Distributional Semantics and Unsupervised Clustering for Sensor Relevancy Prediction

Myriam Leggieri
Insight Centre for Data Analytics
National University of Ireland, Galway
Email: myriam.leggieri@insight-centre.org

Brian Davis
Insight Centre for Data Analytics
National University of Ireland, Galway
Email: brian.davis@insight-centre.org

John G. Breslin
Insight Centre for Data Analytics
National University of Ireland, Galway
Email: john.breslin@insight-centre.org

Abstract—**The logging of Activities of Daily Living (ADLs) is becoming increasingly popular mainly thanks to wearable devices. Currently, most sensors used for ADLs logging are queried and filtered mainly by location and time. However, in an Internet of Things future, a query will return a large amount of sensor data. Therefore, existing approaches will not be feasible because of resource constraints and performance issues. Hence more fine-grained queries will be necessary. We propose to filter on the likelihood that a sensor is relevant for the currently sensed activity. Our aim is to improve system efficiency by reducing the amount of data to query, store and process by identifying which sensors are relevant for different activities during the ADLs logging by relying on Distributional Semantics over public text corpora and unsupervised hierarchical clustering. We have evaluated our system over a public dataset for activity recognition and compared our clusters of sensors with the sensors involved in the logging of manually-annotated activities. Our results show an average precision of 89% and an overall accuracy of 69%, thus outperforming the state of the art by 5% and 32% respectively. To support the uptake of our approach and to allow replication of our experiments, a Web service has been developed and open sourced.**

Keywords—***Internet of Things, Sensor Network, Sensor Selection, Distributional Semantics, Human Activity Recognition, Unsupervised Learning.***

I. Introduction

The logging of Activities of Daily Living (ADLs) is becoming increasingly relevant due to the large adoption of wearable devices - such as Fitbit, GoPro and Google Glass - and the ubiquity of mostly Internet-connected and sensors-enabled devices - such as in our smartphones and vehicles -. The amount of these devices is purported to reach 50 to 100 billion by 2020 [14] within the Internet of Things (IoT) pheonomenon. At the same time, the sensors embedded in such devices and the devices themselves are resource constrained, with limited memory and battery lifetime. Consequently, the respective queries for collecting collect data from the aforementioned sensors must be optimised. Part of the query optimisation can consist in preselecting only a subset of the available sensors and query them. This subset should include the most relevant sensors for the task at hand. With regards to the ADL logging, the task at hand is in itself the ongoing activity that is being tracked. Identifying activity-wise relevant sensors is challenging because activities vary greatly. They are performed in idiosyncratic ways and run in real-world - not controlled - environments. Consequently predefining which variables to record during each different activity is difficult. For instance, a user may suddenly decide to switch from walking to driving a car. In this case, the system should recognise the necessity to change what to track via sensors and thus which sensors to query. During the walking activity, the relevant sensors may measure breathing, air pollution, noise and temperature. While during the driving activity, relevant sensors may be those monitoring traffic, fog and fuel levels. This paper addresses the automated preselection of the subset of available sensors that are relevant for the ongoing activity. This preselection can then be used as a sensor query optimisation technique. We describe our approach in Section II along with related work in Section III. We describe our evaluation and discuss its results in Section IV. Section III compares ours with previous solutions, while we draw our conclusions and outline future work in Section V.

II. Methodology

We propose an innovative methodology to predict the likelihood of a sensor producing relevant readings for an ongoing activity (see Algorithm 1). First, we query all the readings and metadata of sensors located in a specific location during a specific time range ($A = \{search\_results\}$). The query is constructed using the SPARQL query language which is designed to span across distributed datasets. Our system runs it against a public list of all open sensor datasets, available on the Datahub framework. In doing so, we assume the data is compliant with our Linked Data representation (Section II-B). This supports the inclusion of additional datasets expected to become available as a consequence of the Internet of Things trend.

