Ubiquity Symposium

Big Data

Developing an Open Source “Big Data” Cognitive Computing Platform

By Michael Kowolenko and Mladen A. Vouk

Editor’s Introduction

The ability to leverage diverse data types requires a robust and dynamic approach to systems design. The needs of a data scientist are as varied as the questions being explored. Compute systems have focused on the management and analysis of structured data as the driving force of analytics in business. As open source platforms have evolved, the ability to apply compute to unstructured information has exposed an array of platforms and tools available to the business and technical community. We have developed a platform that meets the needs of the analytics user requirements of both structured and unstructured data. This analytics workbench is based on acquisition, transformation, and analysis using open source tools such as Nutch, Tika, Elastic, Python, PostgreSQL, and Django to implement a cognitive compute environment that can handle widely diverse data, and can leverage the ever-expanding capabilities of infrastructure in order to provide intelligence augmentation.

Ubiquity Symposium

Big Data

Developing an Open Source “Big Data” Cognitive Computing Platform

By Michael Kowolenko and Mladen A. Vouk

There are five “Vs” in big data: velocity, volume, variety, veracity, and value [1]—the last being the most important. Big data has no meaning unless it offers a value that otherwise would not be there (see Figure 1). Can it help you answer questions that could not be answered before? Can it help you make better decisions? Can you make decisions faster? For this to happen big data need to be processed and analyzed in an intelligent way. Another challenge is to offer access to and analysis of such data without compromising privacy, security, and safety. The computing platform we outline in this article provides all the necessary elements needed to go from big data to something of value—speed of processing; an ability to handle data volumes fast; an ability to classify and to manage a large variety of data types; an ability to insure privacy and security by supporting isolation, compute-to-data, data-to-compute, and open models; and an ability to iteratively verify and validate data necessity, sufficiency, and veracity, and then offer enhanced decision-making services. It all starts by asking (and understanding) the right questions.

Figure 1. The Five Vs of Big Data

The Questions

The grand challenge presented to, and addressed by, the IBM Watson project demonstrated the feasibility of co-processing, using machine learning techniques, structured and unstructured data in order to answer questions posed while playing the television game show Jeopardy [2-6]. This complex task of question disambiguation, fact identification, and analysis provided the general public with a concrete example of how machines can be used to answer fact-based questions. However, the development of these analytical tools can be daunting given the number of software platforms, infrastructure requirements, domain flexibility that can be explored, the type of question posed, and the time and diversity of data required to generate a system of sufficient accuracy and veracity to provide relevant facts necessary for decision-making. To address this problem the Institute of Next Generation Computer Systems (ITng) and the Computer Science Department at North Carolina State University (NCSU) developed a business process coupled with open-source software to generate an adaptive cognitive

computing platform capable of fact extraction and result scoring that is used by students, faculty, and external partners to explore the value of cognitive computing.

The fundamental concept in developing this open-source platform is the system must be capable of ingesting a broad range of common forms of information, perform a progressively more sophisticated filtering, and then offer analytics resulting in user-specific consolidation of relevant information necessary for the user to make an informed decision (see Figure 2). Functionally, the cognitive platform is needed to convert information to knowledge while optimizing infrastructure. User input is needed for analytics models so that the results are relevant. This is accomplished by having several tiers of filters leading to data reduction. Crude filters are followed by in depth filtering and analytics based on the user input. Scores and results are displayed to the user.

Figure 2. Design Concept

The Problem Statement

To develop a successful cognitive compute platform requires a reduction of the decision-making process into a series of compute activities. Prior to the development of selective data

filters, it is necessary to develop a process for interviewing subject matter experts (SMEs) and understanding how these SMEs deconstruct ambiguous statements to a series of fact-based (often domain specific) questions. In our approach, the steps/questions in the decision-making process (DMP) are:

1. **What** data do we have or need?
2. **Classification**: What is the problem type? Strategic? Tactical? What are the facts?
3. **Alternatives**: What else could be done? What are the reasons the alternatives can't be done?
4. **Decision**: Based on objective judgment criteria, what is/are the action(s) taken?
5. **Implementation**: What is the tactical execution stemming from/leading to the decision?
6. **Assessment**: Are metrics in place to determine if outcome is successful?

This DMP is illustrated in Figure 3.

