Semantic Content Enrichment of Sensor Network Data for Environmental Monitoring

Dustin R. Franz and Ricardo A. Calix

Purdue University Calumet, CITG, 2200 169th Street, Hammond, IN, 46323-2094
{dfranz, ricardo.calix}@purduecal.edu

Abstract

The Semantic Sensor Web (SSW) will eventually revolutionize how we perceive and query information about the physical world. Currently, there is an ongoing effort to develop a searchable web of things that sense and control the world. As this new internet of things expands, there will be an explosion of available raw data that may not always be reachable by users. Bridging this gap between what the user wants and the information collected and represented by embedded devices is a critical issue. In order to really maximize the benefit of such a web of networked sensing devices, new semantic approaches that can infer and predict additional information about the sensors and their context need to be developed. This paper proposes a content enrichment approach that uses sensor and context data as features to predict new meta-tags that can further identify relevant categorizations for the embedded devices and the physical data they collect. Specifically, machine learning classification and regression techniques are used to predict semantic tags for each embedded system context. Results of the 10-fold cross validation analysis and feature ranking are presented and discussed.

Introduction

The Semantic Sensor Web (SSW) will eventually revolutionize how we perceive and query information about the physical world. Currently, there is an ongoing effort to develop a searchable web of things that sense and control the world. Eventually, this technology will improve traffic conditions, will be able to reduce costs in elderly care, help to monitor safety in factories and other work environments, and develop entire new industries that help to improve the way humans live and work.

As this new internet of things expands, there will also be problems such as an expansion in the digital universe. This will present many challenges for accurate information retrieval. Information that could be useful to people may not always be reachable by those who need it because it lacks proper meta-tags. Bridging the gap between what the user wants and the information collected and represented by embedded devices is a critical issue. In order to really maximize the benefit of such a web of networked sensing devices, new semantic approaches that can infer and predict additional information about the sensors and their context need to be developed. To address these issues, this paper proposes a content enrichment approach that uses sensor and context data as features to produce new meta-tags that can further identify relevant categorizations for the embedded devices and the physical data they collect. The proposed methodology uses a 3 step approach for content enrichment. In step 1, the collected data from the sensors is combined with other API data to create features which can be used to build a classification model. In this case, the model can predict 2 classes which are polluted and not-polluted. For the classification analysis to predict between polluted and non-polluted classes, machine learning classification algorithms such as Naïve Bayes and Support Vector Machines are used. Once this class is predicted, this new information plus the features are added to a context list. In step 2, cosm.com, a website for the internet of things, is leveraged to find additional information related to the embedded devices. Cosm.com is similar to a social network for embedded devices. This site provides additional tags which can be associated with each embedded device. Therefore, the results from the classification and features, are combined with meta-tags from cosm.com to extend the context list. Finally, in step 3, a knowledge base is used to generate new words that can be related to the words in the context list. The final content enrichment stage uses simple heuristic rules and knowledge bases to produce additional semantic tags for each embedded system context.

Packet based networks can sometimes be unreliable and lose packets. To address the issue of missing data, such as missing sensor data, a regression approach is also proposed which could be used to calculate values for missing data based on regression modeling. This regression approach is also presented and discussed.

Literature Review

Background

The semantic sensor web is a new emerging field that promises great benefits for the future. By including semantic content in web pages, the semantic web enables users to find, share, and combine information easily. This approach encourages including semantic content such as meta-data to express relationships specifically between web pages. Important recent works in this field include Klusch et al. (2009); Thirunarayan and Pschorr (2009); Broring et al. (2011); Roman and Lopez (2009); Dlodlo (2012); and Ramya et al. (2012). In Klusch et al. (2009), the authors created ontologies to better define the structure of the semantic web. Structures are the key components that define the relationships of a semantic web. Recently Thirunarayan and Pschorr (2009) showed how annotating semantic meta-data to raw sensor observations provides better interpretation and context. Wireless sensor networks (WSN) consist of devices such as the Arduino micro-controller connected to sensors, which collect data and pass their data through a network to a central location for further analysis. Initially the WSN was not concerned with the Internet. However, projects such as Roman and Lopez (2009) have involved using these sensor networks through the Internet to create the so called Internet of Things (IoT). Machine learning, which is a common approach to semantic analysis, has also been applied to sensor networks. One important area has been for environmental analysis. Dlodlo et al. (2012) shows how using the IoT can be used in environmental monitoring to improve the South African economy. Ramya et al. (2012) shows how environmental monitoring using embedded systems can improve the life of sensor networks inside a motor vehicle while maintaining quality of monitoring. The issue of how to search data in this new semantic web of things has been addressed by Broring et al. (2011). They indicated various technologies and frameworks to provide structure to the web of things. Sensor meta-data is described using SensorML, which helps support discovery, provide information, and describe post-processing steps.

