Developing an open source "BigData" cognitive computing platform

Michael Kowolenko and Mladen A. Vouk
NC State University, Raleigh, NC 27695

Introduction

There are five "Vs" in "BigData": Velocity, Volume, Variety, Veracity [1] and Value. The last one is the most important one. "BigData" has no meaning unless it offers a value that otherwise would not be there (Figure 1). Can one answer questions that could not be answered before? Can one make better decisions? Can one make decisions faster? etc. For this to happen "BigData" need to be processed in and analyzed in an intelligent way. Another challenge is to offer access and analysis to such data without compromising privacy, security and safety. Computing platform we outline in this paper provides all necessary elements needed to go from "BigData" to value – speed of processing, ability to handle data volumes fast, an ability to classify and 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.

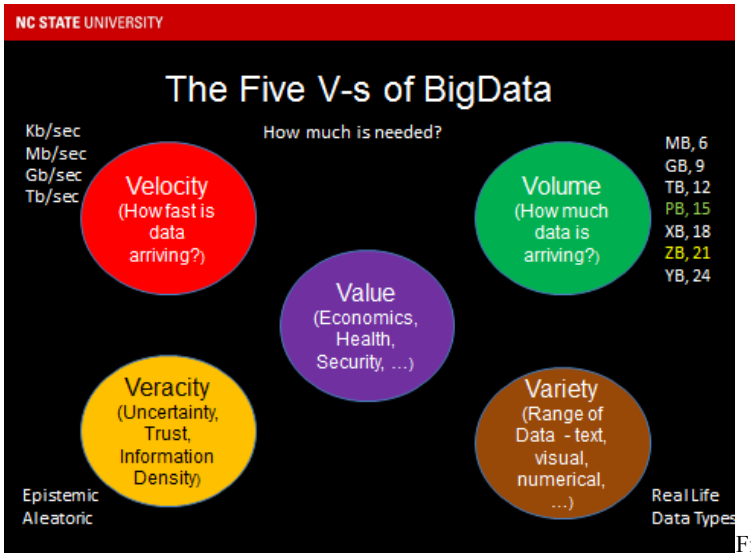

Figure 1. The five V's of "BigData"

Questions

The grand challenge presented to, and addressed by, the IBM Watson project [2-6] 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 may be explored, the type of question posed, and the time and diversity of data required to generate a system of sufficient accuracy 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 a 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.

Comment [v1]: ITng? Or We? Or something like that – perhaps with reference to the class you are teaching (if you cover the platform there)?

The fundamental concept in developing this open source platform was that the system must be capable of ingesting common forms of information, perform a progressive series of filteings and offer analytics resulting in user-specific consolidation of relevant information necessary for the user to make an informed decision (Figure 2). Functionally, the cognitive platform needed to convert information to knowledge while optimizing infrastructure. User input needed to be fed to the analytics models so that results were relevant. This was accomplished by tiering the filters leading to data reduction (crude filters) followed by in depth analytics based on the user interface. Scores and results could then be displayed to the user.

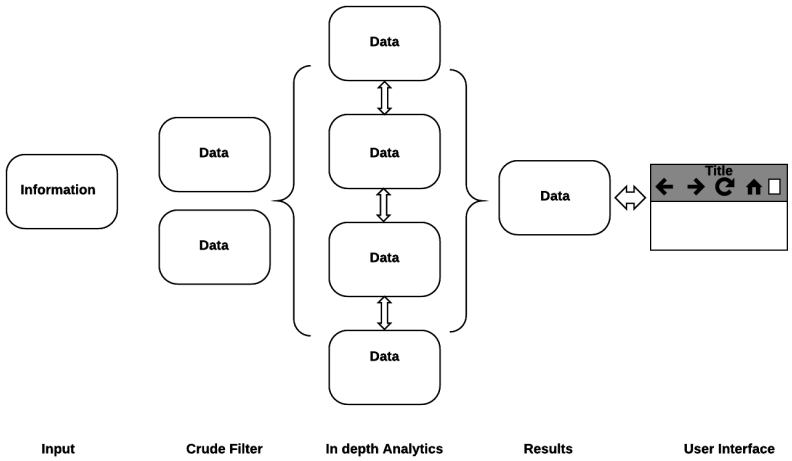

Figure 2 Design Concept

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 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 a 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 serval proof of concept experiments that 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 that 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 match-making begins with the collection of information. The collection process requires that correct decision 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

