Improve Contextual IoT service discovery with semantic models

Mário Antunes, José Quevedo, Diogo Gomes and Rui L. Aguiar
Instituto de Telecomunicações
University of Aveiro, Aveiro, Portugal 3810–193
Email: {mario.antunes,quevedo,dgomes,ruilaa}@av.it.pt

***Abstract*—The Internet of things (IoT) is an ecosystem of smart connected devices that exchange data over a communication network. By integrating these devices into different vertical applications, the IoT has the potential to have a major impact on both the economy and society. However, the plethora of heterogeneous devices with varying ways of describing the information raise interoperability issues. In this context, the development of appropriate service discovery mechanisms enriched with semantic capabilities for understanding and processing context information is a key feature for turning raw data into useful knowledge and ensuring interoperability among different devices and applications. In previous work, we focused on surpassing the IoT semantics barriers while exploring novel networking approaches. To this end, we proposed a service discovery mechanism, realised on top of Named Data Networking (NDN), that relied on a semantic matching mechanism for achieving a flexible discovery process. Since the initial work, several improvements were made to the semantic similarity model at the basis of the semantic matching algorithm. This work replicates the scenario proposed on the former contribution and assesses the impact of the improved semantic model. Results show that while the previous semantic model achieves a mean Average Precision of *0.29*, the best performing current solution achieves *0.68*.**

***Index Terms*—Internet of Things, Information-Centric Networking, Service discovery, Semantic similarity**

I. Introduction

Internet of Things (IoT) has had extensive attention from industry and academia over the last few years. The connectivity of every piece of technology in the environment with the Internet has opened many avenues for innovation. From applications to devices, all elements have evolved to accommodate the decentralized asynchronous demands of IoT scenarios. Moreover, the adoption of IoT by the industry and its integration into more complex scenarios has led to the Industrial IoT (IIoT) which is at the basis of the next industrial revolution Industry 4.0 [1].

In parallel, a key area that also emerged as an ideal fit for IoT communications is Information-Centric Networking (ICN) [2]. ICN is an emerging future Internet architecture focused on content delivery. Unlike the current underlying architecture of the Internet, ICN intrinsically couples networking procedures with important supportive mechanisms, such as security, mobility support and efficient caching, thus reinforcing its suitability for IoT scenarios [3], [4]. Moreover, it has also found a way into mobile communications systems [5]–[7], and is expected to further impact IIoT scenarios.

In current IoT deployments, there is no uniform structure for data sharing, different devices/manufacturers specify their structure for sharing data leading to information silos [8]. This has hindered the interoperability between different applications and the realization of more complex IoT scenarios. Therefore, to make data useful and to ensure interoperability among different applications, it is necessary to provide data with adequate and standardized formats, models and semantic descriptions of their content (metadata), using well-defined languages and formats [9]. However, the lack of standards and the heterogeneity of formats for describing IoT content has triggered research on techniques to deal with unstructured information, where particular emphasis has been given to semantic similarity. The goal behind its application is to enable the adoption of the IoT on a wide scale by allowing the proper identification of information with a similar context, regardless of the vocabulary used therein [8].

In our previous work, we integrated and evaluated the unsupervised semantic similarity solution proposed in [10] with an ICN-based discovery mechanism developed on top of the Named Data Networking (NDN) architecture [11]. In doing so, some core concepts of [10] had to be further evolved and a novel service-query matchmaking interface was developed. The main focus of the previous work was on the foundations of the discovery mechanism. In this paper, the focus is shifted to the evolution and impact of the core semantic similarity model. As such, the semantic model was significantly improved, including aspects such as: (i) using unsupervised learning to identify categories within the word profile [12], [13], (ii) considering the impact of different clustering algorithms [14], and (iii) the usage of latent features to predict the missing frequencies in a word profile [8].