We consider $B \subseteq A$ is the set of sensors whose readings represent a change in status, e.g., a change in temperature. The set of sensor readings C is such that $\forall x \in B : reading(x) = y \in C$ with the function reading() being $injective \wedge surjective$. Our system then predicts which other sensor $z \in (A \backslash B)$ is likely to produce readings that will be relevant for the current ongoing activity. It obtains the semantic relatedness of each pair (x, z) where $z \in (A \setminus B) \wedge x \in B$ via a web service (Section II-A) that

978-1-4799-5344-8/15/$31.00 ©2015 IEEE

```
Data: Location, TimeRange
Result: Clusters of objects likely to be relevant
        (i.e., used) during the same activity
searchResults = queryDatahub(Location,
TimeRange);
activatedSensors =
getSensorsfromReadings(searchResults);
for sensorX in activatedSensors do
    for sensorY in searchResults do
        if sensorY not in activatedSensors then
            similarity = getESASimilarity(sensorX,
            sensorY);
            addToDistanceMatrix(similarity, matrix)
        end
    end
    clustering(matrix)
end
```

Algorithm 1: Algorithm used in our methodology to predict sensors relevant for an activity.

had previously applied ESA [7] on the English Wikipedia archive dump dated 2013 [17]. Such relatedness is *semantic* or meaningful because it is calculated by considering the pair of sensors as not just mere electronic devices but rather in terms of the (semantic) function that they have from a natural language (human) perspective. For example, a switch sensor attached to a fridge is uniquely identified in terms of its semantics as $< switch, fridge >$ because once it is deployed, the human end user will not be interested in it as an electronic component, but rather as a provider of switch information about the fridge. As a use case, let us consider users who deploy a door switch sensor to monitor information about how many times they go to a fridge during a typical week, for dieting purposes. In our terminology, a sensor is identified by $< op, foi >$, e.g., $< switch, fridge >$, where op is the $observedproperty$, foi is the $featureofinterest$, and together they are referred as the $sensormetadata$. In other words, our methodology leverages on a correspondence between the lexical realisation of sensors and the conceptual objects to which they are attached. Finally, our system collects all sensor similarities in a sparse matrix and runs three different hierarchical clustering algorithms on this matrix. Each resulting cluster corresponds to an activity, and its members are those sensors that will likely sense a change of status relevant for that activity. For example, the fridge switch sensor will likely be relevant whenever the microwave switch has previously sensed a change of status, i.e., fridge switch and microwave switch sensors will be part of the same cluster. The requirements for this methodology are: **1.** one or more sensors that have recently sensed a change in status (e.g., light switched on after it had been switched off) in a specific location; **2.** sensor metadata which must include the sensor's *observed property* and *feature of interest*. The sensor's *observed property* is the property that it is designed to sense; while its *feature of interest* is the object which the observed property belongs to. For example, if a sensor measures the temperature of a microwave, the temperature is the observed property and the microwave is the feature of interest.

A. Distributional Semantics

Distributional semantics is built on the *distributional hypothesis* stating that words that occur in similar contexts tend to have similar meaning [16]. The distributional view on meaning is inherently differential, i.e., the differences of meaning are mediated by differences of distribution. Consequently, Distributional Semantic Models (DSMs) quantify the amount of difference in meaning between linguistic entities. Such differential analysis can be used to determine the semantic relatedness between words [6] which fits with our modelling of activities as sequences of features of interest (i.e., words). Explicit Semantic Analysis (ESA) [7] represents texts by relying on the co-occurrence of words in a large corpus of articles, e.g. Wikipedia. A document containing a string of words is considered as the centroid of the vectors representing its words. Words are represented by vectors of their associations to each concept. Each association is determined using TF-IDF scoring, while cosine similarity measures the semantic relatedness between pairs of words. Given a set of concepts $C_1, ..., C_n$ and a set of associated documents $d_1, ..., d_n$, ESA builds a sparse table T where each of the n columns corresponds to a concept, and each of the rows corresponds to a word that occurs in $\bigcup_{i=1...n} d_i$. An entry $T[i, j]$ in the table corresponds to the TF-IDF value of term t_i in document d_j. The size of the textual corpus on which semantic models rely upon is critical to the quality of the results. This leads to high hardware and software requirements on the implementation side (e.g., the English version of Wikipedia 2013 contains 43 GB of article data). For simplicity, we use the *EasyESA* [5] public instance, a JSON webservice which implements ESA based on Wikiprep-ESA on the English version of Wikipedia 2013. Our query asks for semantic relatedness of pairs of sensors represented as tuples of terms like $< switch, fridge >$.