Machine Learning Methods and Feature Ranking

In this study, the Naïve Bayes classifier and Support Vector Machines are used. Naïve Bayes is a probabilistic classifier that assumes that all features contribute independently to the prediction of a class. SVMs are binary classifiers based on statistical learning theory. They seek to maximize the margin that separates samples from two given classes (Vapnik and Cortes (1995); Burges (1998)). The maximization of the margin is defined using the support vectors. These support vectors are the samples closest to the optimal separation line. The calculation of the separation line parameters is performed using a quadratic optimization technique with Lagrange multipliers. The constraints of the optimization model are defined as the boundaries set by the support vectors. Formally, the "objective function" and constrains are formulated as follows:

$$\text{Minimize} \quad \frac{1}{2}\|w\|^2 + C\sum E_i \tag{1}$$

$$\text{Subject to} \quad Y_i(w \cdot X_i + b) \geq 1 - E_i \tag{2}$$

where w represents the weight vector of the model, C controls the priority between empirical and structural risk, E_i is used to calculate the sum of the errors in the model, Y_i is the sample class for each sample, X_i is the feature vector for the sample, b is the offset from the origin used for the hyper plane, and "i" represents each sample.

For the purposes of feature ranking, the information gain method has been used. Finally, to address the issue of missing data, regression techniques are applied. Regression analysis is commonly used to predict real valued variables using linear regression techniques. However, for non-linear data other techniques may be required. Support Vector Regression (SVR) (Smola and Scholkopf (2004)) has the advantage that it can rely on kernel methods to map non-linear data to higher dimensional spaces where a linear regression can be calculated. Formally, the optimization problem for SVR is formulated as follows:

Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{3}$$

Subject to:

$$Y_i - w \cdot \phi(X_i) - b \leq E + \xi_i \tag{4}$$

$$w \cdot \phi(X_i) + b - Y_i \leq E + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

where "i" represents each sample, the variable C (cost) represents the tradeoff between prediction error and the weight vector, and ϕ() maps the data to higher dimensional feature space. The optimization model formulated with equations (3) and (4) can be represented as follows:

$$g(x)_i = \sum_{i=1}^{vs} (\lambda_i - \lambda_i^*) K(x_i, x) + b \qquad (5)$$

here K(x_i, x) = $\phi^T(x_i) \cdot \phi(x)$ is the kernel function that maps the input vectors to higher dimensional space. Here, x represents the input vectors, and λ's are the LaGrange multipliers.

Methodology

Overview

The proposed methodology uses a 3 step approach for content enrichment which combines data classes predicted using machine learning techniques with meta-tags extracted from an Internet of things website. The 3 steps are described as follows:

Step 1: The collected data from the sensor is combined with other API data to create features which can be used to build a classification model. In this case, the model can predict between 2 classes which are polluted and not-polluted. Once this class is predicted, this information plus the features are added to a context list which is associated with each embedded system context.

Step 2: The cosm.com website is leveraged to find additional information related to the embedded devices. This site provides additional tags which can be associated with each embedded device. Therefore, the results from the classification and features, are combined with meta-tags from cosm.com to extend the context list.

Step 3: A knowledge base called ConceptNet (Havasi et al. 2007) is used to generate new words that can be related to the words in the context list. These new words provide the additional content enrichment.

Web Sensor Architecture

A total of 5 web sensor devices were developed and located in 5 different locations throughout northwest Indiana. Each embedded device had 2 sensors to measure environmental gases. The sensors are MQ-7 and MQ-131 (Figure 1). The MQ-7 sensor has a high sensitivity to carbon monoxide (CO) gases and the MQ-131 sensor has a high sensitivity to Ozone (0_3) gases. The sensors are set up directly connected to Arduino micro-controllers via a breadboard. Both sensors require a warm-up period of 5-10 minutes before sensing accurate data. Each Arduino is connected to either an Ethernet or Wi-Fi shield to have a connection to the Internet. The type of shield is determined by the type and location of the network to which it needs to connect. For instance, when collecting polluted data in an outside environment such as a parking garage, the embedded devices use the wi-fi shields to be able to reach the internet.