The remaining document is organized as follows. In Section II describes our previous solution for service discovery in NDN networks. Section III discusses the limitation of the previous solution, the motivation for this new work, and describes the solution. After this, we discuss the evaluation of the improved service discovery in Section IV. Section V, briefly summarizes the related work for semantic similarity models. Finally, the main conclusions are discussed in Section VI.

II. Discovery Solution Overview

In the current section, we present the fundamentals of the solution proposed in [15] to provide the reader with the necessary information to understand the developments in semantic similarity which are the core contributions of this paper. For further details, readers are encouraged to read the original contribution.

The discovery solution comprised four different entities: i) Clients, ii) Service Providers, iii) Discovery Brokers and iv) Semantic Matching Engines (SME), The different entities interact with each other, as depicted in Figure 1, and their principal functions may be described as follows:

1) *Client:* An entity interested in a certain information (e.g., actuators, end-user terminals). It supports two operations: *(i)* Service Discovery: The client issues a request to the Discovery Broker to find out the available services which are providing content suitable to its needs; *(ii)* Content Retrieval: The client issues a content request to a given Service Provider, which in turn provides it with the desired piece of content.
2) *Service Provider:* An entity providing one or more services (e.g., sensors, actuators). It communicates, using the NDN protocol, The Service Providers, support two operations: *(i)* Service (Un)Registering: Sends a request to the Discovery Broker to add/remove its services to/from the list of services it announces to potential clients; *(ii)* Content Providing: Listens/Satisfies interests from potential clients and provides them with the corresponding content.
3) *Discovery Broker:* The entity responsible for holding the information about the available services and for matching incoming queries against the available services (by interacting with the Semantic Matching Engine).
4) *Semantic Matching Engine:* The entity responsible for performing the actual matching of queries and services. It keeps track of the registered services and matches the incoming queries with the available services.

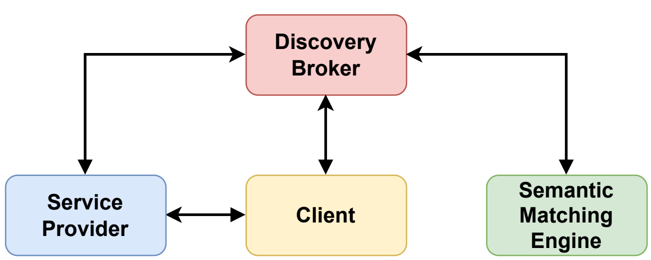

Fig. 1. Solution overview

III. Evolved Semantic Matching Engine

This section discusses the foundations of the Semantic Matching Engine and the limitations of the original version. Afterwards, the evolution of the engine is presented, mainly by providing the multiple improvements done to the semantic model over time.

A. Semantic Matching Engine: Foundations and Limitations

The Semantic Matching Engine depicted in Figure 2, calculates the similarity between a request and the multiple services registered within its local database.

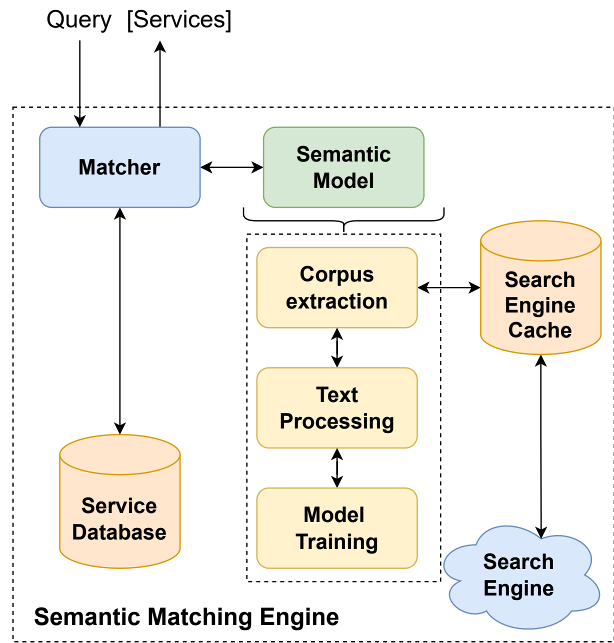

Fig. 2. Semantic Matching Engine

Two different metrics were considered: Jaccard Index (Equation 1) and Cosine similarity (Equation 2). Both metrics are widely known and used for the task of computing the similarity between sets.

