Intelligent Environments 2018
I. Chatzigiannakis et al. (Eds.)

doi:10.3233/978-1-61499-874-7-119

Semi-Automatic Ontology Population for Online Quality Assessment of Particulate Matter Sensors

Aboubakr Benabbas [a,1], Hannes Hornig [a] and Daniela Nicklas [a]
[a] *University of Bamberg, Germany*

Abstract. The increasing coverage and heterogeneity of sensor-based systems due to the low acquisition cost, ease of deployment and data collection makes the range of possible applications in the context of smart cities bigger. This makes the process of quality measurement of the generated data as well as anomaly detection some of the toughest challenges to overcome. The absence of a common mean to express general information about the sensor deployment, measurement capabilities, output data and the conditions, under which the sensor could produce anomalies, casts a big shadow over the trustworthiness of the data. The large number of low cost sensors makes the description of their context information even more difficult. We propose an architecture that enables the semi-automatic creation of context information about the sensors through the use of available information about the output data and the available information about the deployment. The definition of the sensor with its relevant context information is done by populating ontology instances for every sensor using the SSN and SWEET ontologies. We take the use case of the luftdaten.info platform with particulate matter sensors. This architecture with its implementation lay the groundwork for further steps such as including anomaly detection rules and quality measurement conditions into the sensor model. The ontology instance produced is the input used for automatic generation of data stream processing queries with data quality assessment.

Keywords. Data Stream Processing, Data Quality, Sensor Networks, Ontologies, Context Modelling

1. Introduction

In a time where the industrial development enabled the increase in the production of cheap and accessible sensors, a wide range of applications are made available at a low cost. These sensors help generate a lot of data about the sensed environment and make the prospect of using their data in smart city use cases a really good prospect.

The assembly of some sensors is so simple that people with next to no knowledge can assemble a sensor after following a simple documentation on how to put together the building blocks of the sensing device. A good example of such devices is Particulate

[1] E-mail:aboubakr.benabbas@uni-bamberg.de

Matter (PM) sensors provided by the OK Lab Stuttgart for the luftdaten.info Citizen Science project[2]. The project is part of the ”Code for Germany” program[3]. The PM sensor system can be built in a short amount of time and provides data about the particles of types PM10 and PM2.5 in addition to the current temperature and humidity. The goal of the project is to enable the participation of citizens in the collection of data about the air quality in one of the biggest and most air-polluted cities of Germany (Stuttgart), to cover the largest possible area of the city and provide extensive information about the current air quality. More of these sensors are also spread over different areas in Germany. Those sensors are easy to build and deploy and provide simple access to air quality data for the community and city management. However, they become faulty under certain circumstances. Furthermore, they do not have enough installation information, which means very low context information is available for every sensor.

Anomaly detection and quality measurement are closely related to the context of the sensors. Context can be any information that describes directly or indirectly the conditions of the sensor with regards to its deployment area and the sensing process. The huge number of sensors, coupled with a lack of provenance and context information makes the anomaly detection and the quality assessment a tricky task. We propose an architecture to create ontology-based context information about the sensors from the available resources in the internet, which makes the ontology model instantiation for every sensor occur in a semi-automatic manner. The instances created for every senor can then be used to generate queries for online quality assessment to run on a customizable Data Stream Management System called Odysseus [1].

The rest of the paper is structured as follows: In the second section we discuss the related work about ontology based approaches to describe context, define anomaly detection rules, instantiate ontologies and describe quality assessment methods. The third section describes in detail the use case we take to implement our approach where we describe the environment and situation of the sensors. In the end of the section we derive the requirements from the use case. In the fourth section we present the approach and the underlying architecture of our solution. The fifth section contains the evaluation of the architecture in terms of feasibility and impact. Finally we discuss the achieved work so far with the future steps.

2. Related Work

Batini et al. offer an interesting view on the data quality dimensions and their respective definitions in [2]. The definition of Batini covers the most important dimensions of data quality that sensor data need to have like accuracy, completeness and timeliness.

