scalability: Machine Learning for Large-Scale Information Extraction

With the interest in "big data", new scalable techniques have been proposed to train on massive datasets. Most of these techniques are concerned with training independent classifiers for which inference is trivial. Unfortunately in natural language processing, where we represent entities and relations across the whole corpus, the models tend to be much more connected and complex. These additional dependencies results not only in NP-hard complexity of exact inference (a crucial inner step in machine learning), but polynomial time even for approximate inference, which we cannot afford with data size in the millions. To facilitate efficient machine learning for such models on massive datasets, my doctoral thesis focused on scalable inference algorithms for high-tree width graphical models [Singh, 2014].

**Entity Disambiguation and Linking:** I have been investigating entity-based information extraction tasks such as disambiguation and linking, and exploring novel models and distributed computing for efficient approximate inference. In Singh et al. [2011b], we introduced general purpose distributed sampling for Map-Reduce, and further, proposed additional higher-level entity clusters to not only obtain **accurate disambiguation across 1.5 million mentions** (error reduction of $38\%$ over previous methods), but also enable fruitful data partitions to facilitate efficient inference. To facilitate large-scale research in entity disambiguation, we released the *WikiLinks* **dataset in collaboration with Google** that spans 40 million mentions of 3 million entities, which is more than a hundred times bigger than the largest dataset available at the time [Singh et al., 2012b]. I also expanded upon the research ideas; we extended hierarchies of entities to be arbitrarily deep, resulting in further gains in accuracy while providing **more than 30x speedup** [Wick et al., 2012], and proposed stochastic approximations to sampling (which we call "Monte-Carlo" MCMC) in Singh et al. [2012c] that are **more than ten times faster than existing sampling approaches** on entity disambiguation.

**Joint Modeling:** Extracting information from documents is often divided into a number of sub-tasks: typing, disambiguating, linking, relations, and so on, that are clearly related to each other. Natural modeling of these tasks, unfortunately, results in large, densely-connected structures for which inference is intractable. Most practitioners simplify the models by ignoring these dependencies and constraining the flow of information to be *pipelined* (for

example predicting types of entities before identifying the relations), making it impossible for downstream tasks to inform previous ones. I have instead been exploring ***joint models* over multiple tasks**, such as entity linking and disambiguation [Wick et al., 2013], per-document disambiguation, typing, and relations [Singh et al., 2013a], and cross-corpus segmentation and disambiguation [Singh et al., 2009]. I designed efficient inference techniques for these models that utilize distributed and parallel computing along with model-specific sparsity approximations [Singh et al., 2011a, 2013b], obtaining an **error reduction of 20−75% over pipeline methods**.

**Other ML approaches:** My recent research has focused on extending scalability to further classes of machine learning algorithms. In a recent paper, we explored second-order gradient techniques for structured prediction with gradient-boosted tree potentials [Chen et al., 2015]. I also investigated the capacity of low-dimensional distributed representations to accurately represent large datasets with complex structures [Bouchard et al., 2015]; a data science application of these ideas was awarded the **grand prize in the Yelp Dataset Challenge** [Gupta and Singh, 2015].

## 2 Interactivity: Injecting Domain Knowledge into Extractors

To extract information from unseen documents, machine learning approaches often require a large number of labeled data points that have been annotated with the *true* structures. Such annotations are quite challenging for natural language processing: the underlying tasks require annotation of complex structures, and often need both linguistic expertise and domain knowledge, thus making the labeling process expensive and error-prone. Further, labeling data is also quite wasteful: annotators utilize world knowledge and common sense, and spend significant effort thinking and reasoning about the appropriate label, yet the resulting label communicates very little of the thought process to the machine learning approach. The question I am interested in addressing is: *How can we inject the annotator's knowledge directly into machine learning, in order to utilize the human effort better, and encourage faster learning?*

**Generalized Expectations:** My initial contributions focused on probabilistic graphical models: I explored how annotator-provided *constraints* can be incorporated into complex models. In a collaboration with Yahoo!, I worked on a large-scale corpus of search queries that had to be tagged with the entities contained therein. Since human annotations for such a massive and varied set of queries is out of the question, **we trained a machine learning sequence tagger without *any* labeled data**, instead relying on existing word lists to provide the appropriate signal [Singh et al., 2010a]. However, existing methods for injecting domain knowledge do not suffice for non-trivial models and large datasets since they require exact inference. We proposed a ranking-based objective for injecting knowledge that supports efficient approximate inference [Singh et al., 2010b]. We also introduced an alternative probabilistic formulation that is computationally efficient by approximating the marginals using sampling [Singh et al., 2012a], and used it to obtain $\sim 90\%$ **accuracy using only two simple constraints** on a citation disambiguation benchmark.