B. Linked Data for Sensors

We follow a learning-based approach (clustering and distributional semantics) while relying on specification-based sensor representation. Our representation uses the Resource Description Framework (RDF) model (a machine-understandable graph of statements in the form of connected subjects and objects) and align to the Linked Data principles [2] using the Turtle [1] syntax for serialising RDF graphs. This makes our data machine-understandable and supportive of references across multiple data sources. As a workaround to the learning barrier that such annotating and linking process may cause, we implemented *Linked Data for Sensors (LD4S)* [10] a JSON web service which exposes a RESTful API and a SPARQL endpoint to automate the annotation and linking process while facilitating the data browsing and querying. The LD4S SPARQL endpoint is published on Datahub, i.e., the data management platform from the Open Knowledge Foundation, based on the CKAN data management system. This was meant to facilitate the discovery of such a service by third parties. We query this SPARQL service for sensors in a specific range of time, as in Listing II-B (prefixes have been omitted). Once we retrieve the data by querying LD4S, we then calculate the semantic relatedness between couples of sensors by querying EasyESA and then apply a clustering algorithm on the result.

Listing 1. SPARQL query to select sensor readings for a time range.

```
SELECT ?sens ?starttime ?endtime ?obs
?foi ?value ?location
{        ?sens spt:obs ?obs.
         ?ov spt:outOf ?sens;
                  spt:value ?value;
                  spt:tStart ?starttime;
                  spt:tEnd ?endtime.
         ?tsp spt:temporalOf ?sens;
                  ssn:featureOfInterest
                                    ?foi;
                  dul:hasLocation
                           ?location.
FILTER (xsd:dateTime(?starttime)
                  >='2003-03-31T02:00:00Z'
                  ^^xsd:dateTime
         && xsd:dateTime(?endtime)
          <='2003-05-31T01:00:00Z'
                  ^^xsd:dateTime). }
```

C. Unsupervised Hierarchical Clustering

We applied **three** different hierarchical clustering algorithms in our experiments: 1) Weighted Pair Group Method with Arithmetic mean (WPGMA), 2) Unweighted Pair Group Method with Arithmetic mean (UPGMA), 3) Farthest Point Algorithm, also called VoorHees (VH). We chose unsupervised methods because we believe that given the amount of different activities and sensors involved, supervised methods are not likely to scale with the expansion of the Internet of Things phenomenon. In particular, we chose hierarchical clustering because it is the approach that has so far achieved the better precision [8]. We applied UPGMA mainly because it reflects observable similarities between activities by the distance of their semantic distribution. Thus, it fit our goal perfectly. WPGMA was chosen to explore the possibility that the structural subdivision of the objects (i.e., cluster items) had an influence in the belonging of the object to the activity (i.e., cluster). Finally, the application of VH was investigated to explore the possibility that a particular object may be central and more critical in the creation of the clusters.

III. Related Work

Wyatt et al. [18] model activities as sequences of features of interest and consider these as words. They analysed the co-occurrence of the words in the textual content of different websites in order to assemble a Hidden Markov Model [13] for activity inference. We rely on the same model but we use a different methodology. Yet, we compare our results to theirs [18] in the evaluation (see Section IV) since their work is the closest to ours. To the best of our knowledge, among the unsupervised approaches for activity recognition, Kwon et al. [8] achieved the highest precision when the number of activities k is known. For this reason, we also use hierarchical clustering and compare our results to theirs in our evaluation (see Section IV). Leggieri et al. [9] envisioned the usage of digitised common sense to improve reasoning over sensor data, leveraging the Linked Data principles as subsequently realised by [3], [11]. The web services [12], [4] attempt to facilitate the creation of Linked Data for sensors but, unlike us, without allowing the client to customise the link creation.