Work in the area of data quality and data anomaly using Ontology-based solutions is available and has provided solutions that use ontologies to describe the quality metrics. Geisler et al.[3] propose a Data Quality ontology-based framework for data stream applications, where they define quality metrics for content, queries and applications that use the data. The framework uses an ontology to define all the meta-data for the data quality

[2] http://luftdaten.info/

[3] https://codefor.de/

metrics. The framework offers the option of describing the sensors and their quality metrics through semantic rules, but all the meta-data about the sensor need to be provided. Kuka et Nicklas [4] provide a solution for general quality-aware sensor data processing, that uses probabilistic processing to provide continuous data quality values for the incoming data. In [5], kuka enhances the process by adding a description of the context by using the SSN ontology [6] to describe context information about the sensors used.

The issue of Automatic Ontology Instantiation was addressed by Shchekotykhin[7], where they present a comprehensive ontology instantiating system that mainly instantiates information provided in tables. Makki et al. [8] propose a semi-automatic ontology instantiation in the domain of risk management, where they use *Natural Language Processing* to extract knowledge and information. Alani et al. [9] achieve ontology population by linking a knowledge extraction tool to an ontology to provide the information extracted in a machine readable format. The work in the area of ontology instantiation did not cover the cases of sensor data and is limited to some specific use cases.

The use of ontology-based solutions for anomaly detection rules description is presented by Sarno and Sinaga [10]. Roy and Davenport [11] develop a maritime domain ontology backed by automated reasoning service for anomaly detection, classification of vessels and identification of threats. An often used field of application for anomaly detection and context are in the area of intrusion detection in Network surveillance. Some attack scenarios are only detectable by interpreting the network metrics given the Context information like network load by time of the day [12]. From the above contributions it becomes apparent that ontology based solution to describe anomaly detection rules for sensors is missing. We note also the lack of mechanisms to populate automatically or semi automatically ontology based descriptions of deployed sensors.

3. Use Case

Sensors are known in general for their faulty behavior. We can get information about the behavior and data quality of the sensor from:

- Data sheets containing conditions, under which it functions properly. It also specifies the extreme conditions that cause the sensor to provide erroneous values and give anomalies.
- Learned dependencies between the deployment environment and the sensor; the accuracy of the values of a sensor can also depend on other values like external factors, which could impair its accuracy. In a previous work[13], we have shown that historical data from people counting sensors and distance sensors can be used to find correlations that are used to determine the accuracy of the former sensor based on the distance measured by the other.

Both sources count as context information and its inclusion in online processing of the sensor data helps to improve quality measurement and to detect anomalies. There are a couple of possible use cases for the deployment of context information through ontologies. Context information like weather, location or time can be used to evaluate the measurements given by the sensor nodes. This information defines ranges of acceptable data points. An example is the particulate matter sensors (called PM sensor for the rest of the paper) of the OK Lab Stuttgart. The sensor nodes have a maximum survival range related to the humidity of the environment. Such knowledge can be introduced into a context representation through ontology instantiation of the sensor, which can be then used to generate data stream queries that perform anomaly detection on the incoming data. Context information is also valuable, when it enables the use of data fusion to measure the quality of the sensors. Considering the sensors used in the luftdaten.info, we know that the humidity affects considerably the accuracy of the particulate matter values. In the used sensors systems, the used particle meter is the SDS011 of Nova Fitness[4]. The data sheet describes a Humidity upper bound of 70% for a working environment. If the humidity rises above this value, the quality of data decreases, and more anomalies can occur. On some sensor platforms are PM sensor installed together with humidity and temperature sensors. For these sensor platforms the anomaly detection rule derived from the data sheet could be used with the on-board humidity sensors. The rule specifies a relative error of 15% ($\pm 10\ \mu g/m^3$) in a humidity less than 70%. Above this range the quality is undefined. Since these values also come from cheap unreliable sensors they need verification and cannot be used alone to check the PM values for anomalies. We need to use other data sources like the weather stations located in an acceptable range of the deployed sensors to perform data fusion to closely check, whether the sensor have the normal conditions to function properly and to check the accuracy of the values of humidity. Fig.1 shows how the weather stations within a close range from the sensors can be related to each other and later used as data source for quality assessment and anomaly detection of the sensor measurements.