$$jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

$$cosine(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \tag{2}$$

Before computing the similarity between services and queries (with the previously mentioned metrics), we have to compute the similarity pairs between each service tag and each query term. That is done using the previously mentioned semantic model, if the semantic similarity between a tag and a term is greater than a predefined threshold, we mark the pair as similar.

Besides the semantic-based similarity mechanism, it also provides matching similarity based on exact string matching (i.e., returns 1 or 0 depending on whether the words are the same or not) and matching within a certain Levenshtein distance (i.e. a given number of single-character edits).

The semantic model was developed using the original Distributional Profiles (DP) semantic model [10]. That model relies on web search engines to extract the distributional profiles of words (i.e., the weighted neighbourhood of the word). At the time, two different search engines were used to gather the corpus for the learning task, namely Bing[1] and Seek Storm(previously known as Faroo)[2]. Unfortunately, these

[1] www.bing.com
[2] seekstorm.com

search engines dropped their free Application Programming Interface (API)s.

The original corpus was lost and given the restrictions applied to the previously mentioned search engines, the later iterations of the semantic model have relied on an entirely different search engine named USearch[3].

The original DP was defined as

$$DPW(u) = \{w_1, f(u, w_1); ...; w_n, f(u, w_n)\} \qquad (3)$$

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). The profile can also be interpreted as a vector that represents a point in high dimensional space, where each word w_i represent a dimension and $f(u, w_i)$ represents its value in that dimension. To measure the similarity between two profiles we use the Cosine Similarity (Equation 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.

In our previous work, the overall performance of the semantic model was lacklustre, as can be seen in Figure 3. In the figure, we can see that the average precision of the semantic model is rather small when compared with conventional string matching and Levenshtein distance. The main advantage was that for the last two query types considered (full details in Section IV) the semantic model was able to deal with three and four synonyms better than the previous models (that achieve an average precision of 0).

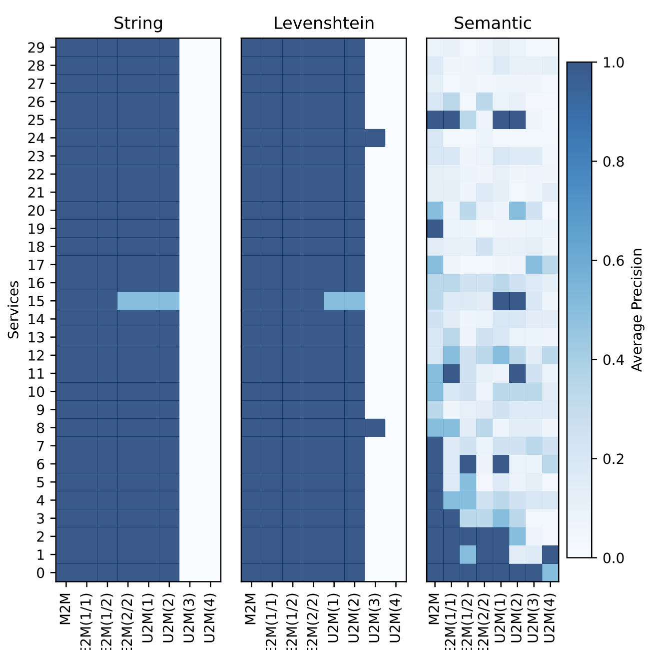

Fig. 3. Average Precision Heatmap

[3] https://usearch.com/

The main issue was that the semantic model learns the profiles using the snippets from a search engine. This data source was selected since it allows the corpus acquisition on the fly and selects the most relevant documents based on their internal algorithm. However, it also produces quite a noisy corpus, that has been the focus of the multiple evolutions of the semantic model. Thus, the current paper builds upon the previous contribution by providing functional enhancements to improve the overall performance.