**Relation Extraction:** Recently, I have been extending similar ideas to extracting types of relations from text. Obtaining high-quality labeled data for relation extraction is notoriously difficult due to the massive amounts of data and variety in text expressions, and hence machine learning approaches rely on noisy labels, leading to imprecise predictions. On the other hand, relation extraction has historically been associated with user-provided *rule-based* systems that provide precise extractions, however provide poor coverage. My research has focused on methods to combine these two paradigms; we would like to retain the high-precision from the rules, whilst using machine learning algorithms to generalize. We introduced an approach to **embed first-order logic rules into distributed representations** using a tensor-based formulation; this paper received an ***Exceptional Submission Award*** [Rocktaschel et al., 2014]. We extended this work by introducing a generic, model-agnostic formulation, using which we demonstrate that **combining rule-based and machine-learning-based approaches is beneficial for real-world relation extraction** [Rocktaschel et al., 2015], and supports **learning of accurate extractors without any labeled data**.

## 3 Programmability: Democratizing ML via Probabilistic Programming

Automatically extracting information from a large corpus is currently quite an involved process: an appropriate model has to be designed, efficient learning and inference methods have to be selected and implemented for the model, hyper-parameters have to be tuned, and so on. This iterative process not only requires substantial programming effort (and an increased chance of bugs), but also an expertise in machine learning, thus excluding a large fraction of the potential users. As access to data and the need for analysis becomes increasingly commonplace, the next generation of programming languages need to support the tools for data analysis in the language itself. *Probabilistic programming languages* (PPLs) are such higher-level programming languages with operators and constructs that allow users to design complex machine learning models for their task, and with automatic efficient black-box implementations that can train and deploy arbitrary models on real data.

I have worked on probabilistic programming languages that vary considerably in the family of supported models, programming language paradigms, and expected expertise of the target users. I was a core contributor of **Factorie** [McCallum et al., 2009], a PPL that introduces **imperative constructs for expressing immensely complex models** and performs incredibly efficient inference, although it requires basic familiarity with machine learning in order to achieve this. At the other end of the spectrum, I worked on automated construction of the model and generation of inference code given only the *schema* of the data, thus not requiring any machine learning expertise, or even the ability to program, from the user [Singh and Graepel, 2013]. As noted in Gordon et al. [2013], **these ideas directly led to Microsoft developing *Tabular***, a probabilistic programming toolkit for Excel. In an effort to understand the appropriate level of abstraction to make machine learning usable, I have recently been working on **Wolfe** [Riedel et al., 2014, Singh et al., 2015], a PPL that allows the users to provide **a *declarative* description of machine learning** (similar to their mathematical definition), but performs learning and inference automatically. This abstraction is not only intuitive and precise, but more importantly, is universal, i.e., existing, and future, machine learning techniques can be expressed in the formalism.

# Future Research

My contributions so far are first steps in realizing the goal of *easy and accurate information extraction from large corpuses*. As the size of datasets and user base grows, the research agenda I have described above for scalability, interactivity, and programmability will provide an abundance of immediate and important problems to investigate. Furthermore, I am also excited about pursuing novel areas of research that are becoming increasingly important as the field matures.

## Interpreting and Debugging Machine Learning

Machine learning approaches, when successfully trained, produce incredibly impressive extractions that provide a convincing illusion of intelligence. However, as is unfortunately the case more often, machine learning algorithms fail and make errors that are quite obvious and blatant for humans, and further, the approaches are unable to explain why the errors were made. Users are then resorted to labeling more data in order to fix the errors, an exercise that is both expensive and error-prone for natural language processing, but further, is a poor utilization of the annotator's time.

I am interested in moving towards ***teachable* machine learning**, where the learning process is more interactive and dialog-like between the user (teacher) and the machine learning approach (student). Two crucial steps are needed in order to achieve this interactive exchange of information.

1. We would like to extract the model's explanations for why an extraction is made, such that the explanations are easy to understand by the user, faithful to the model, and agnostic to the mechanics of the algorithm being used.
2. Given the explanation for why a prediction is made, the user will provide feedback on the explanation that needs to be incorporated into the machine learning model.

This interactive exchange of information between users and the model has the potential to make machine learning easy to train, and thus be quickly deployed for any task of interest.

## Beyond Facts: Extraction of Non-Canonical Information from Multi-Modal Data

Even though information extraction has been studied for a long time, and has been successfully deployed in a plethora of applications, the task definitions have evolved surprisingly little over the years. The output of information extraction has been primarily constrained to a pre-defined, fixed schema with pairwise relations, and thus is unable to adequately provide a complete understanding of the text. Similarly, extractions are most commonly restricted to the factual content, which is a very shallow representation of the text as compared to the rich and deeply intricate knowledge a human is able to amass when reading. Further, information extraction currently restricts itself exclusively to text, however relevant information is often spread across audio, videos, and photos.

