Towards IoT data classification through semantic features

Mário Antunes, Diogo Gomes, Rui Aguiar

Instituto de Telecomunicações
Universidade de Aveiro
Aveiro, Portugal

Abstract

The technological world has grown by incorporating billions of small sensing devices, collecting and sharing huge amounts of diversified data. As the number of such devices grows, it becomes increasingly difficult to manage all these new data sources. Currently there is no uniform way to represent, share, and understand IoT data, leading to information silos that hinder the realization of complex IoT/M2M scenarios. IoT/M2M scenarios will only achieve their full potential when the devices work and learn together with minimal human intervention. In this paper we discuss the limitations of current storage and analytical solutions, point the advantages of semantic approaches for context organization and extend our unsupervised model to learn word categories automatically. Our solution was evaluated against Miller-Charles dataset and a IoT semantic dataset extracted from a popular IoT platform, achieving a correlation of 0.63.

Keywords: IoT, M2M, context information, semantic similarity

1. Introduction

With the advent of the Internet of Things (IoT) [1], an increasing number of devices has been equipped with sensing and processing capabilities. These allow them to communicate with each other, and even with services on the Internet, to

Email addresses: mario.antunes@av.it.pt (Mário Antunes), dgomes@av.it.pt (Diogo Gomes), ruilaa@av.it.pt (Rui Aguiar)

accomplish a given objective. A major component of this connectivity landscape is machine-to-machine communications [2]. M2M generally refers to information and communication technologies able to measure, deliver, digest and react upon information autonomously, *i.e.* with none or minimal human interaction.

Context-awareness is an intrinsic property of IoT scenarios. The data gathered by these devices has no value in its raw state, it must be analyzed, interpreted and understood. As discussed in [3], an entity's context can be used to provide added value: improve efficiency, optimize resources and detect anomalies. Context-awareness computing plays an important role in tackling this issue [4]. However, recent projects follow a vertical approach [5, 6, 7], with devices/manufacturers not sharing context information, or sharing it with a different structure, leading to low interoperability and information silos respectively. This has hindered interoperability and the realization of powerful IoT/M2M scenarios. Another important issue is the need for a way to manage, store and process such diverse machine made context information, unconstrained and without limiting structures.

Being able to gather data from multiple sources, analyse and understand the data, and discover new patterns and relations will be fundamental to develop and deploy complex IoT and M2M scenarios. Thus, in our view, the full potential of IoT/M2M scenarios can only be achieved when we overcome the previous limitations. However, the potential and definitions of context information [8, 9] is so broad that any information related to an entity can be considered context information. These definitions also do not provide any insight about the structure of context information. Currently there is no uniform way to share/manage vast amounts of IoT information. It is possible (but unlikely) that in the future a context representation standard will be widely adopted.

In our approach we accept the diversity of context representation as a consequence of economic pressures, and have developed concepts that excel in these environments. In previous works we proposed a d-dimension organization model [3] and semantic features specifically for IoT [10]. We extend our semantic model to support multiple word categories and devised an unsupervised learning method to learn word categories automatically from public Web Services. Our new model was tested against Miller-Charles dataset and a IoT semantic dataset (extracted from a popular IoT platform) achieving a correlation of 0.64. Apart from context-aware applications and IoT/M2M scenarios, several other areas benefit from semantic based context organization. For example these methods could provide a decisive contribution towards the exploration of name-based information centric network architectures in IoT environments [11].

The remainder of this paper is organized as follows. In Section 2 we discuss the limitations of conventional databases and analytical tools when dealing with IoT information. The advantages of semantic features and similarity approaches are detailed in Section 3. In Section 4 we present the most relevant methods for estimating semantic similarity. We detail our semantic model and the devised unsupervised learning method in Section 5. Section 6 contains implementation details of our prototype. The results of our evaluation are in Section 7. Finally, the discussion and conclusions are presented in Section 8.

2. Dealing with IoT data

In order to develop and deploy complex IoT/M2M scenarios we need to address the issues regarding storing, analyzing and understanding IoT data. However, correctly managing IoT data has become a difficult task to accomplish. The volume and diversity of data puts a toll on conventional storage and analytical tools, restricting and limiting the realization of complex IoT/M2M scenarios. Due to the volume and lack of formal representation, IoT data can be characterized as a combination of the unstructured data and Big Data paradigms. These paradigms are inherently connected, and are one of the factors that led to the advent of NoSQL databases [12, 13].

This insight highlights one of the limitations of current technology when dealing with massive unstructured data. Relational databases rely on predefined representations and *a priori* relations in order to correctly store and retrieve information. That is rather difficult to accomplish when the data is mostly unstructured, as is the case of IoT data. NoSQL databases relax some constraints and are good alternatives to several workloads and even small IoT scenarios. However, they lack advanced query capabilities, restricting the discovery of information and complex patterns [3].