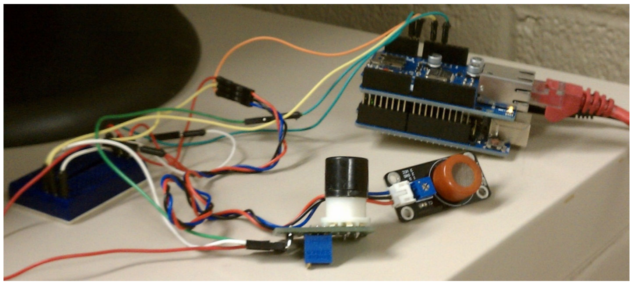

Figure 1 Embedded System Device and Sensors

In Figure 1, a prototype of the embedded systems device with the 2 sensors can be seen. The device is housed indoors and has an electric power supply. The Arduino Uno microcontroller boards are used for this work. The MQ-7 sensor has a range of 20 parts per million (ppm) to 2000 ppm for CO. The MQ-131 sensor has a range of 10 parts per billion (ppb) to 2000 ppb for 03.

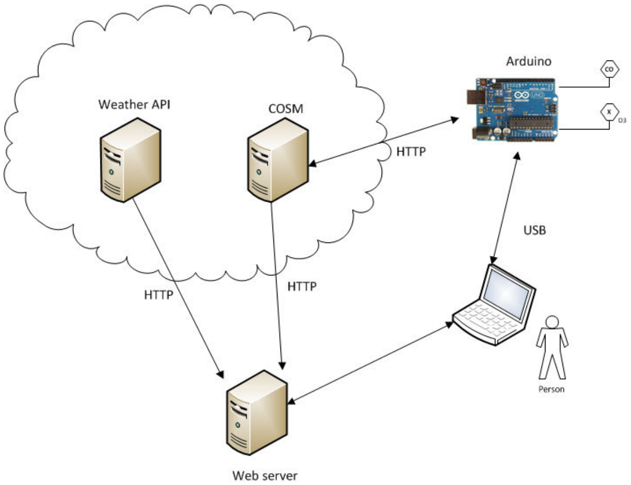

Figure 2 Project Reference Model

The data from the sensors was collected as ppm for CO and ppb for O_3 and sent to the cosm server every 5 minutes. Figure 2 shows an overview of how the project data flows through the network. The user configures the Arduino via USB so that the device knows how to connect to the Internet and to what pins the sensors are connected. The Arduino communicates with the cosm server to upload the data it collects. A different web server, using a Python script, then collected data from the cosm server along with other data.

Pollution Data Corpus and Features

The data set includes samples from 2 classes which are polluted and not-polluted. A total of 2600 un-polluted samples and 98 polluted samples were collected. The data was collected during a 3 month period whenever the Arduinos were connected to the Internet and at the designated sites. Polluted data was collected by placing the collecting device near the exhaust of a car. For this work, the PUC race car project was used. Collection took place in the PUC race car and manufacturing laboratory.

A total of 10 features were used in the data set for each reading sample. The features of the dataset are: MQ131, MQ7, latitude, longitude, temperature in Fahrenheit, dew point, precipitation in the last hour in inches, pressure in Hg, relative humidity, and date and time. The MQ131 and MQ7 data represents the concentration for each described gas. Latitude, longitude, time and date help to make each location and time period distinct in the analysis. Finally, each feature related to the weather was collected to determine if and how gas concentration was related to weather.

Machine Learning and Feature Ranking

Once the corpus was collected, the classification analysis and feature ranking were performed. Classification analysis was performed using Support Vector Machines (SVM) and Naïve Bayes. These methods are selected because Naïve Bayes tends to be very fast but less accurate than SVM. In contrast, SVM can be more accurate but slower and harder to tune model parameters. By using both methods, a better understating of the classification accuracy and speed boundaries for the data can be determined. Feature ranking was performed using information gain based feature ranking. To address the issue of missing data, a regression modeling approach has been implemented. Both linear regression and Support Vector Regression are used and compared using 10-Fold cross validation. As a proof of concept, a regression function is estimated for the MQ7 variable. The independent variables consist of the collected features for the data set. The SVR technique uses the RBF kernel as a mapping function. Results are presented and discussed in the results section.

