International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 6, December 2021, pp. 5066~5080
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp5066-5080

Mining knowledge graphs to map heterogeneous relations between the internet of things patterns

Vusi Sithole, Linda Marshall
Faculty of Engineering, Built Environment, and Information Technology, Department of Computer Science, University of Pretoria, Pretoria, South Africa

Article Info

Article history:

Received Nov 17, 2019
Revised Apr 13, 2021
Accepted May 11, 2021

Keywords:

Internet of things
Knowledge graphs
Patterns
Text processing
Topic modelling

ABSTRACT

Patterns for the internet of things (IoT) which represent proven solutions used to solve design problems in the IoT are numerous. Similar to object-oriented design patterns, these IoT patterns contain multiple mutual heterogeneous relationships. However, these pattern relationships are hidden and virtually unidentified in most documents. In this paper, we use machine learning techniques to automatically mine knowledge graphs to map these relationships between several IoT patterns. The end result is a semantic knowledge graph database which outlines patterns as vertices and their relations as edges. We have identified four main relationships between the IoT patterns-a pattern is similar to another pattern if it addresses the same use case problem, a large-scale pattern uses a small- scale pattern in a lower level layer, a large pattern is composed of multiple smaller scale patterns underneath it, and patterns complement and combine with each other to resolve a given use case problem. Our results show some promising prospects towards the use of machine learning techniques to generate an automated repository to organise the IoT patterns, which are usually extracted at various levels of abstraction and granularity.

Corresponding Author:

Vusi Sithole
Department of Computer Science
University of Pretoria
Lynnwood Road, Hatfield, Pretoria, 0002, South Africa
Email: u04409477@tuks.co.za

1. INTRODUCTION

At the moment, there is a lot of buzz about the internet of things (IoT) and its impact on our day-to-day lives and activities [1]. The IoT is expected to improve all spheres of our lives: from the way we live at our homes, the way we travel, the way we do shopping, and the ways retailers and manufacturers keep track of their inventory [2]. In the literature, various definitions for the IoT exists. While these definitions may vary, by way of vocabulary, they all describe the IoT as a paradigm in which multitudes of devices are connected to each other and to the internet [3]-[5]. In other words, the IoT represents a giant network of connected digital objects or things and people, although the later have a limited intervention [4], [6], [7]. Ultimately, these objects have the capabilities to collect and share data about their surrounding environment and their activities [8], [9]. In the IoT, this is possible since the devices and objects consist of built-in sensors which are in turn connected to the IoT platform, which integrates data from various objects and applies analytics to share valuable intelligence with the connected applications [3], [4], [8], [10].

Due to its make-up, the IoT consists of multiple patterns that are at the core of the various solutions used for its design architectures [1], [11]. These patterns include conventional solutions in the form of

technologies, patented electronic products, and accepted standards that govern the activities in the IoT [11]. These patterns are generally accepted by both experts and the IoT practitioners to be effective, reusable, and conventional solutions for addressing known IoT use cases. Inevitably, due to its size and magnitude the IoT consists of a large number of solutions that are used for the realisation of the IoT domain use cases. The astronomical number of patterns in the IoT makes it harder for IoT practitioners to find and select the right solutions for a given IoT use case problem in a timely manner.

To organise and store this large number of patterns for easy retrieval, we need an automated knowledge base which consist of many of these patterns gathered from a wide variety of sources. This will allow us to store these IoT patterns in a programmatic way using machine learning algorithms, and interlinking patterns that are, in some way, related. In this paper, our objective is to build such a prototype knowledge graph database which links the existing IoT patterns based on some pre-defined relationships. Using this connected dataset, the IoT practitioners can simply locate a desired pattern based on the context of their queries and intents. The vision is to build an intelligent classification scheme for all the IoT patterns, which requires minimal human intervention.