The limitations are not purely technological. Even if we were able to store and query all data gathered by IoT devices, we would still need methods to organize, analyse and discover relevant relations between data sources and target functions. Most analytical tools rely on *a priori* relations or on a human to analyse data. Both approaches impose some latent knowledge to the underlying model, this type of model is called top-down classification. Top-down classification limits the dimension along which one can make distinctions, and local choices at the leafs are constrained by global categorizations in the branches. It is therefore inherently difficult to put things in their hierarchical places, and the categories are often forced. For illustration lets us considers the following example. The information gathered from an accelerometer inside a vehicle can be used by city officials to detect potholes and other anomalies on the road. But can also be used by policeman to detect dangerous manoeuvres and behaviours, a complete orthogonal classification structure. These examples illustrate how difficult it can be to define *a priori* relations in complex environments.

Some authors [14, 15, 16] point out that probabilistic models based on bottom-up characterization produce better results than binary schemes based on top-down classification. Based on this approach we have devised a bottom-up model to organize context information without enforcing a specific representation. Our organization model is divided into two main parts, as depicted in Figure 1.

The first part is composed by two components that represent the structured part of our model and account for the source ID and fixed d-dimensions respectively. These d-dimensions allow human users to select information based on time, location or even other dimensions, and can be understood as an OLAP cube helping in the process of filtering information. The second part represents machine learning features, that can be used to find similar or related sources of data. Up until known we have worked on semantic [17] and stream features [18]. In

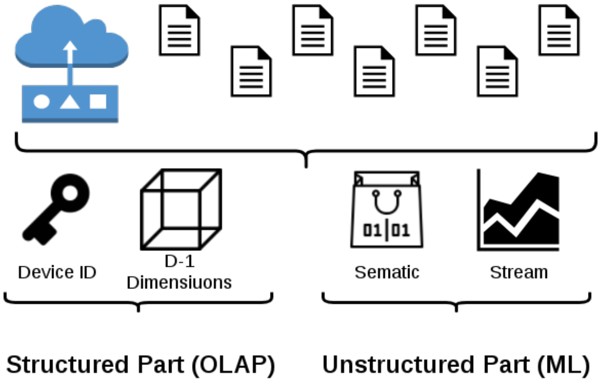

Figure 1: Context organization model based on semantic and stream similarity.

this paper we focus on semantic features, specifically on extending our semantic model to account for multiple word categories.

3. Semantic Features for IoT

Semantic distance/similarity is a property of lexical units, typically between words but this notion can be generalized to larger units such as phrases, sentences, etc. Two words are considered semantically close if there is a lexical semantic relation between them. There are two types of lexical relations: classical relation (such as synonyms, antonyms and hypernymy) and ad-hoc non-classical relation (such as cause-and-effect). If the closeness in meaning is due to a certain classical relation, then the terms are said to be semantically **similar**. On the other hand, semantic **relatedness** is the term used to describe the more general form of semantically closeness caused by any semantic relation. For instance the nouns *liquid* and *water* are both semantically similar and related, whereas the nouns *water* and *boat* are semantically related but not similar.

Semantic features allow us to estimate similarity between concepts (formal discussion in Section 5). This similarity allow us to organize, extract and cluster information based on concepts and not on sub-strings nor regular expressions. In other words, the devices are able to autonomously learn concepts and not only strings. These concepts provide latent knowledge to the underlying information and do not depend on human users or context representation. This is specially important for IoT/M2M scenarios. IoT/M2M devices share a diverse amount of data. We can classify the data into two different categories: semantically rich and poor. In order to better understand these concepts let us consider the following example. A sensor node in a greenhouse measures 6 variables: air and soil temperature, air and soil humidity, CO2 and leaf wetness. The node can periodically share the measurements individually or grouped in a single file. Each document shared in the first option is semantically poor. Based on the semantic value of its attributes it is quite difficult to associate the greenhouse concept with each stream individually. By contrast, a single document with all the attributes is closer to the greenhouse concept, and is semantically rich.

We can improve our IoT/M2M data organization based on this observation. Through semantic methods [19, 20] it is possible to learn higher level concepts from semantically rich documents. Moreover, these high level concepts can be propagated to other data sources based on other features (e.g. stream similarity [18]).

4. Background and Related Work

There are three major types of semantic measures: i) lexical-resource-based measures that rely on manually created resources such as Wordnet, ii) corpus-based measures that rely only on co-occurrence statistics from large corpora, iii) hybrid measures that are distributional in nature, and also exploit information from a lexical resource.

Lexical-resource-based measures rely on manually created and annotated lexical resources, such as WordNet [21], to determine the distance between two words. WordNet is a curated hierarchical network of nodes (taxonomy), where each node represents a fine-grained concept or word-sense. An edge between two nodes represents a lexical semantic relation such as hypernymy or troponymy. WordNet interlinks not just word forms (strings of letters) but specific senses of words. As a result, words that are found in proximity to one another in the network are semantically related. Several authors proposed semantic measures based on WordNet [22, 23, 24].

Semantic measures can only be used in languages that have (a sufficiently developed) WordNet. However, creating and maintaining lexical databases is a tedious task that requires human interaction. Furthermore, updating a lexical resource is expensive and there is usually a lag between the current state of language usage/comprehension and the resource representing it. For example, due to funding and staffing issues the WordNet project is no longer accepting comments and suggestions[1]. Due to these limitations, several authors proposed methods for large-scale acquisition of lexical knowledge, such as KnowNet [25] and BabelNet [26]. KnowNet is an extensible, large and accurate knowledge base, which as been derived by semantically disambiguating small portions of Topic Signatures [27] acquired from the Web. BabelNet is a very large, wide-coverage multilingual semantic network. It combines lexicographic and encyclopaedic knowledge from WordNet and Wikipedia.