Meta-tags and Cosm

For this experiment, cosm (cosm.com) has been used. Cosm is a platform for which users can share and search data related to embedded devices. It contains tags that help to describe the embedded device and the data that the device collects.

Semantic Content Enrichment

The semantic content detection methodology proposed in this work is useful in content enrichment of embedded devices connected to the Internet of Things (IOT). Here, the predicted polluted vs. non-polluted class is combined with additional tags or keywords downloaded from the cosm website and other API's to help better describe the site. Each identified tag helps to enrich the overall embedded device. In the enrichment process, the system takes one embedded device as input and generates an XML representation of the device with a natural language text description. The XML mark-up serves as a semantic representation of the embedded device which can be used as input for various systems such as in building information modeling (BIM), information retrieval indexing engines, for engineering tasks, and automatic 3-D generation systems.

The XML representation uses tags similar to what is used in current Natural Language Processing (NLP) systems. The tags used for representation can range from low level features to high level semantically rich tags. The extracted feature vectors described in previous sections are an example of low level data. High level tags can include predicted class tags and inference tags generated using the knowledge base ConceptNet. For example, if the embedded device tags are for exposure indoor, disposition as fixed, physical domain, and a user who is associated with a university by their email or contact name; then, the inference rules may determine that this device could be located on or near a university campus and that the purpose of the embedded device is for educational purposes. The automatic semantic tag generation step can use predefined text descriptions with variable parameters.

Analysis and Results

Classification Analysis

The corpus has a high class imbalance. To address this issue, SMOTE oversampling technique (Chawla et al. 2002) was used. This technique allows for balancing the 2 classes in the data.

Table 1: Classification Results

	Naïve Bayes	SVM
Polluted	0.99	0.99
Not-Polluted	0.99	0.99
All	0.99	0.99

After applying SMOTE, the corpus consisted of 2600 samples of un-polluted data and 2548 of polluted data. Naïve Bayes and SVM with a linear kernel were used. The analysis was performed using 10-fold cross validation. As can be seen from Table 1, both classifiers performed well on the classification task. These results show promise for this type of classification task. However, it is important to note that future work must include more polluted samples.

Feature Analysis

Results of the feature analysis using the information gain ranking technique can be seen in Table 2. From these results, it can be seen that the MQ131 and MQ7 sensors have the highest contribution to classification accuracy. Additionally, it can be seen that pressure and dew point have a higher contribution than temperature and precipitation.

Table 2: Feature Ranking

Rank	Score	Feature
1	0.98	MQ131
2	0.94	MQ7
3	0.93	Pressure
4	0.91	Dew point
5	0.89	Relative Humidity
6	0.80	Temperature
7	0.00	Precipitation

Regression Analysis

To address the issue of missing data, a regression analysis was conducted for the MQ7 parameter using both linear regression and support vector regression. For this regression, the imbalanced dataset was used without applying the SMOTE oversampling technique. From Table 3, it can be seen that the regression function for MQ7 obtains a good Root Mean Squared Error (RMSE) and a good correlation coefficient. Additionally, the SVR technique obtained slightly better results than the linear regression approach. The linear regression model can be seen in Table 4.

Table 3: Regression Analysis Results

	RMSE	Correlation Coefficient
Linear Regression	0.058	0.8739
SVR	0.058	0.90

Table 4: Linear Regression Model

Coefficients	Variables
	MQ7=
-0.0256 *	mq131 +
-136.6782 *	lat +
97.5126 *	lon +
0.5069 *	temp_f +
-0.5919 *	dewpoint_f +
-48.3164 *	precip_1hr +
-24.7047 *	pressure_in +
23.9518 *	Relative_humidity +
14968.1263	