Figure 3. Illustration of the Decision-making Process

The process of question disambiguation is the most challenging part. Humans form mental models and develop “shortcuts” rather than follow a linear process for sense-making [7, 8]. Focusing on the development of fact-based statements allows us to create robust filtering processes and fact extraction algorithms. Understanding the context of the analytics that must be performed enhances the capability of returning valid data.

The process of coupling a critical thinking process with both quantitative, and more importantly, qualitative information has proven to be very effective in several proof of concept experiments we have been performed with various business verticals.

Selection and Implementation of the Software Stack

One important software engineering lesson learned long time ago [9] is software used to support decision-making must be capable of matching (mimicking or molding to) the actual decision process used by humans. “Grinding” between software and the actual processes they support is a sure path to delays and incorrect decisions. This matchmaking begins with the collection of information. The collection process requires correct decisions be made regarding the data types and data sources in play. The acquisition system must have the flexibility of handling a wide array of inputs and data types.

The general data-flow architecture of our approach is illustrated in Figure 4. Inputs (data sources) can be structured and unstructured. Once collected, data are classified, indexed and stored for domain- and user-specific analytics.

Figure 4. Platform Data-flow Architecture

When building our cognitive compute engine we used open-source codes. The pilot system we developed runs on Ubuntu (currently v16.04, [10]) as the base Linux operating system. This provides a robust platform for installation of the core databases—PostgreSQL (or Postgres) [11] and Apache Cassandra [12].

Postgres serves two centralized roles. One is that of a general relational store for data acquired from external sources, such as data.gov. The second is to provide home for metadata and for analysis scores. Cassandra, on the other hand, is a distributed store that offers ease of expansion, and a friendly and rich query language. Its filters help us reduce and classify unstructured text data sets. As the complexity of the annotations increases, the challenge is balancing hardware versus software performance. With Cassandra, less restrictive filters are applied to the data followed, by more detailed analysis using subsequent analytical methods and tooling. For example, crawls of news sites can be searched for key words pertaining to the topic under investigation while more extensive analytics are performed on the search returns.

Open data ingest is performed in one of three ways. One is via a general web crawler. We selected Nutch [13] as the crawler because of its flexibility in configuration, its ability to understand and convert a wide variety of file types, and the fact that Gora [14] allows it to be easily joined to Cassandra and indexing engine Solr-Lucene [15, 16].

For efficiency, a methodology was developed for collecting information leveraging search engines that may already exist. For example, if the website has a key word search tool, files are downloaded based on that URL ontology. Once URL ontology is identified, cURL (Linux) commands are used to download the necessary files. These files are then processed using either Tika [17] or BeautifulSoup [18]. Our toolkit component for data ingestion and wrangling is listed in Table 1.

Table 1. Data Ingestion and Wrangling Components

Component	Purpose	Source URL
BeautifulSoup	Html conversion	https://www.crummy.com/software/BeautifulSoup/
Tf-idf	Stop words	http://www.tfidf.com/
csvkit	CSV conversion	https://csvkit.readthedocs.io/en/1.0.1/
pdfminer	PDF conversion	https://github.com/euske/pdfminer
Tesseract	Ocr conversion	https://github.com/tesseract-ocr
Tika	File conversion	https://tika.apache.org/
Nutch	Web Crawls	http://nutch.apache.org/

A human-facing analytical workbench requires a robust set of tools to handle diverse skill sets data scientists and analysts may have and/or need, as well as to handle queries that could be posed to the system. The workbench needs to process both structured and unstructured information, allow the building of metadata tables of structured domain specific facts with the concomitant analytical processes using, for example, machine learning or multi-criteria decision- making. The table of metadata is then accessed via a web portal.

There is a wide array of analytical tools in a variety of languages is available. For example, we use Python [19-22] or *R*-based [23] tools for many analytical applications that make use of matrix algebra. The greater challenge is in determining how to perform unstructured text analytics. Here, we found the use of the Unstructured Information Management Architecture (UIMA) [24] to be the most flexible system for the development of annotators. This open source Eclipse workbench allows for the generation of parsers and annotators that can be used with Solr [24]. Much of the functionality found in UIMA is present in NLTK [25], however, the ability to quickly configure the annotator led to its predominate use of UIMA in this system. We have explored other open-source annotation systems, such as BRAT [26], in the context of developing machine learning classification models. Interestingly, the latter have been met with user resistance. SMEs find the task of labeling tedious. Rather, the use of domain specific dictionaries combined with rules generated in the UIMA system provides the specificity and context needed in a text extraction system without frustration.