B. Enhanced functionalities for better matching performance

This section details the evolution of the semantic model, and the methods used to deal with the corpus gathered from a search engine Although public web services offer some important advantages, they also have some disadvantages. DP 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. We proposed two filters can be applied to reduce the profile's unwanted dimensions The first one uses stemming to simplify each word to its stem (minimizing issues with plural words, and some written mistakes). The second filter uses employs the Pareto Principle to discard unwanted dimensions, based on their frequency. Other methods can be employed such as a p-value statistical significance test or knee/elbow point estimation. At this point, we designed the model as distributional profile of words (DPW).

Additionally, a profile can contain several senses of the target word (sense-conflation). Multiple word senses in a single profile may also change the word vector direction, limiting the potential of this method.

To minimize this issue, we proposed a novel model named distributional profile of multiple words categories ($DPWC$). This model uses clustering on the distributional profile to identify word senses. The rationale is that dimensions belonging to the same category are closer to each other than words from other categories.

These clusters do not represent word senses from a thesaurus. This means that there is not a one-to-one relationship between the clusters and a word in a thesaurus. Conceptually the clusters are more similar to categories in Latent Semantic Analysis (LSA) 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 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.

To minimize this issue, our model incorporates an affinity value between the target word and each category, which can be understood as a bias, and it measures the natural tendency of a word to be used as a specific category. The affinity is computed as the strength between the target word and the category. After computing all the affinity values, they are normalized with the following expression:

$$a_i' = \frac{a_i}{\sum a_i} \quad (4)$$

The profile is defined as follows:

$$DPWC(u) = \begin{Bmatrix} a_1; \{w_i, f(u_1, w_i); ...\} \\ ... \\ a_n; \{w_j, f(u_c, w_j); ...\} \end{Bmatrix} \quad (5)$$

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

$$S(u, v) = max(cosine(u_c, v_c) \times (a_{u_c} + a_{u_v} \Big/ 2)) \quad (6)$$

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 possible categories weighted by the average category's affinity. By incorporating affinities, the model minimizes the effect of low relevance categories.

To cluster the DP we create a square matrix that contains the frequencies of all the words within the profile. In short, each row represents a DP for the hyper-space defined by the target word profile. However, we end up with a sparse matrix. The only row (and column) that is guaranteed to be dense, is the row that contains the target word u. The remaining rows tend to be sparse, there are no guarantees that w_i a w_j (two dimensions from the target word u) have appeared together in our constrained corpus. Our intuition states that the 0 coefficients in the matrix are due to a lack of data, and do not capture the real distribution of the co-occurrence.

Following a similar approach as word2vec and LSA, we use a matrix factorization to reduce the latent dimensions and reconstruct a co-occurrence matrix where the 0 coefficients are replaced with predictions of the actual value. The co-occurrence matrix becomes dense and provides more information for the clustering algorithm, similarly to the factorization used in recommendation systems [16].

The factorization and reconstruction are also helpful to optimize the profile of the target word u. Due to the dimension reduction, the reconstruction "corrects" all values within the matrix (including the profile of the target word u).

IV. Evaluation

To evaluate the impact of the evolution of the semantic models, we create one simulation that implemented the Proof of Concept developed in [15]. In the previous work, we developed a representative dataset. We extracted the most common terms from well known IoT Platform Providers (e.g., libelium[4], carriots[5]).

