Data Catalysis: Facilitating Large-Scale Natural Language Data Processing

Patrick Pantel

USC Information Sciences Institute, Marina del Rey, CA, 90292, USA
pantel@isi.edu

Large-scale data processing of the kind performed at companies like Google is within grasp of the academic community. The potential benefits to researchers and society at large are enormous. In this article, we present the Data Catalysis Center, whose mission is to stage and enable fast development and processing of large-scale data processing experiments. Our prototype environment serves as a pilot demonstration towards a vision to build the tools and processing infrastructure that can eventually provide level access to very large-scale data processing for academic researchers around the country. Within this context, we describe a large scale extraction task for discovering the admissible arguments of automatically generated inference rules.

1. Introduction

In many sectors, it has become apparent that experiments over large data can enable better science: we are witnessing a growing number of scientific disciplines that are extending their traditional approaches of theoretical and observational research to include large data computer experiments. Large data experimentation requires not only considerable computational infrastructure, but also sophisticated toolsets and large research/support teams to handle the added complexities of the large scale. Consequently, we are seeing an ever-widening gap in large data experimentation capabilities between the larger and the smaller academic institutions. The effects on science are of important consequence:

- *Slowed progress towards scientific breakthroughs*: advances in developing large-scale solutions to problems are hindered by the limited number of researchers that have enabling capabilities;
- *Widening competitive disadvantage for smaller institutions*: lack of enabling capabilities at small institutions makes it difficult to attract the best researchers and students wishing to work on large-scale data problems; and
- *Incomparable scientific results*: researchers at smaller institutions are unable to compare their theories with state of the art solutions requiring large data processing.

There have been significant investments by institutional and governmental organizations to build high performance open computing resources and supporting infrastructure. Making use of this infrastructure, however, has been difficult for many computer science researchers because of the overhead required for even the simplest tasks – an example of which we describe in the next section. In response, we propose a *Data Catalysis* initiative which is a large, cross-cutting, focused effort to build large data processing capabilities over this existing infrastructure and opening it to the entire research community. This vision is not limited to the NLP community, but is cross-cutting in that it may facilitate large data experiments in many scientific disciplines. Realizing the vision requires a consortium of projects for studying the functional architecture, building low-level infrastructure and middleware, engaging the research community to participate in the initiative, and building up a library of open-source large data processing algorithms.

2. The Challenge of Large Data Processing

The natural language processing (NLP) community is a prime example of a community in need of level access to large data processing capabilities. More and more cutting edge NLP research incorporates extensive analysis over very large textual resources like the Web. For example, the current state of the art machine translation system developed at Google relies on statistics over 8×10^{12} words on the Web (Och 2006); and state of the art information extraction efforts seek to scale to all documents on the Web (Banko et al. 2007; Paşca et al. 2006). This research is enabled by sophisticated custom-built tools and processing environments requiring years of development time. Such environments are often feasible only for large teams with sustained research programs. Google's focused investment on data processing middleware, for example, has enabled their researchers to rapidly develop data experiments at an unparalleled scale by abstracting the underlying complexities of specifying and managing distributed computation.

Lacking this middleware, even if given access to computational resource allocations and low-level tools, academic groups have reported difficulty carrying out even the most conceptually simple large data experiments, resulting in limited-scale studies and long experimental cycles.

Consider, for example, the conceptually simple task of counting the frequencies of all words in a very large collection of textual data. This task is prototypical of the many large data processing tasks encountered in NLP, all characterized by:

- Large *partitionable data* which can be distributed over a cluster of compute nodes; and
- *Aggregation of intermediate results* generated by the distributed computation to form the final output.

An efficient solution requires: i) the distribution of the text to several computational nodes, ii) parallel accumulation of local frequency counts, and iii) aggregation of the results over all nodes. The researcher must manage the distribution to optimize bandwidth utilization, monitor the parallel accumulation of frequencies to handle straggling processes and memory overruns, and synchronize the nodes and optimize bandwidth utilization for the aggregation of the results.