IV. Evaluation and Discussion

Our goal is to predict which sensors provide relevant information during an activity logging. We compare the list of "relevant sensors per activity" returned by our system with the manual annotations from the public dataset MITes [15]. We pre-processed this dataset (i.e., CSV files of sensor readings and metadata about both sensors and activities) to forward HTTP PUT requests to the LD4S API with the appropriate JSON payload to get a Linked Sensor Data annotation. Based on such comparison, the overall accuracy and precision of our system are calculated when applying either of the clustering algorithms UPGMA, WPGMA or VH. DataHub (see Section II-B) was then queried for all the sensor datasets available thus returning a JSON list of details of these datasets such as their ID, title, tags, license and endpoint URIs. The system filters only those datasets that either have no license or grant an open-access 1) expose a SPARQL endpoint and forward the query in Listing II-B towards each of them.. Our query is forwarded to all the endpoint returned by DataHub, which also contains the LD4S endpoint as well. The results obtained from each endpoint are XML files - as per W3C standard recommendation - that the system merges and parses to distinguish between sensors that sensed a change in status and the others who just happened to share the same location. In this experiment we evaluated the worse case: only one sensor has recently sensed a change in status. The semantic relatedness must be calculated between the higher amount of possible pairs that share the same location at the same time. This is used to fill a distance matrix on which the hierarchical clustering algorithms were applied. In addition to precision and overall accuracy, we also evaluated the performances in terms of execution time for the different HTTP requests, the SPARQL queries, the whole pre-processing step and the overall system.

A. MITes Dataset

Tapia et al. [15] published the MITes dataset from an experiment where human activities were tracked and logged for two weeks. They installed 200 switch sensors deployed on 27 different *features of interest* (FoIs) in two single-person apartments. The sensors were deployed on objects such as drawers, refrigerators, containers, etc. to record opening-closing events (activation deactivation events) as 2 subjects carried out everyday activities. The subjects were manually annotating each activity as in Table IV-A. In our experiment we used the data from both subjects combined together.

B. Similarity Results

We considered the worse case in which only one of the sensors sharing the same location at the same time

TABLE I. Activities labelled in the MITes dataset.

Number of Examples per Class		
Activity	Subject 1	Subject 2
Preparing dinner	8	14
Preparing lunch	17	20
Listening to music	-	18
Taking medication	-	14
Toileting	85	40
Preparing breakfast	14	18
Washing dishes	7	21
Preparing a snack	14	16
Watching TV	-	15
Bathing	18	-
Going out to work	12	-
Dressing	24	-
Grooming	37	-
Preparing a beverage	15	-
Doing laundry	19	-
Cleaning	8	-

range has recently sensed a change in status for the current ongoing activity, while all the other nearby ones which will likely do so in the near future have to be predicted. In this case, given n sensors, the amount of pairs to check for semantic relatedness is the binomial coefficient. In our case since there are 27 different features of interest, there are 27 different types of sensors and 351 distinct pairs. Even though the binomial coefficient grows quickly, it only depends on the amount of features of interest rather than on the amount of actually deployed sensors. At the same time, the amount of ICOs is expected to grow but the amount of ”types” of sensors is not, since there is only so much in the real world that can be monitored by sensors. Our method then is not expected to hinder the system from scaling during the Internet of Things expansion. The growth of time cost is analysed more thoroughfully in Section IV-D. The lowest semantic similarity value calculated was -1.0 for the pair $< switch, tv >$ and $< switch, hamper >$, followed by 0.00036 for the pair $< switch, jewelry\_box >$ and $< switch, microwave >$. While the highest similarity value was 0.75839 for the pair $< switch, cabinet >$ and $< switch, medicine >$, followed by 0.11285 for the pair $< switch, refrigerator >$ and $< switch, freezer >$.