Knowledge graph databases provide an effective solution to support the task of storing related IoT patterns. At the moment, several models and architectures are being tested for their capabilities to mine knowledge graphs from text [12], [13]. This usually involves using a hybrid of natural language processing (NLP) techniques to extract important information from large corpora of text [14]. In this paper, we test a combination of proven methods with new novel techniques to fulfil this task.

The rest of this paper is organised as follows: Section 2 outlines our proposed approach for mining knowledge graphs to showcase relations between the IoT patterns. Section 3 is a brief description of the research methodology followed in conducting this study. In section 4, we present some findings as proof of concept and provide some results from the conducted experiments. Section 5 sums up the study and summarises the key takeaways.

2. THE PROPOSED PROCEDURE FOR MINING KNOWLEDGE GRAPHS TO MAP THE RELATIONS BETWEEN THE IOT PATTERNS

In this section we present a multifaceted approach for mining knowledge graphs to map heterogeneous relationships between the IoT patterns. In the literature, there are several methods used to organise the IoT patterns, including but not limited to: i) organising patterns by their scope and purpose [11]; ii) organising patterns by their semantics or properties [15]; iii) organising patterns by their design level scalability [8]; iv) organising patterns by their relationships [16]. The focus of this paper is on an approach for organising the IoT patterns by their heterogeneous relationships. Although the pieces that make up this approach are segmented, the holistic approach attested contributes largely to a converged prospectus of a unified model for mining knowledge graphs for the IoT patterns.

2.1. Entities extraction

In knowledge graph data extraction, the names of entities are generally accepted to be proper nouns (NNP) and are extracted using probabilistic models [17]. Some of the common techniques used here are dependency parsing [18], [19], part of speech tagging [20], [21], and named entity recognition [14], [22], [23]. However, such techniques are particularly useful in identifying entities from generic texts. In this study, we test topic modelling as a technique for extracting the entities (i.e., the IoT pattern names). This technique is briefly discussed below.

2.1.1. Extracting the IoT patterns as topics

In topic modelling, a topic is viewed as a probability distribution over a fixed vocabulary [24]-[26]. The basic idea behind topic modelling is that documents contain multiple topics, and thus the key idea is to discover a topic distribution over each given document as well as a word distribution over each topic [27]. This is represented by a $N \times K$ and a $K \times V$ matrix, respectively. In our approach, d denotes a document narrating a given IoT pattern, z is a possible topic (i.e. the pattern name), w denotes a word (which might characterise the pattern), and N_d is the total number of words in a given document. The probability of topic z in a document d is denoted by $P(z|d)$, and $P(w|d)$represents the probability of word w in topic z. In brief, the process followed can be summarised in this way: (a) first, we randomly choose a topic z from the distribution over topics, $P(z|d)$, and then (b) randomly choose a word w from the corresponding distribution over the vocabulary $P(w|d)$. In our case, both $P(z|d)$and $P(w|d)$are assumed to be multinomial distributions, $x_i \epsilon \{0, \dots, n\}$, expressed in the form,

$$P(x \mid \theta) = \frac{n!}{\prod_{i=1}^{d} x_i!} \prod_{i=1}^{d} \theta_i^{x_i}, n = \sum_{i=1}^{d} x_i, \sum_{i=1}^{d} \theta_i = 1, \theta_i \geq 0 \quad (1)$$

This means that the topic distributions in all the documents in the corpus share the common Dirichlet prior σ [27]. Similarly, the word distributions of topics have a common Dirichlet prior η. Given these parameters for document d, parameter θ_d of a multinomial distribution over K topics is expressed from Dirichlet distribution $Dir(\theta_d|\sigma)$. In a similar manner, for topic k, parameter $\beta_k$of a multinomial distribution over V words can be derived from Dirichlet distribution $Dir(\beta_d|\eta)$. Given all the parameters, the joint distribution of all the hidden and observed variables-joint distribution of topic mixtureθ, a set of K topics, word mixture β, as well as a set of Nwords $w$can be expressed as (2):

$$p(\beta, \theta, w, z \mid \sigma, \eta) = \prod_{i=1}^{d} p(\theta_d \mid \sigma) \prod_{i=1}^{N_d} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \prod_{k=1}^{K} p(\beta_k \mid \eta) \quad (2)$$