The *Data Catalysis* paradigm aims to facilitate the development time and processing time of exactly this class of problems by enabling for the academic community the MapReduce technology pioneered at Google (Dean and Ghemawat 2004; Ghemawat et al. 2003). This framework significantly simplifies the development effort required on the part of the researcher, allowing her to think at a conceptual level by abstracting away the underlying processing complexities. For the above task of counting word frequencies, the researcher needs only to define the following two functions:

```
map(String key, String value):
// key: document name
// value: document contents
   for each word w in value:
          AssertIntermediate(w, "1");
```

```
reduce(String key, Iterator values):
// key: a word
// values: a list of counts
   int result = 0;
   for each v in values:
          result += ParseInt(v);
   Assert(result);
```

The functions are submitted to a Large Data Processing Abstraction Layer (LDPAL), which transparently handles the parallelization. The *map* function is applied to each document in the textual collection (the underlying infrastructure manages the parallelization) and the *reduce* function is used to combine intermediate results from *map* to form the final output[1]. A distributed file system, which stores the large data collection, allows the LDPAL to greatly reduce network bandwidth requirements by moving the computation of the *map* function to the data instead of moving the data to the machines. Network bandwidth is mostly only used during aggregation when the LDPAL consolidates matching intermediate results from the *map* invocations and moves them to compute nodes ready for *reduce* invocations.

We realize the great number of technologies and research efforts aimed at simplifying large-data processing for both grid and cluster environments. The key differentiating features of the *MapReduce* approach are:

- Algorithms may be expressed in an easy to grasp special-purpose language;
- LDPAL layer which provides an API abstracting away the complexities of large-scale computation and providing rapid and robust aggregation of results;
- Accelerated access and processing over data by preparing data and keeping it distributed over a large cluster of compute nodes.

3. Data Catalysis: A Vision for the Functional Architecture

The data catalysis vision is not limited to the natural language processing community, but is cross-cutting in that it may facilitate large data experiments in many computer science disciplines. Realizing it is a large effort requiring a consortium of projects for studying the functional architecture, building low-level infrastructure and middleware, engaging the research community to participate in the initiative, and building up a library of open-source large data processing algorithms. Focusing on a vertical such as NLP may, however, serve as a small demonstration of the potential impact of the data catalysis initiative, helping us gain insight into the problems that will need to be

[1] The *MapReduce* implementation of the LDPAL is attributed to Google for accelerating development cycles. Our goal is to bring this capability to the research community by leveraging existing open source resources.

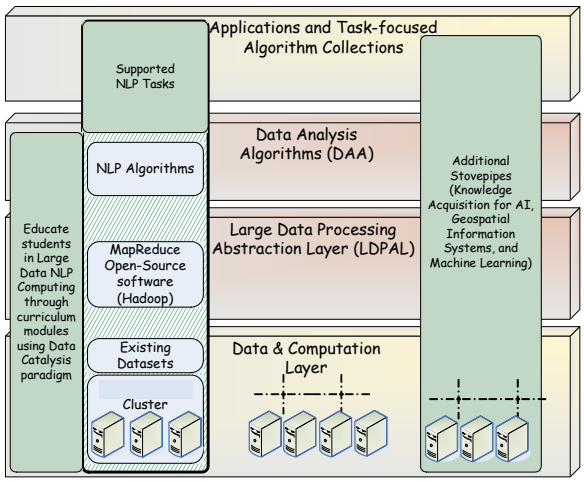

Figure 1. *Data Catalysis* functional architecture.

addressed in order to realize the overarching vision of large-scale data catalysis for the academic community. Below is a high-level description of our envisioned functional architecture (see Figure 1) as it stands today.

Data & Computation Layer, the foundational layer comprised of high performance computing infrastructure, potentially harnessed into large grids, and large data storage and management services.