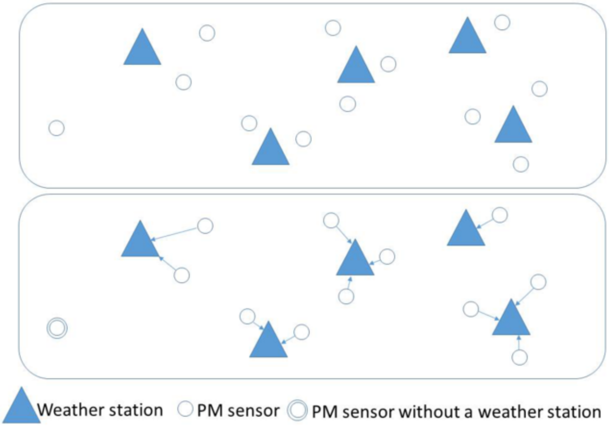

Figure 1. Linking of the sensors to the weather stations based on location

For the sensor platforms without humidity sensors, the context information (location of the nearest weather station and how to access its data) provides the possibility to deploy the anomaly rule. The deployment location of the sensor can also influence the measurements. If a sensor is deployed indoors, then we get constantly values that are far from the reality. The quality of a sensor data can be measured by using such a query operator in Listing1

[4] https://inovafitness.de/shop/sds011/

```
output= QUALITY({ATTRIBUTES = ['PM10'],
PROPERTIES=['Accuracy']}, FoI:PMSensor)
```

Listing 1: Query Operator to Measure the Accuracy of PM10 values

Such a query operator can measure the accuracy of the sensor, but it needs information on how to estimate the accuracy. This information must be at least provided once (in the case of fact sheet about the working conditions of the sensor), or be regularly updated if the sensor makes behavioral changes. If such information has to be entered every time by a human user for a new sensor or for an update of an existing one, then the whole process of quality assessment becomes burdensome and unwieldy.

Since those sensors are assembled and put into use completely independently by end users, the task of estimating the quality of their data becomes very difficult without an automatic or a semi-automatic process. Such quality assessment operators can be kept simple, if the ontology instantiation of every sensor is created and maintained with the needed information about the sensor and its dependencies.

From the description of the sensors and used we can derive the following requirements:

- We need to serialize location information of the sensors based on the available information.
- For every sensor, we need to find data sources like weather stations or neighbouring sensors to compare the reported values of every sensor.
- The ontology based population of every sensor must contain information about the output schema.
- The ontology instantiation of every sensor should also describe directly the data sources related spatially to the sensor.

The next sections present the approach adopted and the system implementations to populate the ontology for every sensor, while adhering to the requirements set from the use case.

4. Approach

4.1. Architecture

The architecture consists of two parts as shown in Fig. 2: the Ontology Processing Engine (OPE) and the Stream Processing Engine (SPE). The first part is concerned with collecting the context information about the sensors and weather stations to generate instances of the ontologies. Ontology instances express all the needed information about the sensor. This information is gathered from three main sources: information about the sensors from the data sheet, information about weather stations, and the data generated by the sensor. The first source gives the location of the sensor, minimal working conditions of the sensor and data quality related conditions. The second gives a full list of the existing weather stations and their coordinates. The third one helps to extract the schema and the types of every attribute in the data. The Stream Processing Engine receives data

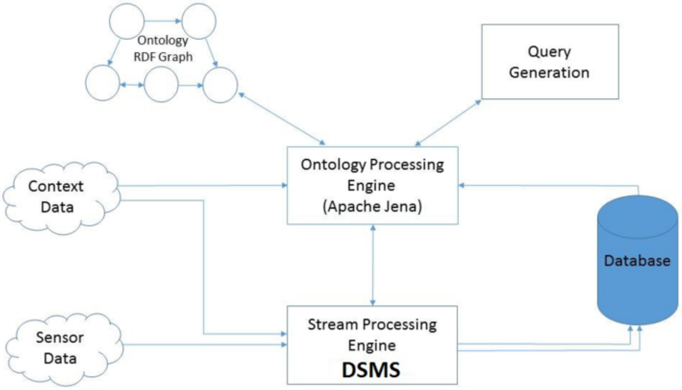

Figure 2. Architecture of the ontology population system

streams from the sensors and pulls information about the weather stations and their data. It prepares the data for storage and writes it to the database.