The indexing and presentation of text was performed using Solr-Lucene [14,15]. Nutch, was selected because it is highly configurable and has good integration properties. The challenge is increasing the speed of annotation. We are exploring the use of GPUs as a possible solution to the indexing bottleneck. We have built, but not yet fully tested Gremlin-based graph database [27]. Preliminary results are promising.

Machine Learning

Integration of machine learning algorithms, like everything else in the cognitive engine, is based on the query request. The use of classification systems has been helpful when validating rules-based systems used by UIMA. Underlying activities in text analytics allow for the development of a series of tools for clustering and classification, such as n -gram analysis, vector mapping, etc. [18, 19, 20, 21]. The development of word relationships by interviewing the SMEs leads to efficient use of machine time. Bias can be addressed by running naïve clustering algorithms and comparing that to supervised systems [28].

Generally, having a series of decision-tree algorithms has been found useful when assessing facts associated with multi-criteria decision-making. When dealing with business related decision-making, the technique of Order of Preference by Similarity to Ideal Solution (TOPSIS) [29] was deployed. Also of interest to us is the use of GPU compute platforms. We have recently begun to explore the use of Tensorflow [21] and its GPU deep learning library.

Table 2. Components for Analytics

Component	Purpose	Source URL
NLTK	Text analytics	http://www.nltk.org/
Scikit	Sci. - biology	http://scikit-learn.org/stable/index.html
R	Global statistics	https://www.r-project.org/about.html
Tensorflow	Machine learning	https://www.tensorflow.org/
UIMA	Text analytics	https://uima.apache.org/
Solr	Search	http://lucene.apache.org/solr/
Lucene	Index	http://lucene.apache.org/core/
Gremlin	Graph analytics	https://github.com/tinkerpop/gremlin/
UIMA	Text analytics	https://uima.apache.org/

There is redundancy in the packages we deployed in our cognitive environment. Rather than focus on efficiency in this aspect of the platform, the goal was to provide flexibility to the programmers and data scientists who would use the system.

User Interface

Most end users of a cognitive platform seek unambiguous answers to their question. Overexposure to the wide array of information and analytics used to derive the answer are often met with confusion. To overcome this problem, we developed a simple web-based interface (illustrated in Figure 5 for an application called “CEO Pay Evaluator”) that allows the user to query the metadata present in the structured database of facts related to the domain of concern. This platform is based on Django and is referred to as a field-based return system. A typical query return consists of a union of the facts and analytics necessary to answer the question posed.

The image shows a web application window titled "C.E.O. Pay Rater". Inside the window, there are four dropdown menus on the left, each with a downward arrow icon. The labels for these menus are "Industry", "Compensation", "Performance", and "Peers". To the right of each dropdown menu is a horizontal slider with a small house-shaped cursor. Below these controls is a large rectangular box labeled "Results:".

Figure 5. User Interface for “CEO Pay Evaluator”

The interface can be configured with a number of filters based on user input. These filters are used to further refine the conditions of analysis performed on the metadata.

We have added geolocation capabilities to the system with the inclusion of PostGIS as a metadata reference for exposing information on the user Web portal.

Table 3. User Interface Tools

Component	Purpose	Source URL
Django	Web interface	https://www.djangoproject.com/
D3	graphics	https://d3js.org/
Scipy	graphics	http://scikit-learn.org/stable/index.html
Gephi	Graph data	https://gephi.org/

Table 4. Structured Data Store – Postgres packages

Component	Purpose	Source URL
Sqlalchemy	Python-db interface	http://www.sqlalchemy.org/
Psycopg	Python-db interface	http://initd.org/psycopg/
PostGIS	geolocation	http://www.postgis.net/

Pilots

The platform we describe has been tested in a number of use cases in multiple verticals. For example, in collaboration with pharmaceutical company partners it was successfully used to investigate markets and regulatory compliance issues. In collaboration with government agencies the platform was assessed in the context of security and regulatory compliance situations, such as might occur in financial industry.

Because of its flexibility and range, the platform has proven to be particularly useful in training computer science students in data-driven decision-making. Students are given assignments that focus on developing interactive big data applications that solve real-world issues. Projects have ranged from an application that could adjust for shifts in political power in the Middle East to determining the appropriate compensation for corporate executives. Further improvements to and development of the platform continues.