Via the joint distribution in (2), we can estimate the posterior distribution of unknown model parameters and hidden variables, $p(\beta, \theta, z|w)$. To obtain an estimate of a parameter, we use the Gibbs sampler for inference, which we find to be more accurate than its alternatives [24]. Our approach is to use semi-supervised guided LDA to seed certain topics by controlling the parameters such that the desired topics are given an extra boost to achieve higher probabilities. This approach is briefly discussed in appendix.

2.2. Attributes extraction

In natural language processing, several techniques for attributes extraction exist. Common approaches and architectures that are used include conditional random fields (CRFs) [28], convolutional neural network (CNNs) [29]-[31], and long short term memory networks (LSTMs) [14], [32], [33]. These are not necessarily exclusive for attributes extraction. In this study, we use a linguistics-based approach for attributes extraction that relies on text and document analysis. This involves the identification of the important sentences which represent the document. Text normalisation is applied before syntactic and semantic analysis of the text which include extracting the text from the original document (format conversion is needed to convert the file into XML), removing floating objects like images, figures, tables, and dividing the text into sentences. After text normalisation is completed, the normalised text is a passed into an attributes extraction system. In this study, our aim is to build an intuitive text-processing model for extracting the IoT pattern attributes from a document. We use a combination of features to extract those sentences that describe the core attributes of a particular pattern from the given document. The features selected are based on linguistics knowledge and document structure. Our attributes extraction technique is not only applicable to well-structured documents but can also be used to extract attributes from informal, unstructured sources including web pages such as Wikipedia. The attributes extraction features used include *header blocks*, *sentence length*, *sentence types*, *lexical units*, and *sentence position*. These features are summarised in Table 1.

Table 1. Features for attributes extraction

Features	Description
Header Blocks	Headings and sub-headings mark the beginning of key topics or themes in a document. In scientific literature, important sentences that describe the core essence of a subject- matter can be found under certain headings, e.g., Literature Review, and Discussions. Generic headings such as Introduction, Methodology, and Conclusion, usually do not contain sentences that are subject matter driven. Using a taxonomy of various headings collected from several scientific papers on the IoT patterns, we selected common 'subject-matter headings' across several articles and allocated higher scores to sentences under those headings.
Sentence Length	Nobata *et al*. [34] identifies sentence length as one of the key attributes to extract important sentences within a given document. In this study, we allocate higher scores to regular or longer sentences, and penalise short sentences by giving them lower scores. That is, for each given sentence in a document, we build a function that returns its length L_i. Similar to Nobata *et al.* [34] and Edmundson [35], we penalise sentences which are shorter than a L_b (i.e. the benchmark length) as (3). $\ell(S_i) = 0 \; if L_i \geq L_b$ $L_i - L_b(otherwise)$ (3) For all the sentences that satisfy the benchmark length, the function (3) returns a value of 0, and returns a minus (-) value for all the sentences which do not satisfy L_b.

Table 1. Features for attributes extraction (*Continue*)