The OPE is the main component that takes the information described above and generates the ontology instances for the sensors. It stores the instances in an RDF Graph and sends the model (ontology instantiation) of every sensor to the query generator. The generated queries measure the quality of the sensor data based on the sensor model described by the ontology instance. The queries can be deployed on a Data Stream Management System like *Odysseus*, where the data from the sensors come in a continuous stream to be assessed.

4.2. Ontologies

Ontologies have an important role in the architecture. The OPE must have a clear set of base ontologies that satisfy the requirement of giving full details about the sensor, its data quality requirements and the possible reference data sources in the form of weather stations. We use a combination of SWEET ontology[14] from Nasa and SSN ontology [6]. The SWEET ontology is used to model context information while the SSN ontology mainly models information about the sensor node. The SWEET ontology consists of nine top level ontologies and 200 sub ontologies with around 6000 concepts. The top level ontologies describe different concepts like matter, processes, relationships, phenomena, human activities, properties and states.

The Realm ontology, is one of the top level ontologies. It describes location related concepts with a great degree of precision. The main context information we use for our use case is weather data provided by the German Weather Service. Since we need to specify the location information about the involved weather stations, *the Realm* ontology fits perfectly the need for a complete description.

The SSN ontology only allows us to model the sensors with its technical data and output schema. However, the sensor platforms and observations are a bit abstract, thus, not intended to describe domain concepts like the actual type of measurements and the physical properties they measure e.g. fine dust and its corresponding measurement $\mu g/m^3$. The SWEET ontologies provide also a solution for this problem. To model concepts like measurements, we use the Representation sub ontology.

4.3. Implementation

The Ontology Processing Engine is responsible for populating the ontology and storing it on a linked data graph. It runs on the basis of Apache Jena framework[15]. Jena is a good choice mainly, because it provides next to its RDF data manipulation, features for RDF Graph storage, spatial search and querying. It receives data from PM sensors through the Stream Processing Engine and from the German Weather Service (DWD) about all weather stations.

The SPE, implemented using a DSMS, gets weather stations' data by querying the GeoServer of the DWD through a simple http-get-request, it formats the data as JSON data (id,location, and measurement data) and forwards the id and location to the OPE. The aforementioned JSON data contains all weather stations. Weather stations are instantiated using *the Realm* ontology with its location information and stored into the RDF Graph. The SPE receives the sensor data as JSON and inputs it into our Data Stream. We transform the JSON data to relational data and write it into a database. Upon receipt of JSON data containing basic information about the sensor like ID, the data is passed to the OPE to check if the sensor node is already available in the RDF Graph of the ontology, then, the node is instantiated and added to the Graph. The JSON data is parsed and location information and working conditions of the sensor are extracted and modelled using the classes of the SSN ontology.

To link the sensors to the nearest weather stations, we use Apache Jena ARQ API. It combines simple spatial querying with SPARQL[16]. This search is done using a spatial index created by Apache Lucene[17] from the spatial information provided by the input data. The weather station location is represented by a polygon through a *bbox* and the PM sensor has a geolocation, thus, a spatial SPARQL query is issued for every sensor to find the nearest weather station. The SPAQL query could yield more than one station. If this happens the centroid of the polygon given by the weather station is computed and the weather station with the nearest centroid is assigned to the PM sensor. After the sensors and the weather stations are linked, the RDF Graph now contains all the information about every sensor with the related weather station.

With the RDF Graph available, the Query generator creates for every sensor a query that measures the accuracy of the PM and Humidity values. The query is then started by *Odysseus*, where Humidity data streamed by the weather station is fused with the data from the related sensor. From humidity values of the weather station, the humidity of a sensor is directly checked, whether it's accurate or not. Furthermore, the continuous query reports a drop in the accuracy of PM values as soon as the humidity rises above 70%. Anomalies are also detected, if a sensor keeps sending high humidity values, although the station reports a dry weather.