Summary

The ability to leverage diverse data types to help make trustworthy decisions requires a robust and dynamic approach and a flexible support system. The needs of a data scientist are diverse and are as varied as the questions being explored. A system needed to support these activities must be as dynamic as the analytical environment requires. This may challenge the formulation of user requirements. However, design of a system that can support such needs can be approached in a step-wise fashion. That way the requirements become more manageable. By developing an analytics workbench based on acquisition, transformation, and analysis, one can develop a customized open-source cognitive compute environment that can handle widely

diverse data, and can leverage the ever-expanding capabilities of infrastructure in order to provide intelligence augmentation.

References

- [1] <http://www.ibmbigdatahub.com/infographic/four-vs-big-data>
- [2] Ferrucci, D.A. et al. Watson: beyond Jeopardy! *Artificial Intelligence* 199–200, June–July (2013), 93–105.
- [3] Ferrucci, D. A. Introduction to “This is Watson.” *IBM Journal of Research and Development* 56, 3/4 (2012).
- [4] McCord, M. C., Murdock, J. W., and Boguraev, B. K. Deep parsing in Watson. *IBM Journal of Research and Development* 56, 3/4 (2012).
- [5] Chu-Carroll, J., Fan, J., Schlaefel, N., and Zadrozny, W. Textual resource acquisition and engineering. *IBM Journal of Research and Development* 56, 3/4 (2012).
- [6] Fan, J., Kalyanpur, A., Gondek, D. C., and Ferrucci, D. A. Automatic knowledge extraction from documents. *IBM Journal of Research and Development* 56, 3/4 (2012).
- [7] Reeves, W.W. *Cognition and Complexity: The Cognitive Science of Managing Complexity*. Scarecrow, Lanham, MD, 1996.
- [8] Jonassen, D.H. Toward a design theory of problem solving. *Educational Technology Research and Development* 48, 4 (01/2000).
- [9] http://www.sei.cmu.edu/productlines/frame_report/process_def.htm
- [10] <http://releases.ubuntu.com/16.04/>
- [11] <https://www.postgresql.org/>
- [12] <http://cassandra.apache.org>
- [13] <http://nutch.apache.org/>
- [14] <http://gora.apache.org>
- [15] <http://lucene.apache.org/solr/>

- [16] <http://lucene.apache.org/core/>
- [17] <https://tika.apache.org/>
- [18] <https://www.crummy.com/software/BeautifulSoup/>
- [19] <https://www.scipy.org/>
- [20] <http://pandas.pydata.org/>
- [21] <https://www.tensorflow.org/>
- [22] <http://scikit-learn.org/stable/index.html>
- [23] <https://www.r-project.org/about.html>
- [24] <https://uima.apache.org/>
- [25] <http://www.nltk.org/>
- [26] <http://brat.nlplab.org/>
- [27] <https://github.com/tinkerpops/gremlin/>
- [28] Small, S.G., and Medsker. Review of information extraction technologies and applications. *Neural Computing and Applications* 25, 3 (09/2014).
- [29] Hwang, C.L., and Yoon, K.P. *Multiple Attribute Decision Making: Methods and applications*. Springer-Verlag, New York, 1981.

Biographies

Dr. Michael Kowolenko is the Managing Director of the Institute of Next Generation Computing and Industry Fellow in the Center of Innovation Management Studies; Research Professor in the Department of Computer Science at North Carolina State University. His research and teaching activities focus on models of integrating data analytics in the area of critical thinking and decision-making. Prior to joining NCSU, Dr. Kowolenko was a senior executive in the pharmaceutical industry where his last position was as Senior Vice-President of Technical Operations and Product Supply in Wyeth's Biotechnology and Vaccine Division. He has

consulted with and instructed multiple companies and government agencies in the use of analytics in business decision-making.

Dr. Mladen Alan Vouk is a Distinguished Professor of Computer Science, Associate Vice-Chancellor for Research Development and Administration, and Director of the North Carolina State Data Science Initiative. Dr. Vouk has extensive experience in both commercial software production and academic computing. He is the author/co-author of more than 300 publications. His interests include software and security engineering, bioinformatics, scientific computing and analytics, information technology assisted education, and high-performance computing and clouds. Dr. Vouk is a member of the IFIP Working Group 2.5 on Numerical Software, and a recipient of the IFIP Silver Core award. He is an IEEE Fellow, and a recipient of the IEEE Distinguished Service and Gold Core Awards. He is a member of several IEEE societies, and of ASEE, ASQ (Senior Member), ACM, and Sigma Xi.

DOI: 10.1145/3158344