Besides these, several other methods exist to build large semantic networks. However, they rely on some sort of structured information, most of them maintained by human users. For example, BabelNet relies on WordNet and Wikipedia, while KnownNet relies on Topic Signatures. Although the information exchanged in IoT/M2M scenarios is limited in vocabulary, usually consists of very specialized words associated with specific fields, topics and contexts. As a consequence, the lexical resource may not contain the correct vocabulary or even the relevant associations between the words.

Strictly corpus-based measures rely on the hypothesis that words with similar contexts tend to be semantically close [28, 29]. The set of contexts of each target word u is represented by its distributional profile, the set of words that tend to co-occur with u within a certain distance, along with numeric scores signifying this co-occurrence tendency with u. Measures such as cosine and α-skew divergence [30] are used to determine how close two distributional profiles

[1] http://wordnet.princeton.edu/wordnet/

are. These methods are very appealing because they rely solely on raw text, however they tend to perform poorly when compared with lexical-resource-based measures.

These methods do not required a lexical-resource, but require a large corpus with representative usages of the target words. Due to the poor vocabulary present in M2M scenarios, the corpus made up from the information shared by M2M devices is not suitable to learn distributional profiles. Our previous solution [10] minimizes this issue using public web services to gather corpus. It is important to mention that the primary objective of this work is to develop semantic features and metrics that are suitable for IoT/M2M scenarios. Devices in M2M networks may not have enough processing power or memory to analyse large corpus of raw text. We are developing methods that extract reliable distributional profiles with the least amount of raw text.

Another important issue is the sense-conflation problem. The distributional profile of a target word u conflates information about potentially many senses of u. Some authors [31] proposed hybrid measures that are distributional in nature but also rely on lexical resources to exploit the manually encoded information to overcome the sense-conflation problem. For example, they extract distributional profiles for each sense of a word. They use categories from a Roget-style thesaurus [32, 33, 34] as coarse sense or concepts. A Roget-style thesaurus classifies all word types into approximately 1000 categories. Words with more than one sense are listed in more than one category. Each category has a head word that best represents the meaning of all the words in that category. The distance between words u and v is the closest distance between all their possible senses. Hybrid methods require a lexical resource, as such these methods have exactly the same disadvantages as lexical-resource-based measures for M2M scenarios.

It is worth mentioning that the previous solutions provides very accurate methods to estimate semantic similarity. However, those solutions rely heavily on structured information or well maintained corpus. The ever-increasing number of IoT/M2M devices, scenarios and applications makes it very difficult to build and maintain semantic networks or clean relevant corpus. In this paper we proposed an unsupervised learning method to identify categories without the need of a Roget-style thesaurus. The method we propose in this paper trades accuracy with flexibility and simplicity. Our solution does not require a specialized (large) corpus, and learns distributional profiles through web services using minimal textual information. This is also the reason we do not evaluate our solution against the ones discussed here, we would be using strategies outside their design constrains.

5. Distributional profiles from Public Web Services

Given a target word u we use public web services, namely search engines, to gather a potentially relevant corpus and extract the word u distributional profile. The profile is built based on proximity, which means if a word w is within the neighbourhood of a target word u it is properly processed and extracted. This distributional profile of a word (DPW) is defined as

$$DPW(u) = \{w_1, f(u, w_1); ...; w_n, f(u, w_n)\} \tag{1}$$

where u is the target word, w_i are words that occur with u and f stands for co-occurrence frequency (can be generalized for any strength of association metric). A distributional profile can also be interpret as a vector that represents a point in high dimensional space, each word w_i represent a dimension and $f(u, w_i)$ represents its value in that dimension. From this point onward we will refer to words inside a DPW as dimensions. We evaluate the similarity between two DPW with cosine similarity:

$$S(u,v) = cosine(u,v) = \frac{\sum_{i=1}^{n} f(u,w_i) \times f(v,w_i)}{\sqrt{\sum_{i=1}^{n} f(u,w_i)^2} \times \sqrt{\sum_{i=1}^{n} f(v,w_i)^2}} \tag{2}$$

Other similarity measures can be used, however cosine is invariant to scale, which means it does not take into account the vector's magnitude, only their direction. This property is import for unbalanced corpus, such as corpus in M2M scenarios or corpus gathered from web services (due to the ranking algorithms used by web-services).

Although public web services offer some important advantages, they also have some disadvantages. Distributional profiles can be noisy, and contain several dimensions with low relevance. A dimension with low relevance is a dimension with a low value of co-occurrence frequency ($f(u, w_n)$). The combined weight of several low relevance dimensions can change the direction of the word vector and damage the cosine similarity. Also, a profile can contain several senses of the target word (sense-conflation). Multiple words senses in a single profile may also change the word vector direction and decrease accuracy, limiting the potential of this method.