Features	Description
Sentence Types	The type of sentence in a given document can also be used to determine its importance as it relates to the subject matter [36]. For example, in an interrogative document, sentences that terminate with a question mark (?) are generally considered more important. For our system, we allocate higher scores to declarative sentences (i.e., those sentences that terminate with a full point or full stop (.). We penalise all exclamatory sentences (i.e., those sentences that terminate with an exclamation mark (!) as well as interrogative sentences (i.e., that terminate with a question mark (?)).
Lexical Units	Lexical units represent the parts of speech from which a sentence is composed of. Examples of these parts of speech include nouns, verbs, and adjectives. The presence of these lexical units in a sentence can be used to determine its significance in a document. In this study, we allocated higher scores to sentences that contained the lexical units or the terms *'to'* and *'for'* and thereafter followed by a *linguistic unit* such as a *gerund* and an *infinitive*. Technically, if we let S_i be a sentence of length $L_i$which consist of $n$number of terms such that $S_i = W_0, \dots, W_n$in sequential positions in a given sequence, then we can prepend string $[S_i]$ and append string $[S_j]$ so that $W_i = [S_j]$ and $W_{i+1} = [S_j]$. In this way, important sentences will be those that satisfy the condition(s): (i) ($[S_j]$, *'to'* and$[S_j]$, *linguistic unit* and (ii) ($[S_i]$, *'for'* and $[S_j]$, *linguistic unit*).
Sentence Position	Sentence position as a locational attribute is used by making a hypothesis that sentences that are located in the middle of the document are more significant than those found at the beginning or end of the document. This implies penalising all the sentences positioned at the start and towards the end of the document using the function. $\rho(S_i)(1 \le i \le n) = 1\ if\ (i < N)\ 0(otherwise)$ (4) In function (4), the output for each given sentence is normalised to the scale [0, 1] in which N represents the specified threshold that returns the number 1 for all the sentences in the middle of the document.

2.2.1. Score normalisation

When multiple features are used for sentence extraction, a standard method must be developed to make the magnitude of the numbers comparable across the different features. Normalisation [37] is the term for this procedure. The method used in this article is to assign weights to individual features and then add the weighted scores to obtain an overall score. This method is called *simple additive weighting* [38]. The vectors are normalised in this case such that the number of the individual features is 1. That is, the vectors' distribution values are in the range [0, 1]. For a given sentence, the score of a single feature is given by (5).

$$F_i(x) = \sum_{i=1}^{n} \omega_i x_i \tag{5}$$

Subject to zero-one constraints of the form (6).

$$x_i \in \{0,1\},\ \forall_i \in \{1,\dots,n\} \tag{6}$$

In which x_i represents an allocated feature score, and ω_i is the weight for the feature. The allocated feature scores range between 0 and 1, and are divided into three distributional classes: [0.00, $\geq$0.33], [>0.33, $\leq$0.67] and [>0.67, $\leq$1.00]. High ranked features such as declarative sentences are allocated higher scores in the range of [0.67, 1.00] while penalised sentences such as those found at the top section of the document in terms of their position are given scores in the range of [0.00, 0.33], etc. The value allocation for ω_i depends on whether or not the document contains header blocks. We use a weighting choice of the form ω_α if the text corpus includes header blocks. This choice implies that weights ω are distributed uniformly across all five functions or features discussed above, resulting in ω_α=0.2 for each unit feature. Nonetheless, if no text block in a document is recognised as a section header, the form's weight choice ω_β is applied to the four features, giving each unit feature a value of ω_β=0.25. After each sentence's individual feature vectors have been established, the sentence's overall cumulative score can be determined. The aggregate score $\delta(S_i)$ is calculated using the following unit scoring functions and weights:

$$\delta(S_i) = \frac{\sum_{i=1}^{n} \omega_i x_i}{\sum_{i=1}^{n} \omega_i} \tag{7}$$

2.2.2. Parameter optimisation

In order to fit the model parameters or weights, we treat our attributes extraction system as a continuous-state optimisation problem. In other words, we want to find the optimisation weights for each of the attribute's extraction features. These are weights minimising the error in the model and can be computed using the mean-squared error of the form.

$$J_n = \frac{1}{n}\sum_{i=1}(y_i - f(x))^2 \quad (8)$$

To calculate the error, we descend to the minimum of the function using the gradient information, with a typical learning rate, $\alpha(i) \approx \frac{1}{i}$. The regression algorithm for this process is shown in appendix.