There are a number of important ways I am interested in to broaden the existing ideas regarding information extraction.

1. Extracting information about entities that are not named entities, for example concepts such as events (*Boston bombing, hurricane, etc.*), common nouns (*politician, apples, etc.*), issues (*health insurance, abortion, etc.*), and ideologies (*sexism, libertarianism, etc.*).
2. Extending the set of relations between entities to capture various subjective relations and sentiments, including notions of *influence*, *acquaintance*, *inspiration*, *mentoring*, *like/dislike*, and so on.
3. Exploring temporal aspects of information extraction that have not received adequate attention, and combining existing models of time-series with textual extraction systems.
4. Inspired by the recent advances in joint deep models of images and text, investigating similar models of cross-task distributed representations for information extraction, eventually extending them to audio and video.

# References


Guillaume Bouchard, Sameer Singh, and Theo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, 2015.

Tianqi Chen, Sameer Singh, Ben Taskar, and Carlos Guestrin. Efficient second-order gradient boosting for conditional random fields. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Andrew D. Gordon, Thore Graepel, Nicolas Rolland, Claudio Russo, Johannes Borgstrom, and John Guiver. Tabular: A schema-driven probabilistic programming language. Technical Report MSR-TR-2013-118, 2013.

Nitish Gupta and Sameer Singh. Collective factorization for relational data: An evaluation on the yelp datasets. Technical report, Yelp Dataset Challenge, Round 4, 2015.

Andrew McCallum, Karl Schultz, and Sameer Singh. Factorie: Probabilistic programming via imperatively defined factor graphs. In *Neural Information Processing Systems (NIPS)*, 2009.

Sebastian Riedel, Sameer Singh, Vivek Srikumar, Tim Rocktaschel, Larysa Visengeriyeva, and Jan Noessner. Wolfe: Strength reduction and approximate programming for probabilistic programming. In *International Workshop on Statistical Relational AI (StarAI)*, 2014.

Tim Rocktaschel, Sameer Singh, Matko Bosnjak, and Sebastian Riedel. Low-dimensional embeddings of logic. In *ACL 2014 Workshop on Semantic Parsing (SP14)*, 2014.

Tim Rocktaschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

Sameer Singh. *Scaling MCMC Inference and Belief Propagation for Large, Dense Graphical Models*. PhD thesis, University of Massachusetts, 2014.

Sameer Singh and Thore Graepel. Automated probabilistic modeling for relational data. In *ACM Conference of Information and Knowledge Management (CIKM)*, 2013.

Sameer Singh, Karl Schultz, and Andrew McCallum. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *Machine Learning and Knowledge Discovery in Databases (Lecture Notes in Computer Science) and European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2009.

Sameer Singh, Dustin Hillard, and Chris Leggetter. Minimally-supervised extraction of entities from text advertisements. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010a.

Sameer Singh, Limin Yao, Sebastian Riedel, and Andrew McCallum. Constraint-driven rank-based learning for information extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010b.

Sameer Singh, Brian Martin, and Andrew McCallum. Inducing value sparsity for parallel inference in tree-shaped models. In *Neural Information Processing Systems (NIPS) Workshop on Computational Trade-offs in Statistical Learning (COST)*, 2011a.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Association for Computational Linguistics (ACL)*, 2011b.

Sameer Singh, Gregory Druck, and Andrew McCallum. Constraint-driven training of complex models using mcmc. Technical report, University of Massachusetts Amherst, CMPSCI UM-CS-2012-032, 2012a.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia. Technical report, University of Massachusetts Amherst, CMPSCI UM-CS-2012-015, 2012b.

Sameer Singh, Michael Wick, and Andrew McCallum. Monte carlo mcmc: Efficient inference by approximate sampling. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2012c.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. Joint inference of entities, relations, and coreference. In *CIKM Workshop on Automated Knowledge Base Construction (AKBC)*, 2013a.

Sameer Singh, Sebastian Riedel, and Andrew McCallum. Anytime belief propagation using sparse domains. In *Neural Information Processing Systems (NIPS) Workshop on Resource Efficient Machine Learning*, 2013b.

Sameer Singh, Tim Rocktaschel, Luke Hewitt, Jason Naradowsky, and Sebastian Riedel. WOLFE: An NLP-friendly declarative machine learning stack. In *Demo at the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

Michael Wick, Sameer Singh, and Andrew McCallum. A discriminative hierarchical model for fast coreference at large scale. In *Association for Computational Linguistics (ACL)*, 2012.

Michael Wick, Sameer Singh, Harshal Pandya, and Andrew McCallum. A joint model for discovering and linking entities. In *CIKM Workshop on Automated Knowledge Base Construction (AKBC)*, 2013.