Large Data Processing Abstraction Layer (LDPAL), the central layer providing a programming API allowing simple algorithm development by abstracting the complexities of low-level distributed computation. This layer handles the complex management of the computational processes when these algorithms are executed.

Data Analysis Algorithms (DAA), the layer consisting of a variety of algorithms leveraging LDPAL. The MapReduce implementation within the LDPAL potentially supports cross-cutting large-scale experiments in a variety of areas impacting computer science, social sciences, biology, and beyond. Algorithms already known to map into the framework stem from machine learning, graph computations, information extraction, and data mining (Dean and Ghemawat 2004).

Applications and Task-focused Algorithm Collections, the layer interconnecting specialized DAA algorithms in support of applications and research tasks, potentially managed using workflow technologies.

The strengths of the proposed *Data Catalysis* approach are its demonstrated success in industry at very large scale and its broad applicability (with the full scope not yet explored). Thousands of algorithms have already been developed at Google within this framework (Dean 2006) enabling very fast development cycles and processing speeds.

4. Enabling Data Catalysis for the Natural Language Processing Community

Realizing the vision of an open access data catalysis environment for the research community is an ambitious long-term project, requiring the development of the functional architecture and its deployment and integration with computational infrastructures such as high performance computing centers and the TeraGrid, protocols and infrastructure for data collection and management, libraries of MapReduce algorithms, and as importantly community awareness campaigns to engage the community within this paradigm.

Initial strides should leverage existing infrastructure, open-source software, and existing data resources to build a proof-of-concept prototype of the complete functional architecture:

- **Data & Computation Layer** – Here, we make use of existing HPCC and clustered computers and leverage large textual datasets available to the community. Candidates include the Spirit 1TB web collection, Trec's Gov2 0.5TB web collection, a snapshot of Wikipedia, Project Gutenberg collection, and offerings available through LDC such as the Gigaword corpus or the Google Web 1T 5-gram dataset.
- **LDPAL Middleware** – We have deployed Hadoop [2], the open-source software package implementing Google's LDPAL MapReduce and distributed file system framework. Hadoop has been shown to scale to several hundred machines, allows users to write "map" and "reduce" code, and manages the sophisticated parallel execution of the code.

Candidate NLP tasks suitable for MapReduce include building a language model (a key component of machine translation systems, question answering systems, and natural language generation engines), machine learning for text classification (e.g., spam email classification and sentiment classification in online product reviews), and building a thesaurus of conceptually similar expressions (useful in question answering, textual entailment, and information extraction systems).

5. Inferential Selectional Preferences

Semantic inference is a key component for

[2] Hadoop, http://lucene.apache.org/hadoop/

advanced natural language understanding. Several important applications are already relying heavily on inference, including question answering (Harabagiu and Hickl 2006), information extraction (Romano et al. 2006), and textual entailment (Szpektor et al. 2004).

In response, several researchers have created resources for enabling semantic inference. Among manual resources used for this task are WordNet (Fellbaum 1998) and Cyc (Lenat 1995). Although important and useful, these resources primarily contain *prescriptive* inference rules such as "*X divorces Y* ⇒ *X married Y*". In practical NLP applications, however, *plausible* inference rules such as "*X* married *Y*" ⇒ "*X* dated *Y*" are very useful. This, along with the difficulty and labor-intensiveness of generating exhaustive lists of rules, has led researchers to focus on automatic methods for building inference resources such as inference rule collections (Lin and Pantel 2001; Szpektor et al. 2004) and paraphrase collections (Barzilay and McKeown 2001).

Using these resources in applications has been hindered by the large amount of incorrect inferences they generate, either because of altogether incorrect rules or because of blind application of plausible rules without considering the context of the relations or the senses of the words. For example, consider the following sentence:

Terry Nichols was charged by federal prosecutors for murder and conspiracy in the Oklahoma City bombing.

and an inference rule such as:

X is charged by Y ⇒ *Y announced the arrest of X* (1)

Using this rule, we can infer that "*federal prosecutors* announced the arrest of *Terry Nichols"*. However, given the sentence:

Fraud was suspected when accounts were charged by CCM telemarketers without obtaining consumer authorization.

the plausible inference rule (1) would incorrectly infer that "*CCM telemarketers* announced the arrest of *accounts*".