2.2.3. Words segregation

The final activity in attributes extraction is to segregate words in the extracted sentences. This is done to eliminate common English words such as *'the'*, *'is'*, *'of'*, *'for'*, and so on. In other words, we only consider words that are considered a major part of speech (MPoS) as descriptors for each pattern. These include *Verbs*, *Adverbs*, *Nouns*, and *Adjectives* [39]. This step is necessary given that knowledge graphs are better modelled and expressed using single word attributes [40].

2.3. Relations extraction

Our approach to relationship extraction is mainly semi-supervised-we label relationships between patterns based on a set of pre-identified relations extracted from multiple sources. This approach is feasible given that there is only a small set of relations that can be found between patterns. Table 2 summarises the relationships that can be found between the IoT patterns.

Table 2. The IoT pattern relationships

Relationship	Description
Similar	A pattern is similar to another pattern.
Uses	A pattern uses another pattern.
Compositional	A pattern is composed of other patterns.
Complements	A pattern combines with another pattern to solve a problem.

To cater for the variety in language, we use a form of semi-supervised training using relations embeddings. In this case, similar embeddings for a given relation denote paraphrasing or a synonymous expression for that relation. For each relation, we use a thesaurus to extract synonymous expressions that may be used by other authors to document the same relation between patterns. These synonyms are then embedded together with the progenitor relation. The intuition behind this approach is that the use of multiple synonymous words is 'more natural' than using a single term. This is considering that authors generally use free text and will express the same relation using different vocabulary in different documents. This increases the probability of establishing the correct relationship between patterns. The procedure to learn similar relations embeddings is straightforward and is briefly explained below.

2.3.1. Synonyms-enhanced relations embeddings

For relations embeddings of the IoT patterns, we adopt a Word2Vec model in the form of continuous-bag-of-words (CBOW) [41]. This model aims at predicting the target word, given context words in a sliding window. For each progenitor relation, we use its synonyms as context words. In this way, each synonymous word is also embedded in close proximity to the rest of the text in the document. This sort of fine-tuning increases accuracy and precision in the results. Formally, given a sequence of words $D = \{x_i, \dots, x_M\}$, the objective of our relations embedding model is to maximise the average log probability.

$$\ell(D) = \sum_{i=K}^{M-K} \log \Pr(x_i \,|x_{i-K}, \dots, x_{i+K}) \quad (9)$$

In this instance, K is the context window size of the target word, i.e., progenitor relation. Our relations embedding model formulates the probability $\Pr(x_i \,|x_{i-K}, \dots, x_{i+K})$ via a Softmax function as (10).

$$\Pr(x_i \,|x_{i-K}, \dots, x_{i+K}) = \frac{\exp(x_o^T.x_i)}{\sum_{x_i \epsilon W} \exp(x_o^T.x_i)} \quad (10)$$

In (10), W denotes the vocabulary, x_i is the vector representation of the target word or the progenitor relation, and x_o is the average of all the context word vectors represented by (11).

$$x_o = \frac{1}{2K}\sum_{j=1-K,\ldots,I+K,j\neq i} x_j \quad (11)$$

Similar to a CBOW model, our relations embedding model is optimised by making use of hierarchical SoftMax and negative sampling [42].

2.4. Knowledge graph construction and probabilistic models

The final step in our proposed approach to mining knowledge graphs is cleaning the extraction graphs, incorporating ontological constraints, and discovering statistical relationships within the extracted data. This step is necessitated by the errors that are inevitable in the extracted knowledge. The solution to this problem is using high-dimensional probabilistic models to find the most likely knowledge graph by sampling and optimisation [43]. Using this approach, each candidate fact in the extraction graph is treated as a variable. We use one main factor to determine the knowledge graph validity, namely: ontological knowledge about the domain. The ontological knowledge is used to parameterise the dependencies between the existing variables [44], [45]. The ontology framework serves to allow us to discover whether the extraction graph in the structure has any inconsistencies (also known as validation) and to logically extract implicit information from data (known as inference) [46]. For instance, an example of validation is an error in the data whereby an entity does not possess any properties. An example of inference according to the framework above is the derivation of a conclusion that if an object has properties and relations with other objects, then it is an entity. In this paper, our knowledge graph database treats each fact as a Boolean which infers a truth value for each fact via optimisation. To establish the validity of our approach, we insert the extracted knowledge using a simple script in which possible facts are posed as queries. The semantic rules are determined by how close the existing properties in the database match the inserted properties of the possible patterns. In other words, the ontological and semantic rules are grounded by substituting literals into formulas. Simply put, each ground rule consists of a weighted satisfaction metric derived from the formula's truth value which assigns a joint probability for each possible knowledge graph as in (12).