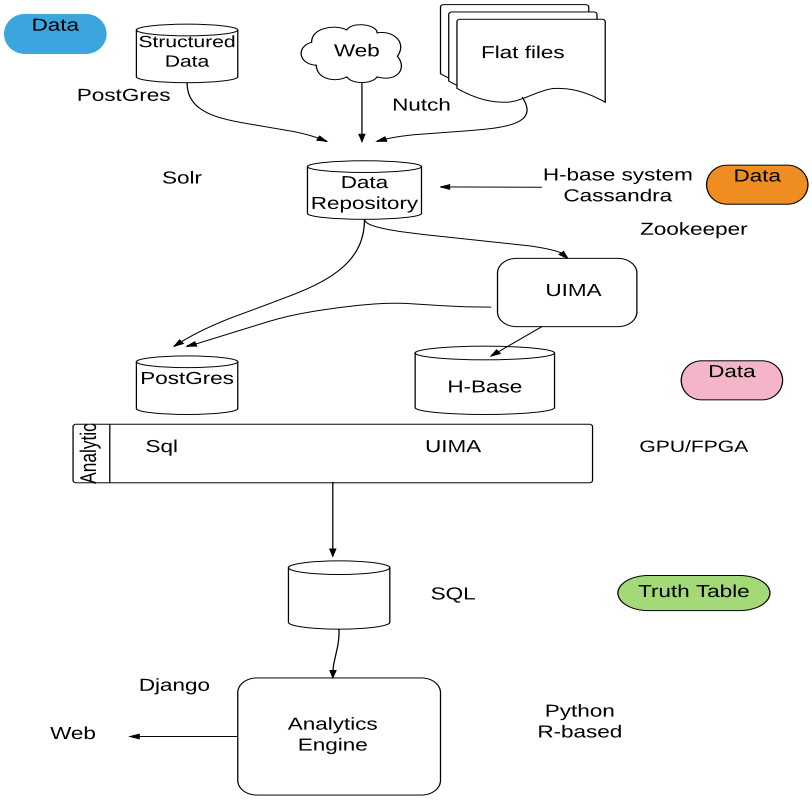

Figure 4. Platform data-flow architecture

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 provide a robust platform for installation of the core databases – Postgres [11] and Apache Cassandra [12].

Postgres serves in 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. Using Cassandra's query language CQL, data stores from web crawls could be quickly filtered of irrelevant data "noise". 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 resulting in improved processing performance.

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 using the Hosting website's search tool. For example, if the web site has a key word search tool, files are downloaded based on the URLontology. 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 ingest 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/

Human-facing nalytical workbench requires a robust set of tools to handle diverse skill sets data scientists and analysists 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 development of machine learning classification models. Interestingly, the latter have been met with user resistance. Subject matter experts find the task of labeling tedious. Rather, the use of domain specific dictionaries combined with rules generated in the UIMA system provide the specificity and context needed in a text extraction system without the frustrating the SME.

The indexing and presentation of text was performed using Solr-Lucene [14,15]. Nutch, was selected because of the highly configurable nature of this tool along with the ease of integration with the above mentioned tools. 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 [27] based graph database. Preliminary results are promising.

Machine Learning

The 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. Over exposure 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.

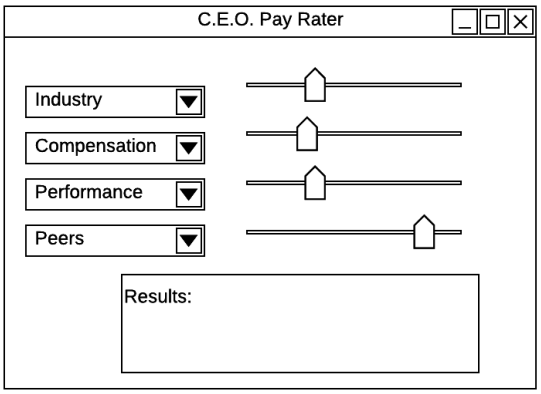

Figure 5. User Interface - CEO Pay Evaluator

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 has been tested in a number of use cases in multiple verticals. In collaboration with pharmaceutical company partners it was used successfully to investigate markets and regulatory compliance issues.. In collaboration with government agencies to the platform was assessed in

the context of security and regulatory compliance situations, such as might occur in financial industry.

The toolkit and the platforms 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 "BigData" applications that solve real word issues. Projects have ranged from an application that could adjust for shifts in political power in the Midle-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 deciesions requires a robust and dynamic approach and support system. . The needs of a data scientist are quire 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 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. "Watson: Beyond Jeopardy!" Artificial Intelligence Volumes 199–200, June–July 2013, Pages 93-105

3. Ferrucci, D. A "Introduction to "This is Watson" IBM Journal of Research and Development, 05/2012, Volume 56, Issue 3.4

4. M. C. McCord, J. W. Murdock, B. K. Boguraev, "Deep Parsing in Watson", IBM J. Res. & Dev., vol. 56, no. 3/4, 2012.

5. J. Chu-Carroll, J. Fan, N. Schlaefer, W. Zadrozny, "Textual resource acquisition and engineering", IBM J. Res. & Dev., vol. 56, no. 3/4, 2012.J.

6. Fan, A. Kalyanpur, D. C. Gondek, D. A. Ferrucci, "Automatic knowledge extraction from documents", IBM J. Res. & Dev., vol. 56, no 3/4, 2012

7 .Reeves, W.W. "Cognition and Complexity: The Cognitive Science of Managing Complexity" Lanham, Md. Scarecrow 1996

8. Jonassen , D.H. "Toward a Design Theory of Problem Solving"Educational Technology Research and Development, 01/2000, Volume 48, Issue 4

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/tinkerpop/gremlin/

28. Small, S.G., Medsker, “Review of information extraction technologies and applications” Neural Computing and Applications, 09/2014, Volume 25, Issue 3

29. Hwang, C.L., K.P. Yoon “Multiple attribute decision making: Methods and applications” Springer-Verlag, New York (1981)

About the authors

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.