We developed two filters to reduce DPW's unwanted dimensions. The first filter uses stemming to merge words that have the same stem, minimizing issues with, *e.g.*plural words.

The second filter uses statistical significance to discard low relevant dimensions, and it is based on the p-value statistical significance test. We defined the null hypothesis (H_0) as the dimension generated randomly and the alternative hypothesis (H_a) as the relevant dimension. Each dimension value is compared with a IID (Independent and Identically Distributed) model, where all the words that compose the distributional profile have exactly the same probability of appearing. If the dimension's value is high compared with the IID model, then we discard the null hypothesis and assume that the dimension is relevant. Every time the DPW learning method finds the target word u, it extracts the corresponding neighbourhood. We count the number of distinct words extracted from the neighbourhood (named V) and the total number extracted words (named P). Assuming that each words has the same probability of appearing, the probability of a word appearing exactly k times is express as follows:

$$p(k) = \frac{\binom{P}{k} \times (V-1)^{P-k}}{V^P} \tag{3}$$

Based on the previous expression we can compute the probability of a word appearing at least k times as follows:

$$p(\geq k) = 1 - \sum_{i=1}^{k} \frac{\binom{P}{i} \times (V-1)^{P-i}}{V^P} \tag{4}$$

Using the previous expression we compute the probability for each dimension, if the result is greater than a predefined p, the dimension is discarded[2].

These filters minimize the impact of low relevant dimensions, improve accuracy and processing speed. However, they do not minimize the effect of sense-conflation, where a distributional profile can learn dimensions from multiple word senses. In order to minimize this issue we propose using clustering on the distributional profile to identify categories/word senses. The rational is that dimensions belonging to the same category are closer to each other than words from other categories. Clustering methods require a distance metric in order to group similar elements. From this point we will discuss similarity metric, knowing that a similarity can be converted to a distance using the following expression

$$D(u, v) = 1 - S(u, v) \tag{5}$$

Since we are dealing with semantic similarity, a natural solution is to use cosine similarity over each dimension's distributional profile. However, as stated previously, profiles extracted from Web Services may contain multiple senses of the target word and low relevance dimensions. Alternatively we propose using co-occurrence frequency as an estimator of similarity metric. Co-occurrence does not take into account the neighbourhood of a target word, preventing the previously stated issue. In Section 7 we evaluate the performance of both metrics.

These clusters do not represent word senses from a Roget-style thesaurus. Which means that there is not a one-to-one relation between the clusters and a word in a thesaurus. Conceptually the clusters are more similar to categories in latent semantic analysis, and may not have a correspondence to our human perception. Since a cluster may not represent a classical word sense, from this point onward we will refer to them as categories. One implication of this statement is that some clusters represent high relevance categories, while others

[2] In the evaluation we used $p = 0.01$, which means 99% confidence of being a true relevant dimension.

represent low relevance categories. Consider the following scenario, two target words u and v are not related, but may end up with the same low relevance category. This category will match and produce a false positive.

In order to minimize this issue our model incorporates an affinity value between the target word and each category, can be understood as a bias, it measures the natural tendency from a word to be used as a specific category. The affinity is computed as the average similarity between the target word and all the cluster's elements. After the clustering and computing the affinity of the target word to each cluster, the distributional profile of multiple words categories ($DPWC$) is extracted from the DPW and grouped according to the clusters obtained. After computing all the affinity values, they are normalized between $]0, 1]$ with the following expression

$$a'_i = \frac{a_i}{max(a)} \tag{6}$$

The profile is defined as follows:

$$DPWC(u) = \left\{ \begin{matrix} a_1; \{w_1, f(u_1, w_1); ...; w_n, f(u_1, w_n)\} \\ ... \\ a_n; \{w_1, f(u_c, w_1); ...; w_n, f(u_c, w_n)\} \end{matrix} \right\} \tag{7}$$

where u is the target word, w_i are words that occur with u in a certain category, f stands for co-occurrence frequency and a_i is the affinity between u and a word category.

Finally, the similarity between two $DPWC$ is given by the following expression

$$S(u, v) = max(cosine(u_c, v_c) \times (a_{u_c} + a_{u_v} \big/ 2)) \tag{8}$$

where u_c and v_c represent a specific category from u and v respectively and a represents the category's affinity. Our final similarity measure is the maximum similarity between all the possible categories weighted by the average category's affinity. By incorporating affinities our model minimizes the effect of low relevance categories.

6. Implementation

In this section we discuss some relevant details about our prototype implementation. Our prototype is divided into 5 different components as depicted in Figure 2. All the components were written in Java.

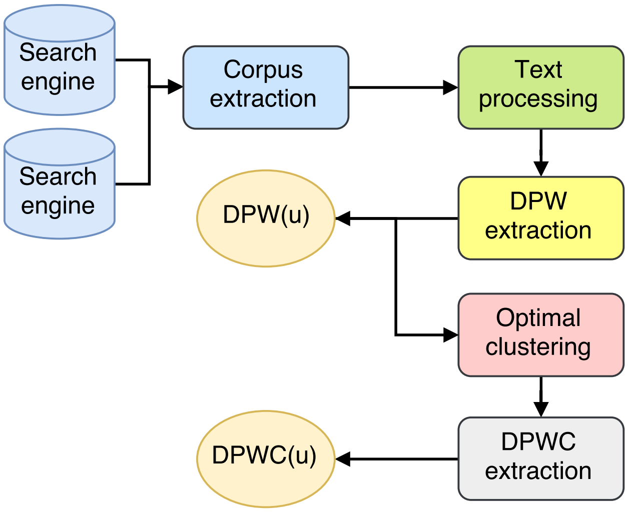

Figure 2: Proposed DP extraction system's architecture.