[4]http://www.libelium.com

[5]https://www.carriots.com

TABLE I
Groups of Query

Group	Description	Sample Terms
M2M	Exact Match	moisture, greenhouse, soil, agriculture
E2M(1/1)	One word with one error	moisture**s**, greenhouse, soil, agriculture
E2M(1/2)	One word with two errors	moistur**is**, greenhouse, soil, agriculture
E2M(2/2)	Two words with one error each	moisture**s**, greenhouse**s**, soil, agriculture
U2M(1)	One word replacement	**wetness**, greenhouse, soil, agriculture
U2M(2)	Two words replacement	**wetness**, **hothouse**, soil, agriculture
U2M(3)	Three words replacement	**wetness**, **hothouse**, **ground**, agriculture
U2M(4)	Four words replacement	**wetness**, **hothouse**, **ground**, **cultivation**

The dataset is composed of services and queries each of which is described by 4 keywords. In the case of the queries, we considered 3 different approaches: *(i)* Machine-to-Machine (M2M) scenarios – the requester knows the exact keywords that better represent the service, *(ii)* Engineer-to-Machine (E2M) – the requester knows the proper keywords but is subjected to typing mistakes, *(iii)* User-to-Machine (U2M) – the requester has some knowledge about the service but does not know the exact keywords so it would most likely use synonyms of proper keywords. Following these approaches, and varying the number of errors/synonyms included in the query, we defined 8 groups of queries as described in Table I. The resulting dataset is composed of 30 services and 240 queries. Each service has 8 queries associated, each of which falls into one of the mentioned groups. We should mention that for each service as a different set of tags(represent the vocabulary) since we do not use a curated corpus (it is gathered automatically from the internet) the vocabulary will dictate the quality of the model.

We evaluated the performance of the different string matching algorithms (i.e., exact string matching, Levenshtein distance of 2 and semantic similarity) over the whole evaluation dataset, using two different statistics for comparing the similarity of the set of words (i.e., Jaccard Index and Cosine similarity). However, for all the cases the results obtained for Jaccard and Cosine were almost identical and therefore for the remaining of this subsection, we will be presenting only the results obtained for the Cosine similarity.

The reason for this is that the semantic engine uses hard votes to compute the similarity between sets. This means that the vectors are composed of either zero or ones, without considering any value in between. This is required for the Jaccard Index, as the metric computes similarity based on intersection and union of sets. But is not required for the Cosine similarity. This is a major drawback of the semantic engine, but the focus of this work is to evaluate the impact of an improved semantic model. The semantic engine, with the exception of the semantic model, was kept as faithful as possible in this implementation.

The original code was written in Java, but the current version was ported to Python since the newest semantic model is being developed in Python. The code is publicly available here[6]. As previously stated, the original corpus was lost. This

[6]https://github.com/mariolpantunes/semantic-matcher/releases/tag/0.1

limits the comparison between the previous semantic engine and the current one. To overcome this limitation, and allow future comparisons with the current version of the semantic engine, we uploaded the dataset[7] and corpus[8] to Kaggle.

Figure 4 represents the average precision of the answers provided by each of the string matching algorithms. In the figure, the small squares represent a query (e.g., the query within the group "M2M" that is associated with service "0") while its colour tone indicates the obtained average precision. In calculating the average precision we used Equation (7), where k is the rank in the sequence of retrieved documents, n is the number of retrieved documents, $P(k)$ is the precision (i.e., the fraction of the retrieved relevant documents) at cut-off k in the list and $rel(k)$ is an indicator function equal to 1 if the item at rank k is a relevant document and zero otherwise. For our evaluations, we considered relevant only the service associated with the query.

$$AP = \frac{\sum_{i=1}^{n}(P(k) \times rel(k))}{number\ of\ relevant\ documents} \tag{7}$$