$$P(G|E) = \frac{1}{Z} exp\left[\sum_{r\epsilon R} w_r \Phi_r(G|E)\right] \quad (12)$$

Together, the ontological rules and semantic rules mined from data determine the level to which the knowledge graph is correct. That is, the level to which the inserted extracted knowledge matches and correspond to the already existing knowledge in the knowledge graph database.

3. RESEARCH METHOD

The research study employs a pragmatic research philosophy, relying primarily on text analysis and aspects of corpus linguistics, with the latter being used in particular for data normalization. The process of creating the artefact mentioned in this paper was driven by design science studies. Below is a detailed description of the data preparation and processing system.

Text normalisation is the first step in the process, which is needed for data preparation. This is a data cleansing technique that requires a pre-processing procedure that transforms scholarly documents into sentence-level sequences of text blocks. Our IoT patterns were often in the form of HTML files or scholarly PDF documents. Both the HTML web pages and the scholarly PDF files were mostly not standardised for ready processing as they did not contain block level elements or unique IDs.

Furthermore, the bulk of the papers had inconsistencies in their formatting. Other papers, for example, included photographs, and some used italics or bold text to emphasize key points and themes. To overcome this obstacle, the records were manually pre-processed. Converting each HTML and PDF document to an XML format that recognizes all line breaks was part of this process. Pre-processing also included eliminating images and assigning a specific ID or vector representation to each sentence. All information pertaining to the additional pattern(s) was manually extracted from documents presenting more than a single IoT pattern in order to limit our attributes extraction method to processing a single IoT pattern at a time. This was accomplished without jeopardising the experiment's findings. Data altering for the purposes of fitting the experimental setup is an appropriate practice. According to Saldanha's [47] view of corpus linguistics, data altering is permissible as long as it does not impact the text's validity or natural occurrence.

The steps for mining knowledge graphs are based on an entity-relationship model, which is the base primer for constructing knowledge graphs. In simple language, we focus on these three elements to mine knowledge graphs from text, namely: entities, their attributes and the relationships that exist between them. We treat each pattern name as an entity, and then use a vector space algorithm to determine a postulated relationship between the entities based on their attributes. The objective is to gather multiple documents that discuss the IoT patterns, and then extract important knowledge that can be used to engineer knowledge graphs. The idea is to build a sizeable knowledge base, which can ultimately be used as a platform for entity resolution and linking. This process involves four main steps, namely: i) entities extraction, ii) attributes extraction, iii) relations extraction, and iv) knowledge graph construction using joint probabilities. The entire process involves the building of four artefacts, namely: i) a topic modelling prototype for entities extraction, discussed in section 2.1.1. ii) a machine learning based text processing model, discussed in section 2.2. iii) a neural network-based relations embedding model, discussed in section 2.3.1., and iv) a knowledge graph model, discussed in section 2.4.

4. RESULTS AND DISCUSSION

This section presents results from applying topic modelling, attributes extraction, and relations embeddings models to the IoT patterns data. First, we review the performance of each model in isolation, and then proceed to look at how these models perform when used together to mine knowledge graphs for mapping relations between the IoT patterns.