C. Algorithms Comparison

The hypothesis we want to verify by applying the chosen algorithms are 1) UPGMA: is the distance of the semantic distribution of similarities relevant for predicting the sensor-activity association? 2) WPGMA: does considering the structural subdivision of the sensor objects positively influence such prediction? 3) VH: can we rely on the assumption that each activity is associated with a more central (i.e., critical) sensor object? The evaluation results particularly confirm the second and third of these hypothesis, because VH achieved the highest precision followed by WPGMA. Figure 1 shows the results we obtained by running UPGMA over the MITes dataset. The final clustering actual reflects the common knowledge, e.g., by grouping freezer and cold sink faucet together. However, too many sensors are too distant from any specific cluster. By applying WPGMA we got a better distribution of clusters, as shown in Figure 2. This result confirms that the sensors have structural relationships between each other that can be relevantly considered during the clustering. The results of applying Voor Hees (VH) are shown in Figure 3. The VH algorithm resulted in no sensor being distant from any specific cluster. Unsurprisingly then, this approach achieved the highest precision. When comparing our results with the annotated dataset, since we do not perform cluster labelling, it was not possible to directly map our clusters to the labels in Table IV-A. However, we considered the match verified whenever the sensors belonging to the same cluster according to our system (i.e., *predicted* class) were the ones that sensed the same activity in the MITes annotations (i.e., *actual* class). Consequently, we considered a 2-class classification problem, i.e., whether the sensors actually part of the same activity had been clustered in the same cluster. As a result a separate confusion matrix is created for each of the annotated activity. With such settings, we calculated precision and overall accuracy.

$$Precision = \frac{TP_{11}}{TP_{11} + FP_{12}}$$

$$Accuracy = \frac{TP_{11} + TN_{21}}{TP_{11} + TN_{22} + FP_{12} + FN_{12}}$$

Figure 4 shows the precision percentage achieved by our system on the given dataset, by using each of the hierarchical clustering algorithms. VH achieves an average precision of 89.5% followed by WPGMA which achieves 85.6% and UPGMA with 75.2%. Precision and overall accuracy were calculated and our system managed to predict which sensors were going to provide information relevant for each of the 27 annotated activities with an average accuracy of 69%. Details of the accuracy achieved by each algorith for some of the activities are in Figure 5. We believe our results to be relevant especially when compared with **a.** Wyatt et al. [18] which we consider being the most similar previous research effort, since we both model activities in terms of sequences of features of interest **b.** Kwon et al. [8] who achieved the state of the art in terms of precision with unsupervised hierarchical agglomerative clustering for sensor-based activity recognition. The experiments that we run is compared in Table IV-C with those run by Wyatt et al. and Kwon et al. Although our goals differ between Activity Recognition (AR), Activity Inference (AI) and Relevant Sensor Prediction (RSP), if each cluster is considered an activity we can then compare our results. As precision and accuracy we considered the best values among the distinct attempts made using algorithms such as Unsupervised Hierarchical Agglomerative Clustering (HIER), Hidden Markov Models (HMM) and Unsupervised Hierarchical (UH) Algorithms. Our results are relevant as we can notice that our system improved the accuracy by 32% and the precision by 5% with respect to such previous efforts from the state of the art.