5. Evaluation

5.1. Feasibility and Impact

We measure the feasibility by the requirements set in the use case. The first requirement is fulfilled, since all the available information about the sensor is integrated into the RDF Graph and can be used to generate the queries. We get in important parameters on the working environment like the humidity, the temperature and the geolocation. We also manage to use every weather station in Germany when it is applicable to the target sensor. This approach is semi-automatic because the selection of the classes and relationships to use to create the instances and relations in the ontology has to be done beforehand for each type of sensor. The PM sensors and the weather stations have to be modelled so that the JSON can be used as concrete input data to instantiate a specific sensor in the ontology. The output schema of the sensors and the weather station can be inferred automatically due to the schema inference capability, which makes the process of schema extraction completely user-independent. The generated ontology instance of every sensor has a direct link to the related weather station with the its data schema and how to access the data.

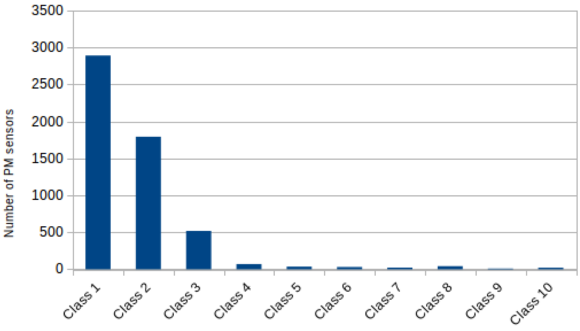

Figure 3. Classes of coverage in 20KM intervals

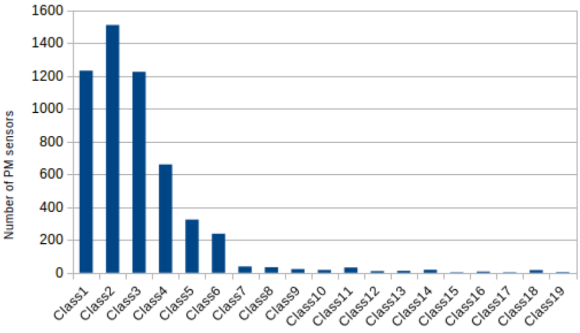

Figure 4. Classes of coverage in 10KM intervals

Given the overall huge number of sensors deployed throughout Germany and the available weather stations in Germany (912), we used a sample of 5410 PM sensors. The coverage quality is measured using 2 different ranges, with every range having up to 20 classes. Every class stands for a radius of the next available weather station for every sensor. The first class stands for the nearest weather stations and the next classes mean that the next possible station is further away. In the histogram depicted in Fig. 3, we see that with a an increasing radius of 20 kilometers coverage, 53% of the available sensors find a weather station within a radius of 20 kilometers and 33% are within 40 kilometers of the next weather station. Starting from the third class, the coverage can no longer be reliable and the weather stations can no longer be used to assess the quality of the sensors. Using a stricter range of 10 kilometers, we managed to get a good coverage too. Fig. 4 shows that 23% are within 10 kilometers of a weather station, 28% of the sensors find a weather station within 20 kilometers and 23% within 30 kilometers.

5.2. Performance

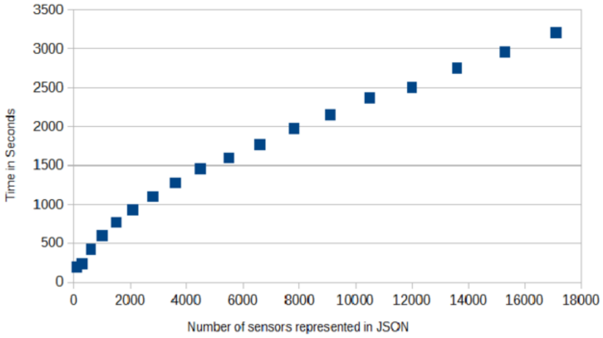

Figure 5. Time needed to create the RDF Graph depending on the number of sensors

We tested the architecture to see how it performs when faced with a huge load of sensors. The computer used for the experiments has the following specifications: Intel(R) Core(TM) i5-2520M @ 2.50 GHz (4 Kerne) with 8GB of RAM and the OS used is Ubuntu 16.04.3 LTS. Fig. 5 shows how the instantiation time evolves with the increase in the number of sensors to process. The run time becomes linear after 4000 sensors, which means that the performance will not suffer much if the sensors increase gradually.