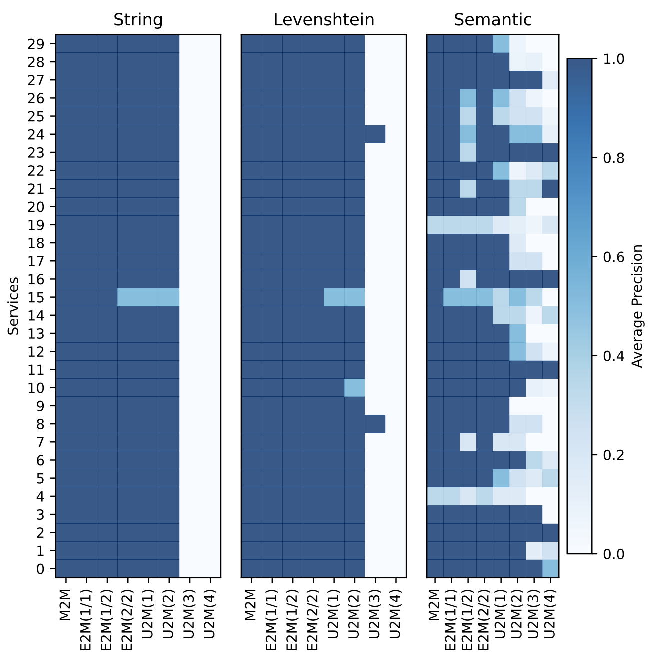

Fig. 4. Average Precision Heatmap

From Figure 4 it can be observed that exact string matching and Levenshtein distance present a great precision for the first groups, but queries with more than two synonyms are not properly matched to the relevant service. However, the semantic similarity matching still manages to get the matching service, although not in the proper rank.

An analysis of the results from Figures 4 and 3 show that the current approach constitutes the first step into further refinements of the semantic matching algorithm. However, they demonstrate the feasibility of using such techniques. Particularly for the case of the queries that include 3 and 4 synonyms, where the conventional methods did not obtain a match for the service, but the semantic method was able to find some matches.

The recent versions of the semantic model provide a significant improvement over the latter version. In the previous work, the semantic model was only able to achieve a mean Average Precision of 0.29 while the newest method was able to achieve 0.68 for DPW and 0.65 for $DPWC$. The versions that use latent features (identified in the table with the capital L) do not show significant improvements. The full results can be found in Table II

TABLE II
FULL RESULTS OF THE EVALUATION.

Query	String	Levenshtein	DPW	DPW (L)	DPWC	DPWC (L)
M2M	1.00	1.00	0.96	0.37	0.92	0.21
E2M(1/1)	1.00	1.00	0.94	0.38	0.90	0.19
E2M(1/2)	1.00	1.00	0.78	0.17	0.76	0.14
E2M(2/2)	0.98	1.00	0.94	0.37	0.90	0.20
U2M(1)	0.98	0.98	0.78	0.30	0.70	0.19
U2M(2)	0.98	0.96	0.50	0.26	0.50	0.17
U2M(3)	0.00	0.06	0.34	0.30	0.35	0.18
U2M(4)	0.00	0.00	0.25	0.25	0.20	0.18
mAP	0.74	0.75	0.68	0.30	0.65	0.18

Although these versions tend to achieve higher accuracy in the task of semantic similarity (see [8]), they also correct the weights in the profile using the matrix reconstruction, by rescaling the values in the profile. This operation does not impact the semantic similarity evaluation since the metric used is the Pearson Correlation, which is scale-invariant. However, as proven by our evaluation, that is an issue for the task of query matching.

Overall, the results of the recent semantic model are quite positive, becoming quite competitive with other similarity metrics.

V. 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 exploit information from a lexical resource.

Lexical-resource-based measures rely on manually annotated lexical resources, such as WordNet [17], 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 lexical semantic relationships 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 metrics have been proposed over WordNet [18].

[7]https://www.kaggle.com/datasets/mantunes/semantic-service-discovery-in-ndn

[8]https://www.kaggle.com/datasets/mantunes/corpus-for-semantic-matching

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[9].