D. Performance

The evaluated system run on a laptop equipped with Intel Core^{TM}2 Duo and 305 GB of disk space. We used the LD4S and EasyEsa service instances running on external servers in order to support and test a modular and distributed architecture. During the pre-processing

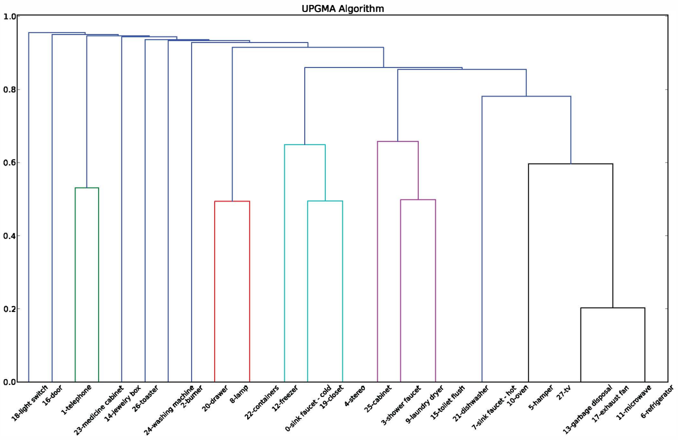

Fig. 1. Clustering performed by the UPGMA algorithm.

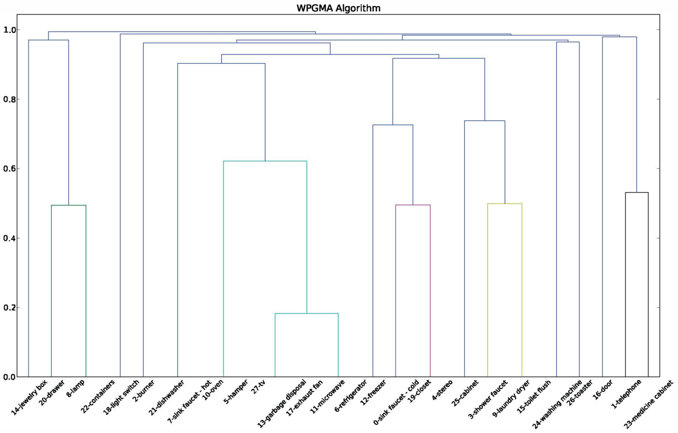

Fig. 2. Clustering performed by the WPGMA algorithm.

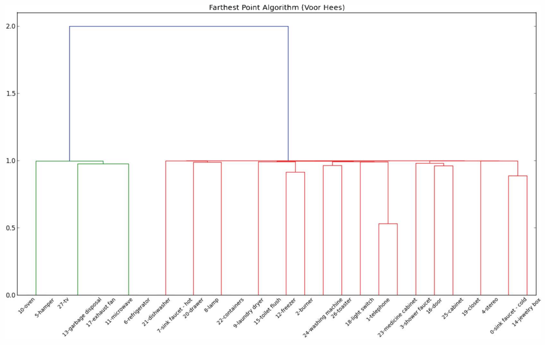

Fig. 3. Clustering performed by the Voor Hees algorithm.

TABLE II. Comparison between the experiment setup and results for our own approach and the previous closest research efforts.

	Kw●n et al.	Wyatt et al.	●urs
# Sens●rs	3	1●●	2●●
# Activities	5	26	16
C●llecti●n Time	5● mins	36● mins	2 weeks
G●al	AR	AI	RSP
Alg●rithms	HIER	HMM	UH
Precisi●n	79%	7●%	89%
Accuracy	-	52%	69%

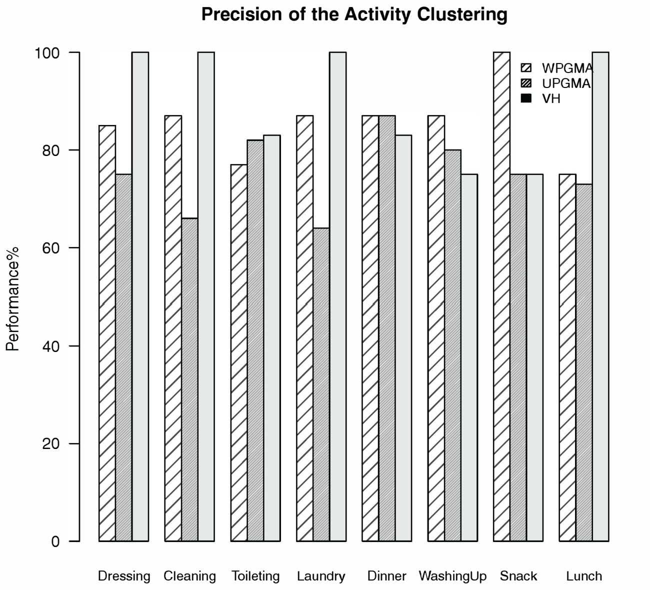

Fig. 4. Comparison between precision percentages achieved by the clustering algorithms for some of the activities.