6. Conclusion and Future Work

The proposed work lays the foundation for an online and adaptive quality-aware processing of PM sensors based on the context information available on the sensors. We extract all the available information about the sensors and relate them to the neighboring weather stations to get reference data to measure the quality of the sensors and to also detect predefined anomalies. The work achieved so far is a preparation for the next step that is the automatic query generation for different streaming systems. The fact that sensors with their quality conditions and anomaly detection rules are modelled through ontologies makes the automatic query generation possible, and also for different streaming systems. We have focused on the ontology instantiation and want to try to generalize it for other types of sensors. The next step is to integrate a Data Stream Management System, which runs the generated queries and informs the OPE if any changes in the anomaly detection rules or quality conditions occur. Since tasks in the Stream Processing Engine can be parallelized, we can deploy it on a cluster to have a better performance.

References

[1] H.-J. Appelrath, D. Geesen, M. Grawunder, T. Michelsen, and D. Nicklas, "Odysseus: A highly customizable framework for creating efficient event stream management systems," in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '12. New York, NY, USA: ACM, 2012, pp. 367–368.

[2] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[3] S. Geisler, C. Quix, S. Weber, and M. Jarke, "Ontology-based data quality management for data streams," *J. Data and Information Quality*, vol. 7, no. 4, pp. 18:1–18:34, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2968332

[4] C. Kuka and D. Nicklas, "Supporting quality-aware pervasive applications by probabilistic data stream management," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '14. New York, NY, USA: ACM, 2014, pp. 330–333. [Online]. Available: http://doi.acm.org/10.1145/2611286.2611319

[5] C. Kuka, "Qualitaetissensitive Datenstromverarbeitung zur Erstellung von dynamischen Kontextmodellen," Ph.D. dissertation, University of Oldenburg, 2014.

[6] M. C. et al., "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25 – 32, 2012.

[7] K. e. a. Shchekotykhin, *AllRight: Automatic Ontology Instantiation from Tabular Web Documents*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 466–479.

[8] J. Makki, A.-M. Alquier, and V. Prince, *Semi Automatic Ontology Instantiation in the domain of Risk Management*. Boston, MA: Springer US, 2008, pp. 254–265.

[9] H. Alani, S. Kim, D. E. Millard, M. J. Weal, W. Hall, P. H. Lewis, and N. R. Shadbolt, "Automatic ontology-based knowledge extraction from web documents," *IEEE Intelligent Systems*, vol. 18, no. 1, pp. 14–21, Jan 2003.

[10] R. Sarno and F. P. Sinaga, "Business process anomaly detection using ontology-based process modelling and multi-level class association rule learning," in *2015 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, Oct 2015, pp. 12–17.

[11] J. Roy and M. Davenport, "Exploitation of maritime domain ontologies for anomaly detection and threat analysis," in *2010 International WaterSide Security Conference*, Nov 2010, pp. 1–8.

[12] A. Aleroud and G. Karabatis, "Contextual information fusion for intrusion detection: a survey and taxonomy," *Knowledge and Information Systems*, vol. 52, no. 3, pp. 563–619, Sep 2017. [Online]. Available: https://doi.org/10.1007/s10115-017-1027-3

[13] A. Benabbas, S. Steuer, and D. Nicklas, "Towards quality aware sensor data stream processing in a smart city living lab," in *Grundlagen von Datenbanken*, 2017.

[14] R. G. Raskin and M. J. Pan, "Knowledge representation in the semantic web for earth and environmental terminology (sweet)," *Computers & Geosciences*, vol. 31, no. 9, pp. 1119 – 1125, 2005, application of XML in the Geosciences. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098300405001020

[15] B. McBride, "Jena: a semantic web toolkit," *IEEE Internet Computing*, vol. 6, no. 6, pp. 55–59, Nov 2002.

[16] I. e. a. Kollia, "Sparql query answering over owl ontologies," in *The Semantic Web: Research and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 382–396.

[17] A. Biaecki, R. D. J. Muir, and G. Ingersoll, "Apache lucene 4," 2012.