Strict corpus-based measures rely on the hypothesis that words in similar contexts tend to be semantically close [19]. These methods do not require a lexical resource, but usually require a large corpus that represents the common usages of the target words. One of the most successful methods was the word2vec model [20], where a shallow neural network learns word associations from a large corpus of text. The model represents each distinct word with a vector. The vectors are learned in such a way that a simple mathematical function (*e.g.* the cosine similarity) indicates the level of semantic similarity between the words represented by those vectors. Other successful models, such as LSA [21] and Latent Dirichlet Allocation (LDA) [22], use dimensionality reduction to learn a compressed feature vector for each distinct work. Similarly to word2vec, these vectors can be used to estimate the semantic similarity between word pairs. Finally, the most successful models employ the recent advances in deep learning (such as BERT [23] and subsequent variants) to enhance the performance. Deep neural network models are built based on two fundamental operations: convolution and pooling. The convolution operation in text data may be defined as the sum of the element-wise product of a sentence vector and a weight matrix. Convolution operations are used for feature extraction. Pooling operations are used to eliminate features that have a negative impact, and only consider those feature values that have a considerable impact on the task at hand.

The lexical-resource-based methods exploit underlying ontologies to disambiguate synonyms, while corpus-based measures are more versatile. However, many authors have found ways to exploit the best of each method and build hybrid models to measure semantic similarity. Camacho Collados *et al*. [24] proposed an approach where the knowledge source BabelNet [25] is used to build a corpus based on which vector representations for concepts (words or groups of words) are formed. Initially, the Wikipedia pages associated with a given concept, in this case, the synset of BabelNet, and all the outgoing links from the given page are used to form a subcorpus for the specific concept. The sub-corpus is further expanded with the Wikipedia pages of the hypernyms and hyponyms of the concept in the BabelNet network. The entire Wikipedia is considered the reference corpus.

It is worth mentioning that the previously mentioned models provide very accurate methods to estimate semantic similarity. However, those solutions rely heavily on structured information or large and well-maintained corpora (as summarized in Table III). The ever-increasing number of constrained devices in highly dynamic environments (consider a Internet of Things (IoT) or edge computing scenarios) makes it very difficult to build and maintain semantic networks or clean and up-to-date corpus. That led us to propose a DP model that trades accuracy with flexibility and simplicity.

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

TABLE III
SUMMARY OF THE MAIN SEMANTIC MEASURES

Measure	Requirements	Feasibility
Lexical-resource	Manually annotated lexical resources	Low (lexical-resource difficult to maintain)
Corpus	Large Corpus with the necessary vocabulary	Possible (depends on vocabulary)
Hybrid	Combines the requirements from the previous solutions	Low (lexical-resource difficult to maintain)

VI. CONCLUSION

In this paper, we focused on improving the semantic similarity mechanisms used in our previous work for semantic-based service discovery in content-centric networks. The new evolved semantic matching engine clearly outperformed the former solution. Moreover, it clearly opened up new avenues for improvements which will be explored in future works targeting not only the evolution of the semantic models and engine from this work but also a deeper integration into the content-centric functionalities (e.g., by targeting its integration within the named-based interest forwarding process).

The main objective of this work was to evaluate the impact of the recent improvements in the semantic model. As seen in Section IV, the recent versions of the semantic model have a substantial impact on the accuracy of the semantic matching engine. Contrary to the previous work, the code, the corpus and the dataset are publicly available and can be used as a framework for evaluation of future improvements.

Several improvements were not implemented and should be properly researched. As previously stated, the Cosine Distance is not being used to the full extension, since the matching engine only used hard votes. We will consider a different matching algorithm that uses the strength of the similarity as the value for the voting. Another possibility would be evaluating other well known semantic models, such as glove [26] and fastText [27]. These models were not considered initially since they do not allow continuous training, and require a considerable large corpus for training. Nevertheless, it would be interesting to measure the impact of these models in the scenario explored in this work.