This example depicts a major obstacle to the effective use of automatically learned inference rules. What is missing is knowledge about the admissible argument values for which an inference rule holds, which we call *Inferential Selectional Preferences*. For example, inference rule (1) should only be applied if *X* is a *Person* and *Y* is a *Law Enforcement Agent* or a *Law Enforcement Agency*. This knowledge does not guarantee that the inference rule will hold, but, as we show in this paper, goes a long way toward filtering out erroneous applications of rules.

In a recently published article (Pantel et al. 2007), we proposed *ISP*, a collection of methods for learning inferential selectional preferences and filtering out incorrect inferences. The described algorithms apply to any collection of inference rules between binary semantic relations, such as example (1). *ISP* derives inferential selectional preferences by aggregating statistics of inference rule instantiations over a large corpus of text. Within *ISP*, we explored different probabilistic models of selectional preference to accept or reject specific inferences. We showed empirical evidence that ISP's can be automatically learned and used for effectively filtering out incorrect inferences generated using the DIRT resource (Lin and Pantel 2001).

Extracting ISP's for all 12 million DIRT inference rules is a challenging task which fits very well the Data Catalysis paradigm. A very simple program allowed us to extract selectional preferences for all DIRT inference rules in a single day of effort. A demo of the resulting ISP's can be found at http://www.patrickpantel.com/demos.htm.

References

Barzilay, R.; and McKeown, K.R. 2001.Extracting Paraphrases from a Parallel Corpus. In *Proceedings of ACL 2001*. pp. 50–57. Toulose, France.

Banko, M.; Cafarella, M.J.; Soderland, S.; Broadhead, M.; Etzioni, O. 2007. Open Information Extraction from the Web. To appear in *Proceedings in IJCAI-07*. Hyderabad, India.

Dean, J. 2006. Experiences with MapReduce, an Abstraction for Large-Scale Computation. In *Proceedings of Parallel Architectures and Compilation Techniques*. Seattle, WA.

Dean, J. and Ghemawat, S. 2004. MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of OSDI'04: Sixth Symposium on Operating System Design and Implementation*. San Francisco, CA.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Ghemawat, S.; Gobioff, H.; and Leung, S-T. 2003. The Google File System. In *Proceedings of SOSP'03*. New York, NY.

Harabagiu, S.; and Hickl, A. 2006. Methods for Using Textual Entailment in Open-Domain Question Answering. In *Proceedings of ACL 2006*. pp. 905-912. Sydney, Australia.

Lenat, D. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Lin, D. and Pantel, P. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering* 7(4):343-360.

Och, F.J. 2006. Oral Presentation: The Google Machine Translation System. NIST 2006 Machine Translation Workshop. Washington, D.C.

Pantel, P; Bhagat, R.; Coppola, B.; Chklovski, T.; and Hovy, E.H. 2007. ISP: Learning Inferential Selectional Preferences. In *Proceedings of NAACL HLT 07*. pp. 564-571. Rochester, NY.

Paşca, M.; Lin, D.; Bigham, J.; Lifchits, A.; and Jain, A. 2006. Organizing and Searching the World Wide Web of Facts - Step One: The One-Million Fact Extraction Challenge. In *Proceedings of AAAI-06*. pp. 1400–1405. Boston, MA.

Romano, L.; Kouylekov, M.; Szpektor, I.; Dagan, I.; Lavelli, A. 2006. Investigating a Generic Paraphrase-Based Approach for Relation Extraction. In *EACL-2006*. pp. 409-416. Trento, Italy.

Szpektor, I.; Tanev, H.; Dagan, I.; and Coppola, B. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP 2004*. pp. 41-48. Barcelona,Spain.