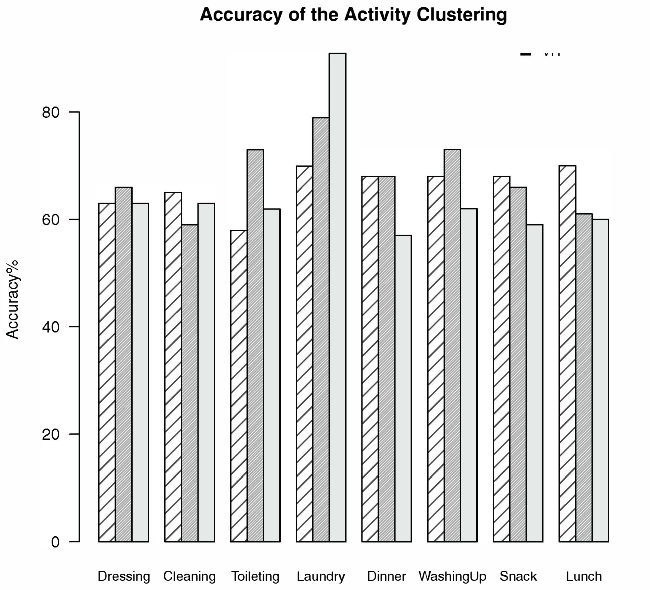

Fig. 5. Comparison between accuracy percentages achieved by the clustering algorithms for some of the activities.

step, the HTTP PUT requests forwarded to LD4S to both create the annotation as in Section II-B and store it in the LD4S triple store had an average execution time of 3 milliseconds. The overall system execution (excluding the MITes dataset pre-processing step) time was of 18 milliseconds. Forwarding a query to DataHub to retrieve all the available sensor datasets had an execution time of 3 milliseconds. This resulted in a list of 20 datasets out of which 3 were both featured with an open license and exposing a SPARQL endpoint. Among them, only LD4S was actually accessible. The average response time for the SPARQL queries we run (Listring II-B) on LD4S is equal to 246 milliseconds. Our system took 14 milliseconds to calculate the semantic relatedness of 351 pairs of sensors, during which the HTTP requests to the Easy-Esa API achieved an average response time of 9 milliseconds. In Figure 6 we analyse the growth of time cost for the similarity calculation with respect to the amount of sensor types. The highest time cost is 1 minute and 26 seconds for comparing 216 distinct sensor types, thus confirming our scaling expectation. As the amount of sensors that have already sensed a change in status for the current activity grows, the amount of sensors to be considered decreases. Finally, during the clustering step, both UPGMA and VH had a running time of 2 milliseconds while WPGMA took 12 milliseconds. The performance values achieved, confirm the possibility of updating the clustering with new sensors similarities at run-time. In fact, despite the devices being resource constrained, a clustering for most of the possible features of interest could be pre-computed offline.