4.1. Evaluating topic models

This section summarises the performance of topic modelling with regard to analysing the IoT patterns documents. To achieve this, we used three indicators or measures, namely: i) Model fit, ii) Analysis of clustered output and, iii) Analysis of the estimator for inference. Model fit measures the goodness of fit and typically summarises the discrepancy between observed values and the values expected under the model in question [27], [48]. Analysis of clustered output involves evaluating the clustered output or words under the topics [49]-[51]. Analysis of the estimator for inference involves analysing the number of iterations for convergence [52].

4.1.1. Model fit

Model fit measures the feasibility of using topic modelling to extract the correct topic from a set of given documents. To measure the goodness of fit, we adopt the Kolmogorov Smirnov Test which is briefly summarised in Appendix. For this exercise, we divided our documents into smaller subsets. In other words, we classified documents that described the same pattern into a subset, and applied topic modelling in an attempt to identify the pattern name as the collective topic for the documents. In this study, the primary objective of using topic modelling was to test its capability to identify the IoT pattern name in a given document as a potential topic. The aim of topic modeling has traditionally been to automatically discover secret (non-observable) topics in a set of documents [53]-[57]. However, LDA is applicable in our case since not all documents use the pattern names as titles. We evaluated 109 IoT patterns represented by more than 500 articles. On average each subset contained 15.3 documents, with 9.6 words per sentence. Our model fit reveals a score of 0.94 indicating that the correct topic was identified in 103 of the 109 subsets. This exhibits a high performance in using topic modelling for topic identification of subset data.

4.1.2. Analysis of clustered output

In traditional LDA, every word partially belongs to all topics or clusters with different probabilities expressed by a score of between 0 and 1. In this work, our aim was to optimise the probabilities of words extracted from the documents as discussed in section 2.2. These words are the key descriptors which characterise the core essence of each pattern. In simple terms, we want the top ranked clustered outputs under the correct topic to match the words extracted in section 2.2. For parameter optimisation, we only consider the top five ranked words as key descriptors for each topic. We found that the extracted attributes were mostly in the top 5 ranking for 99 of the sample subsets. An example of the 6LoWPAN subset with the clustered outputs is presented in Table 3. In Table 3, the word *`Wireless'*, for example, is a key descriptor of the 6LoWPAN technology, and is one of the identified attributes extracted using the technique described in section 2.2. As one would expect, this word is assigned a higher probability under the highest ranked topic.

4.1.3. Analysis of the estimator

In the LDA model, topics t_{dn} and words w_{dn} are generally viewed as discrete random variables and both are modelled using a discrete, or multinomial distribution. In this model, θ_d and ϕ are objects of

inference, which indicate the probabilities of the topics for each document d and the associated words w for each topic t. Our model involves T topics that has $\dim(\theta_d) = T$, and ϕ is an $M \times T$ matrix of probabilities for the M unique words that appear in the collection of the IoT patterns documents. The establishment of topics require running a number of iterations in order to reach convergence [58]. In our experiments, we used the collapsed Gibbs sampler for inferring the topic distributions [59]. We applied the empirical Markov Chain Monte Carlo (MCMC) diagnostic to determine convergence. In particular, we used the estimate potential scale reduction factor.

Figure 1 shows the convergence performance for inferring topic distributions. We test convergence on three levels, namely; i) per document, ii) per subset, and iii) per entire set. In Figures 1(a)-(c), we show time-series plots for all three performance levels. The results show that the parameters used for testing convergence differ significantly after an initial burn-in of 0 and 100 iterations. This indicate that the algorithm has not converged yet. Figure 1(a) shows that the algorithm converges faster for single documents, requiring approximately 150 iterations to converge. On average, the subsets as shown in Figure 1(b) start to converge at approximately 200 iterations, while the entire document set as shown in Figure 1(c) starts to converge at approximately 500 iterations. Figures 1(d)-(f) shows running mean plots for the same performance.