The first component (corpus extraction) bridges our solution with web search engines. Given a target word u our prototype uses web search engines to extract its $DPW(u)$ and $DPWC(u)$. It can be used with any search engine, and currently it uses three: Faroo[3], Yacy[4]and Searx[5]. This component basic function is to extract a corpus from search engines. The corpus is composed of snippets returned by searching for the target word. In a previous work [10] we compared the impact of using only snippets against the full web-pages. We observed that snippets contain enough information to build reliable DPWs.

The second component (text processing) implements a preprocessing pipeline

[3] http://www.faroo.com/hp/api/api.html

[4] http://yacy.net/en/index.html

[5] https://searx.me/

that cleans the corpus and divides it into tokens. The various spaces of the pipeline are depicted in Figure 3. First the snippets are tokenized and the resulting tokens are filtered using a stop word filter. Stop words are deemed irrelevant because they occur frequently in the language and provide little information. We used the MySQL stop word list[6]. For the exact same reason we also remove tokens that are too big or too small: any token with less than 3 or more than 14 (9 being the average word length in English) characters were removed from the pipeline.

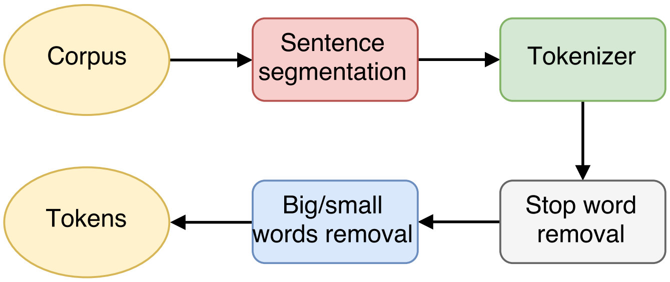

Figure 3: Text processing pipeline.

The *DPW* extraction component analyses the output of the pipeline and extracts the *DPW* of the target word u. This component also applies the filters mentioned in Section 5 that minimize the issue with low relevant dimensions. After extracting and optimizing the *DPW*, we cluster the profile dimensions based on cosine and co-occurrence similarity. K-means++ [35] was used to cluster the profile dimensions and identify the categories. K-means++ is a variant of the well-known and widely used K-means that improves both speed and accuracy.

These types of algorithms have a drawback, as they require the number of clusters *a priori*. Normally gap statistics [36] is used to identify the ideal number of clusters from a possible range. However, this method requires generating reference features based on the elements to compare the clustering with a

[6] https://dev.mysql.com/doc/refman/5.1/en/fulltext-stopwords.html

uniform sample. DPWs are highly dimensional by nature, meaning that using this method is quite expensive. As an alternative, we used the framework proposed in [37], as it only requires the number of dimensions.

Finally, the $DPWC$ component uses the DPW and the clusters to return the $DPWC(u)$ of the target word, this component also computes the affinity between the target word and each category.

7. Performance evaluation

We evaluate our model against Miller-Charles dataset [38], the reference dataset for semantic similarity evaluation. It is composed of 30 word-pairs rated by a group of 38 human subjects. The word pairs are rated on a scale from 0 (no similarity) to 4 (perfect synonymy).

To the best of our knowledge, there is no semantic dataset specifically for IoT/M2M available. In order to evaluate our semantic features against IoT vocabulary, we devised one. We mined a popular IoT plaform[7] to extract the most common used terms (ranked by term frequency). The 20 most used terms were collected and organized into 30 word pairs. Each pair was rated on a scale from 0 to 4 by five fellow researchers. Although not as comprehensive as the Miller-Charles dataset, our still reach 0.8 correlation amongst human classification. In a future work we intend to further explore and improve our dataset. The final similarity of each pair is the average of the previous stated rates. This dataset is publicly available[8] and can be used by other researchers.

Correlation between sets of data is a measure of how well they are related. The correlation r can range from -1 to 1. An r of -1 indicates a perfect negative linear relationship between variables, an r of 0 indicates no linear relationship between variables, finally and an r of 1 indicates a perfect positive linear relationship between variables. In short, the highest correlation indicates the most accurate solution.

[7] ThingSpeak: https://thingspeak.com/

[8] https://atnog.av.it.pt/ mantunes/semantic/

Normally, Pearson correlation is used to evaluate distance measures against the ground truth (human classification). One advantage of Pearson correlation is its independence from scale and distance metric. The rationale is that even in different scales if the linear correlation between the ground truth and the similarity metric is high, then the performance is also high. Our model uses unsupervised learning methods to identify categories and improve accuracy. However, the improvement is not the same to each word pair in the dataset, damaging the linear correlation. As such, we also evaluate our model using mean squared error (MSE), a typical performance metric used in regression problems. It is worth mentioning that in order to use MSE metrics we had to normalize the dataset score.