Content Enrichment

For this work, embedded devices connected to the internet of things can be enriched by taking the results of the classification task (polluted vs. un-polluted), and the extracted meta-tags from cosm, and producing a semantic representation, and a natural language description of the embedded device. For example, features can be used to predict if the area where the device is located is polluted or not-polluted.

```
<?xml version="1.0" encoding="ANSI_X3.4-1968"?>
<embedded_device>
<semantic>
    <inference>
    <location at_location="university"/>
    <location at_location="factory"/>
  …
</inference>
<sensor orderid="3" MarkableID="D001">
              <class is_fixed="1" polluted_site="1" >
<type f="MQ7"/>
</class>
</sensor>
…
<raw_features>
      <feature f=pressure value="30">
      <feature f=dew_point value="28">
      <feature f=relative_humidity value="0.45">
      <feature f=time value="12">
</raw_features>
…
</semantic>
</embedded_device>
```

Figure 3 Semantic Embedded Representation

The resulting classes per device are combined with the extracted tags from the downloaded cosm html file to create a list of information tags associated with the device. Once the classes are assigned, and the tags extracted, the sematic representation and text generation system can automatically produce an XML semantic data structure and generate a text description. Additionally, inference heuristics using the knowledge base ConceptNet are used to infer if the classes and tags describe any additional concepts. To generate the XML representation, the system iterates through a tree element data structure and fills in all relevant tags based on the assigned classes in the context list. An example of the XML representation can be seen in Figure 3. The XML representation has additional meta-information about the semantic inferences, the sensors, the microcontroller device, as well as the raw features information.

Conclusion

In this work, a methodology for semantic content enrichment for embedded devices connected to the internet of things has been proposed. SVM and Naïve Bayes classifiers were used to predict if samples collected by the devices detected pollution. The results obtained in this initial study show promise for this type of classification task. However, it is important to note that future work must include more polluted samples to obtain a better representation of the distribution in the polluted data. Results of the feature analysis using the information gain ranking technique indicate that the MQ131 and MQ7 sensors have the highest contribution to classification accuracy. Additionally, pressure and dew point data have a higher contribution than temperature and precipitation.

To address the issue of missing data, regression techniques were applied. Results from the regression analysis show promising results and indicate that SVR can improve results over regular linear regression. These initial results indicate that the training set with sensor data and API data as features can be useful for automatic content enrichment of embedded systems devices connected to the internet of things.

Future work will focus on collecting additional features and on predicting other classes from the features. These new classes could help to determine other locations such as is on highway, located in city, located in farms, etc. Additional samples will be collected to create a larger corpus and to establish a data set with a smaller class imbalance. Another area to explore is the incorporation and use of other sensor information and tags from linked sites which can help to improve the content enrichment of the embedded devices. Syntactic grammars could also be implemented in the system to provide a greater variety of natural language text descriptions based on semantic analysis of information collected from embedded system devices.

References

Klusch, M., Fries, G., Sycara, K. 2009. OWLS-MX: A Hybrid Semantic Web Service Matchmaker for OWL-S Services. Journal of Web Semantics, 7(2)

Thirunarayan, K., Pschorr, J. 2009. Semantic Information and Sensor Networks. Proceedings of the 24th Annual ACM Symposium on Applied Computing, 1273-1274

Bröring A., Echterhoff J., Jirka S., Simonis I., Everding T., Stasch C., Liang S., Lemmens R.. 2011. New Generation Sensor Web Enablement. Sensors., 11(3),2652-2699.

Roman, R., Lopez, J. 2009. Integrating wireless sensor networks and the internet: a security analysis. Internet Research, 19(2), 246 – 259

Dlodlo, N. 2012. Adopting the Internet of Things technologies in environmental management in South Africa. 2nd International Conference on Environment Science and Engineering. ICESE 2012, Bangkok, Thailand, April 7-8, 45-55

Ramya, V., Palaniappan, B., Karthick, K., Prasad, S., 2012. Embedded System for Vehicle Cabin Toxic Gas Detection and Alerting. Procedia Engineering, 30, 869-873

Havasi, C.; Speer, R.; Alonso, J. 2007. ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. 22nd Conference on Artificial Intelligence

Vapnik, V., Cortes, C. 1995. Support-Vector Networks. Machine Learning, vol. 20, no. 3, pp. 273-297, (1995)

Burges, C.J. 1998. A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, vol. 2, pp. 121-167.

Yang, Y., Pederson, J. 1997. A Comparative Study on Feature Selection in Text Categorization. In Proceedings of the Fourteenth International Conference on Machine Learning, pp. 412-420

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. 2002.SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 341–378.

Smola, A., Scholkopf, B. 2004. A Tutorial on Support Vector Regression. Statistics and Computing, Vol. 14, pp. 199-222.