ACKNOWLEDGMENT

This work is supported by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Lisbon (POR LISBOA 2020) and the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020 framework [Project Augmanity with Nr. 046103 (POCI/LISBOA-01-0247-FEDER-

046103)]; and by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/50008/2020-UIDP/50008/2020.

References

[1] 5G-ACIA, "5g for industrial internet of things (iiot): Capabilities, features, and potential," Tech. Rep., 2021.

[2] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A survey of information-centric networking research," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 2, pp. 1024–1049, Second 2014.

[3] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: Challenges and opportunities," *IEEE Network Magazine*, 2015.

[4] B. Nour, K. Sharif, F. Li, S. Biswas, H. Moungla, M. Guizani, and Y. Wang, "A survey of internet of things communication using ICN: A use case perspective," *Computer Communications*, vol. 142-143, pp. 95–123, jun 2019.

[5] P. Suthar, M. Stolic, A. Jangam, D. Trossen, and R. Ravindran, "Experimental Scenarios of ICN Integration in 4G Mobile Networks," Internet Engineering Task Force, Internet-Draft draft-irtf-icnrg-icn-lte-4g-12, Mar. 2022, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-icn-lte-4g-12

[6] R. Ravindran, P. Suthar, D. Trossen, C. Wang, and G. White, "Enabling ICN in 3GPP's 5G NextGen Core Architecture," Internet Engineering Task Force, Internet-Draft draft-irtf-icnrg-5gc-icn-04, Jan. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-5gc-icn-04

[7] A. Rahman, D. Trossen, D. Kutscher, and R. Ravindran, "Deployment Considerations for Information-Centric Networking (ICN)," RFC 8763, Apr. 2020. [Online]. Available: https://www.rfc-editor.org/info/rfc8763

[8] M. Antunes, D. Gomes, and R. L. Aguiar, "Semantic similarity on constraints datasets: A latent approach," in *2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, aug 2021.

[9] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497 – 1516, 2012.

[10] M. Antunes, D. Gomes, and R. Aguiar, "Semantic features for context organization," in *Future Internet of Things and Cloud (FiCloud), 2015 International Conference on*, Aug 2015.

[11] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[12] M. Antunes, D. Gomes, and R. Aguiar, "Learning semantic features from web services," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, aug 2016.

[13] M. Antunes, D. Gomes, and R. L. Aguiar, "Towards IoT data classification through semantic features," *Future Generation Computer Systems*, vol. 86, pp. 792–798, sep 2018.

[14] ——, "The impact of clustering for learning semantic categories," in *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security*. SCITEPRESS - Science and Technology Publications, 2018.

[15] J. Quevedo, M. Antunes, D. Corujo, D. Gomes, and R. L. Aguiar, "On the application of contextual IoT service discovery in information centric networks," *Computer Communications*, vol. 89-90, pp. 117–127, sep 2016.

[16] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, aug 2009.

[17] G. A. Miller, "WordNet," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, nov 1995.

[18] G. Varelas, E. Voutsakis, P. Raftopoulou, E. G. Petrakis, and E. E. Milios, "Semantic similarity methods in wordNet and their application to information retrieval on the web," in *Proceedings of the seventh ACM international workshop on Web information and data management - WIDM '05*. ACM Press, 2005.

[19] S. Padó and M. Lapata, "Dependency-based construction of semantic space models," *Computational Linguistics*, vol. 33, no. 2, pp. 161–199, jun 2007.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[21] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological Review*, vol. 104, no. 2, pp. 211–240, 1997.

[22] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende, "Knowledge-enhanced document embeddings for text classification," *Knowledge-Based Systems*, vol. 163, pp. 955–971, jan 2019.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectionaltransformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019.

[24] J. Camacho-Collados, M. T. Pilehvar, and R. Navigli, "NASARI: a novel approach to a semantically-aware representation of items," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2015.

[25] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, dec 2012.

[26] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[27] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, dec 2017.