Finally, we evaluated the performance of $DPW(u)$, $DPWC(u)$ with and without affinity for different neighbourhood dimensions and two distinct clustering metrics: one based on co-occurrence and other on cosine similarity. We tested our models on corpus formed from the top 300 snippets returned by three search engines: Faroo, Yacy and Searx.

The results of the evaluation using Miller-Charles dataset are listed in Table 1 and Table 2. The optimal neighbourhood's size appears to be 7. $DPWC$ with affinity outperforms the previous model (DPW) consistently on both metric (Pearson and MSE). This is expected as the affinity value allows the model to minimize the impact of low relevance categories. Clustering based on co-occurrence outperforms clustering based on cosine similarity. Again, this is to be expected since the distributional profiles contain some unwanted dimensions and damage the cosine similarity accuracy. Although co-occurrence similarity is simpler in nature (expresses little information regarding semantic similarity), is robust against unwanted dimensions.

The results of the evaluation using the IoT dataset are listed in Table 3 and Table 4. Again, $DPWC$ with affinity outperforms the previous model (DPW) consistently on both metric. However, this dataset exposes the drawbacks of clustering based on cosine similarity and $DPWC$ without affinity. We can see that clustering based on cosine similarity does not outperform our previous

Table 1: Performance evaluation on Miller-Charles dataset (cosine distance)

	Neighborhood size					
Methods	3		5		7	
	Pearson	MSE	Pearson	MSE	Pearson	MSE
DPW	0.32	0.31	0.37	0.29	0.45	0.29
$DPWC$	0.36	**0.23**	0.30	0.25	0.31	0.27
$DPWC_{Aff}$	**0.47**	0.24	**0.45**	**0.20**	**0.63**	**0.15**

Table 2: Performance evaluation on Miller-Charles dataset (co-occurrence distance)

	Neighborhood size					
Methods	3		5		7	
	Pearson	MSE	Pearson	MSE	Pearson	MSE
DPW	0.32	0.31	0.37	0.29	0.45	0.29
$DPWC$	0.40	**0.21**	0.24	0.26	0.29	0.29
$DPWC_{Aff}$	**0.43**	0.22	**0.55**	**0.19**	**0.63**	**0.15**

model, especially at higher value neighbourhoods. Similarly, $DPWC$ without affinity is outperformed by all the other metrics, small cluster with low relevance are being matched producing a false negative.

As discussed in Section 4, other semantic methods achieved higher accuracy. Nonetheless, our model outperforms some methods that also rely on web-engines (a comparative study of semantic similarity can be found in [39]). We cannot draw a direct comparison, since our model was designed with a specific set of constrains (intended to be a viable solution for IoT). In order to highlight the performance improvement of word category extraction we plotted the best results from both datasets in Figure 4. Similarity based on DPW tends to low values (similarity values close to zero), hindering the ability to make binary choices (similar/not similar concepts). On the other hand, similarity based on $DPWC$ with affinity do not cluster together close to zero, being correctly spaced. Our

Table 3: Performance evaluation on IoT dataset (cosine distance)

	Neighborhood size					
Methods	3		5		7	
	Pearson	MSE	Pearson	MSE	Pearson	MSE
DPW	0.27	0.25	**0.37**	0.23	**0.33**	0.24
$DPWC$	0.15	0.30	-0.01	0.34	-0.04	0.33
$DPWC_{Aff}$	**0.34**	**0.17**	**0.37**	**0.13**	0.24	**0.15**

Table 4: Performance evaluation on IoT dataset (co-occurrence distance)

	Neighborhood size					
Methods	3		5		7	
	Pearson	MSE	Pearson	MSE	Pearson	MSE
DPW	0.27	0.25	0.37	0.23	0.33	0.24
$DPWC$	0.05	0.31	0.07	0.32	-0.11	0.13
$DPWC_{Aff}$	**0.41**	**0.25**	**0.46**	**0.13**	**0.55**	**0.12**

$DPWC$ model does not only improve accuracy, it also aids binary systems by providing a larger margin to make a decision.

8. Conclusions

The number of IoT devices is increasing at a steady step. Each one of them generates massive amounts of diverse data. However, each device/manufactures share context information with different structure, hindering interoperability in IoT and M2M scenarios.

