

Project Proposal for I524

ABHISHEK GUPTA^{1,*} AND AVADHOOT AGASTI^{1,**}

¹School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

* Corresponding authors: abhigupt@iu.edu

** Corresponding authors: aagasti@iu.edu

project-1: Data mining for a wiki url , March 8, 2017

© 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524

<https://github.com/cloudmesh/classes/blob/master/docs/source/format/report/report.pdf>

1. PROBLEM

Given a wiki URL of a person, find out his details like School, Spouse, Coaches, language, alma-mater etc Typically, the wiki page has all this information available but in the free form text. We need to converting it into structured data format so that it can help us analyze the people, from the networks etc We can create a network by navigating the people mentioned in the wiki page.

2. SOLUTION

Use spark [1] to load the wiki data and create word vectors. Train it using spark ML [2] and then use the model for analytics and prediction. The training set will use Word2Vec. Word2vec [3] is a group of related models that are used to produce word embeddings. Word2Vec is used to analyze the linguistic context of the words. In this project, we created Word2vec model using Wikipedia data. Our focus is people and organization names occurring in the Wikipedia data and to see if Word2vec can be used to understand relationship between people. Typically Wikipedia page for people and celebrities contain the entire family and friends, colleagues information. Our idea is to use Word2vec to see if using a smaller training set of known relationships whether we can derive similar relationship for anyone who has presence on Wikipedia. This mechanism can be then used to convert the data hidden in textual format to more structured data.

Technology Name	Purpose
spark [1]	data analysis
sparkML [2]	machine learning
python [1]	development
ansible [4]	automated deployment
collectd [5]	statistics collection for benchmarking

3. DEPLOYMENT

Solution will be deployed using Ansible [4] playbook. Automated deployment should happen on two or more nodes cluster. Deployment script should install all necessary software along with the project code to the cluster nodes.

4. BENCHMARKING

Solution will use collectd [5] to collect statistics. Once the solution is deployed to the cluster. We should benchmark parameters like

- cpu
- memory
- throughput reads/writes

Benchmarking will be done for one or more cloud providers. The deployment scripts should be agnostic to the cloud provider.

REFERENCES

- [1] "Spark Python API (PySpark)," Web Page, accessed: 2017-02-26. [Online]. Available: <https://spark.apache.org/docs/0.9.1/python-programming-guide.html>
- [2] "Spark ml programming guide," Web Page, accessed: 2017-02-26. [Online]. Available: <https://spark.apache.org/docs/1.2.2/ml-guide.html>
- [3] "Word2Vec, learning vector representation of words," Web Page, accessed: 2017-02-26. [Online]. Available: <https://en.wikipedia.org/wiki/Word2vec>
- [4] "Ansible, deploy apps. manage systems. crush complexity," Web Page, accessed: 2017-02-26. [Online]. Available: <https://www.ansible.com/>
- [5] "collectd - the system statistics collection daemon," Web Page, accessed: 2017-02-26. [Online]. Available: <https://collectd.org/>

6. Conclusion Using this wiki analysis we should be able to build a network based on wiki data.

Acknowledgement We acknowledge our professor Gregor von Laszewski and all associate instructors for helping us and guiding us throughout this project.