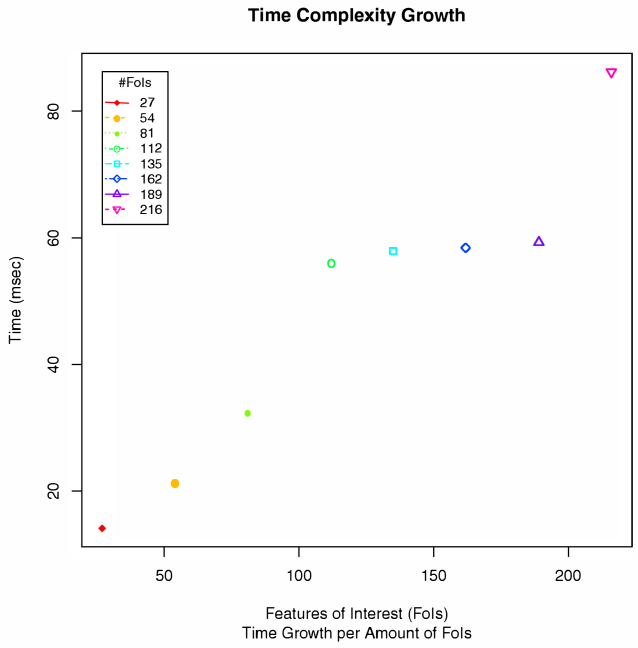

Fig. 6. Time complexity growth for the semantic relatedness claculation as the amount of FoIs increases.

V. Conclusions

We automate the activity-wise prediction of sensors relevancy with outstanding precision. To support the uptake and reproducibility of our methodology we use publicly available services and datasets. In the future we will reason over the clustering centroids to label the activities and improve the similarity calculation by running ESA on a domain-specific corpus.

Acknowledgment

This work is partially funded by the European Union (Grant No. 608682 - Vital), by Science Foundation Ireland (SFI/12/RC/2289) and by RSF1184 Insight Core Platform SW1.

References

[1] David Beckett and Tim Berners-Lee. Turtle - terse rdf triple language, 2011.

[2] T. Berners-Lee. Linked Data. Technical report, 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[3] D. Bimschas, H. Hasemann, M. Hauswirth, M. Karnstedt, O. Kleine, A. Kroeller, M. Leggieri, R. Mietz, M. Pagel, A. Passant, D. Pfisterer, K. Roemer, and C. Truong. SPITFIRE: Toward a Semantic Web of Things. *IEEE CommMag*, 2011.

[4] A. Broering, A. Remke, and D. Lasnia. SenseBox âĂŞ A Generic Sensor Platform for the Web of Things. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, 2011.

[5] D. Carvalho, C. CallÃś, A. Freitas, and E. Curry. Easyesa: A low-effort infrastructure for explicit semantic analysis. In *ISWC Posters and Demos*, 2014.

[6] A. Freitas, E. Curry, J. G. Oliveira, and S. O'Ryain. A distributional structured semantic space for querying RDF data. *Semantic Computing*, 2011.

[7] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Intl. Joint Conf. on Artificial Intelligence*, 2007.

[8] Y. Kwon, K. Kang, and C. Bae. Unsupervised learning for human activity recognition using smartphone sensors. *Expert Systems with Applications*, 2014.

[9] M. Leggieri, A. Passant, and M. Hauswirth. A contextualised cognitive perspective for linked sensor data - short paper. In *ISWC Semantic Sensor Networks Workshop*, 2010.

[10] M. Leggieri, A. Passant, and M. Hauswirth. inContext-Sensing: LOD augmented sensor data. In *ISWC Posters and Demos*, 2011.

[11] M. Leggieri, M. Serrano, and M. Hauswirth. Data modeling for cloud-based internet-of-things systems. In *IEEE iThings*, 2012.

[12] K. R. Page, D. C. De Roure, K. Martinez, J. D. Sadler, and O. Y. Kit. Linked Sensor Data: RESTfully serving RDF and GML. In *ISWC Semantic Sensor Networks workshop*, 2009.

[13] R. L. Stratonovich. Conditional Markov Processes. *Theory of Probability and its Applications*, 1960.

[14] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé. *Vision and Challenges for Realising the Internet of Things.* EU Commission, 2010.

[15] E. Munguia Tapia, S. S. Intille, and K. Larson. Activity recognition in the home setting using simple and ubiquitous sensors. In *PERVASIVE*, 2004.

[16] P. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Artificial Intelligence Research*, 2010.

[17] Wikipedia. English wikipedia dump, 2013.

[18] D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. In *AAAI*, 2005.