In this paper we discussed the limitations of conventional storage and analytical tools, and pointed out the advantages of bottom-up context organization model. We also discussed semantic approaches specifically designed for IoT/M2M scenarios. Our semantic model was extended to support multiple word categories and a new unsupervised learning method was designed. Distributional profiles

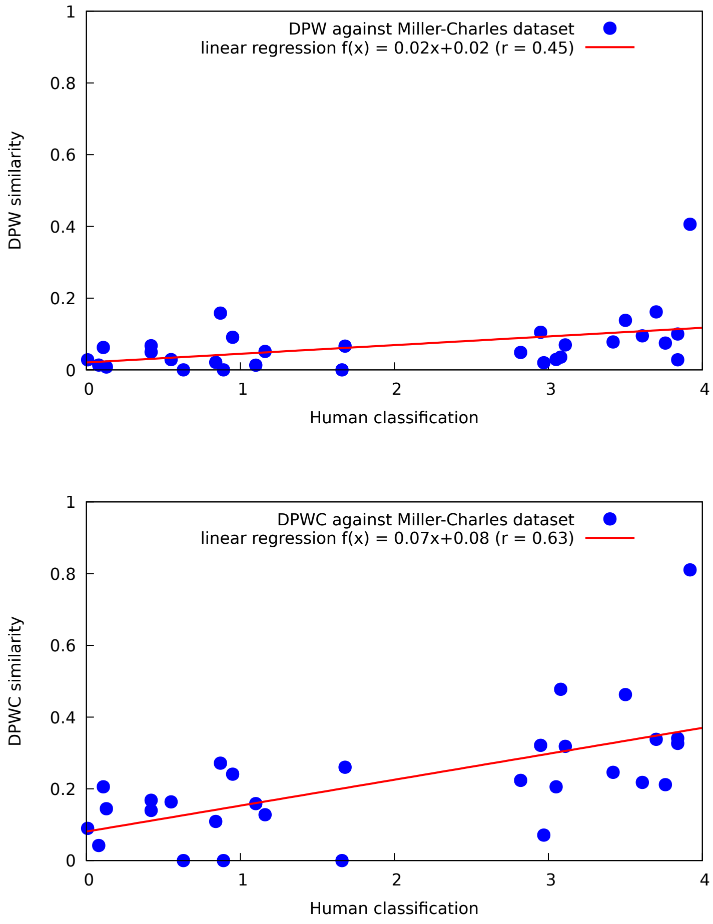

Figure 4: Visual comparison between the DPW and $DPWC$ similarity, using the Miller-Charles dataset.

extracted from web services may contain noisy dimensions and several senses of the target word (sense-conflation). These issues decrease accuracy, and limit the potential of this model. Our learning method minimizes these issues through dimensional reduction filters and clustering.

Our solution was evaluated against Miller-Charles dataset [38] and an IoT semantic dataset, achieving a correlation of 0.63. There is still room for improvement, hypernyms can be used to learn more abstract dimensions improving performance. Non-negative matrix factorization can also be used to discover latent semantic information in distributional profiles and increase accuracy. Furthermore, a recursive method can be used to evaluate distributional profiles, each dimension is evaluated using semantic distances instead of string matching. We intent to explored several of the previous mentions optimizations and improve our model. Nevertheless, our model was able to learn distributional profiles from a small corpus, achieving a relative high accuracy on both datasets.

Acknowledgement

This work was partially supported by European Regional Development Fund (ERDF) under grant agreement No. 7678 (Ref. POCI-01-0247-FEDER-007678) entitled “SGH - SMART GREEN HOME”, and research grant SFRH/BD/94270/2013.

[1] F. Wortmann, K. Flüchter, et al., Internet of things, Business & Information Systems Engineering 57 (3) (2015) 221–224. `doi:10.1016/j.comnet.2010.05.010`.

[2] K.-C. Chen, S.-Y. Lien, Machine-to-machine communications: Technologies and challenges, Ad Hoc Networks 18 (2014) 3–23. `doi:10.1016/j.adhoc.2013.03.007`.

[3] M. Antunes, D. Gomes, R. L. Aguiar, Scalable semantic aware context storage, Future Generation Computer Systems 56 (2016) 675–683. `doi:10.1016/j.future.2015.09.008`.

[4] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: A survey, IEEE Communications Surveys Tutorials 16 (1) (2014) 414–454. `doi:10.1109/SURV.2013.042313.00197`.

[5] R. Fantacci, T. Pecorella, R. Viti, C. Carlini, Short paper: Overcoming iot fragmentation through standard gateway architecture, in: 2014 IEEE World Forum on Internet of Things (WF-IoT), 2014, pp. 181–182. `doi:10.1109/WF-IoT.2014.6803149`.

[6] J. Robert, S. Kubler, Y. L. Traon, K. FrÃďmling, O-mi/o-df standards as interoperability enablers for industrial internet: A performance analysis, in: IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, 2016, pp. 4908–4915. doi:10.1109/IECON.2016.7793138.

[7] S. K. Datta, C. Bonnet, R. P. F. D. Costa, J. HÃďrri, Datatweet: An architecture enabling data-centric iot services, in: 2016 IEEE Region 10 Symposium (TENSYMP), 2016, pp. 343–348. doi:10.1109/TENCONSpring.2016.7519430.

[8] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, P. Steggles, Towards a better understanding of context and context-awareness, in: Proc. of the 1st international symposium on Handheld and Ubiquitous Computing, 1999, pp. 304–307. doi:10.1007/3-540-48157-5_29.

[9] T. Winograd, Architectures for context, Hum.-Comput. Interact. 16 (2) (2001) 401–419. doi:10.1207/S15327051HCI16234_18.

[10] M. Antunes, D. Gomes, R. L. Aguiar, Semantic features for context organization, in: Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on, 2015, pp. 87–92. doi:10.1109/FiCloud.2015.103.

[11] J. Quevedo, M. Antunes, D. Corujo, D. Gomes, R. L. Aguiar, On the application of contextual iot service discovery in information centric networks, Computer Communicationsdoi:10.1016/j.comcom.2016.03.011.

[12] N. Leavitt, Will nosql databases live up to their promise?, Computer 43 (2) (2010) 12–14. doi:10.1109/MC.2010.58.

[13] R. Cattell, Scalable sql and nosql data stores, SIGMOD Rec. 39 (4) (2011) 12–27. doi:10.1145/1978915.1978919.

[14] C. Shirky, Ontology is overrated: Categories, links, and tags, http://shirky.com/writings/ontology_overrated.html, accessed: 22-07-2013 (May 2005).

[15] G. Avram, At the crossroads of knowledge management and social software, Electronic Journal of Knowledge Management 4 (1) (2006) 1–10.

[16] T. Gruber, Ontology of folksonomy: A mash-up of apples and oranges, International Journal on Semantic Web and Information Systems 3 (2) (2007) 1–11.

[17] M. Antunes, D. Gomes, R. Aguiar, Learning semantic features from web services, in: Future Internet of Things and Cloud (FiCloud), 2016 4rd International Conference on, IEEE, 2016. doi:10.1109/FiCloud.2016.46.

[18] M. Antunes, R. Jesus, D. Gomes, R. Aguiar, Improve iot/m2m data organization based on stream patterns, in: 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, 2017.

[19] S. Pradhan, K. Hacioglu, W. Ward, J. Martin, D. Jurafsky, Semantic role parsing: adding semantic structure to unstructured text, in: Third IEEE International Conference on Data Mining, IEEE Comput. Soc, 2003. doi:10.1109/icdm.2003.1250994.

[20] C.-S. Lee, Y.-F. Kao, Y.-H. Kuo, M.-H. Wang, Automated ontology construction for unstructured text documents, Data & Knowledge Engineering 60 (3) (2007) 547–566. doi:10.1016/j.datak.2006.04.001.

[21] G. A. Miller, Wordnet: A lexical database for english, Communications of the ACM 38 (11) (1995) 39–41. doi:10.1145/219717.219748.

[22] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics, ACL '94, 1994, pp. 133–138. doi:10.3115/981732.981751.

[23] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'95, 1995.

[24] S. Banerjee, T. Pedersen, An adapted lesk algorithm for word sense disambiguation using wordnet, in: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, CICLing '02, 2002, pp. 136–145. doi:10.1007/3-540-45715-1_11.

[25] M. Cuadros, G. Rigau, Knownet: Building a large net of knowledge from the web, in: Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1, COLING '08, Association for Computational Linguistics, 2008, pp. 161–168. doi:10.3115/1599081.1599102.

[26] R. Navigli, S. P. Ponzetto, BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artificial Intelligence 193 (2012) 217–250. doi:10.1016/j.artint.2012.07.001.

[27] C.-Y. Lin, E. Hovy, The automated acquisition of topic signatures for text summarization, in: Proceedings of the 18th Conference on Computational Linguistics - Volume 1, COLING '00, Association for Computational Linguistics, Stroudsburg, PA, USA, 2000, pp. 495–501. doi:10.3115/990820.990892.

[28] J. Firth, A synopsis of linguistic theory 1935-55, Transactions of the Philological Society.

[29] Z. Harris, Mathematical Structures of Language, John Wiley and Son, 1968.

[30] L. Lee, On the effectiveness of the skew divergence for statistical language analysis, in: Artificial Intelligence and Statistics, 2001, pp. 65–72.

[31] Y. Marton, S. Mohammad, P. Resnik, Estimating semantic distance using soft semantic constraints in knowledge-source-corpus hybrid models, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2, EMNLP '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 775–783.
URL http://dl.acm.org/citation.cfm?id=1699571.1699614

[32] P. Roget, Roget's International Thesaurus, 1st edition, Cromwell, New York, 1911.

[33] W. Hüllen, A history of Roget's Thesaurus: Origins, development, and design, Oxford University Press Oxford, 2004. doi:10.1093/acprof:oso/9780199254729.001.0001.

[34] M. Jarmasz, S. Szpakowicz, Roget's thesaurus and semantic similarity, arXiv preprint arXiv:1204.0245doi:10.1075/cilt.260.12jar.

[35] D. Arthur, S. Vassilvitskii, K-means++: The advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007, pp. 1027–1035.

[36] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (2) (2001) 411–423. doi:10.1111/1467-9868.00293.

[37] D. T. Pham, S. S. Dimov, C. Nguyen, Selection of k in k-means clustering, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 219 (1) (2005) 103–119. doi:10.1243/095440605X8298.

[38] G. A. Miller, W. G. Charles, Contextual correlates of semantic similarity, Language and Cognitive Processes 6 (1) (1991) 1–28. doi:10.1080/01690969108406936.

[39] A. Panchenko, O. Morozova, A study of hybrid similarity measures for semantic relation extraction, in: Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, HYBRID '12, Association for Computational Linguistics, 2012, pp. 10–18.
URL http://dl.acm.org/citation.cfm?id=2388632.2388634