Table 3. Clustered outputs for the 6LowPAN pattern

Topics and Clustered Outputs		
6LowPAN	Standard	Wireless
Wireless = 0.91	IEEE = 0.80	Access = 0.62
DSSS = 0.90	Control = 0.76	Layer = 0.54
Network = 0.88	Low-rate = 0.51	Devices = 0.44
Encryption = 0.85	Physical = 0.50	Network = 0.41
Access = 0.83	Wi-Fi = 0.44	Control = 0.40

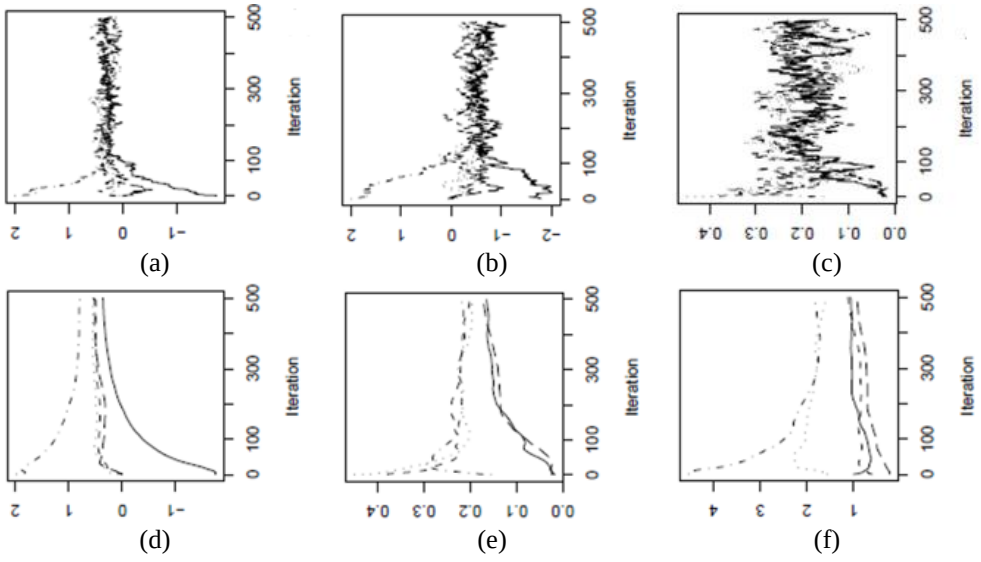

Figure 1. Convergence results for inferring topic distributions

4.2. Evaluation of attributes extraction

In this section, we present our evaluation results to demonstrate the performance of our attributes extraction system described in the previous section. Due to the novelty of the system, and the fact that the system is tested on a new data set, comparative and benchmark performance of related systems is not included in this study. The system results are, therefore, used to set a benchmark. The discussion in this section will concentrate on the most common results from the IoT trends that have been examined. These common results can be categorized into four major themes: i) the presence of the core attributes in the middle parts of the texts, ii) the section headers of sentences defining the core attributes of the IoT patterns, iii) the output of each individual feature, as well as iv) correlations between the features. The first two themes support our theory or hypothesis, while the last two themes examine the output of our features individually and in relation to one another.

4.2.1. Section headers of sentences describing the core attributes of the IoT patterns

We examined 109 documents discussing various IoT patterns and found that 79% or 87 of these documents contained section headers. The other 22 documents were informal documents (i.e., web pages) and did not contain any headings except for the title of the web page. From the 87 documents, we extracted a total of 261 attributes of which 72% or 190 of the attributes were under an identified section heading of a document. Most of the attributes extracted were in close proximity to each other, usually in an ordered list, under a section heading. In Table 4, we show an example of this in which the attributes of the IoT pattern device wake-up trigger, are in succession to each other. The Table 4 also shows the scores per feature for the given sentences.

Table 4. Device wake-up trigger attributes

	Feature Values					
Attributes	$h()$	$\ell()$	$t()$	$u()$	$\rho()$	$\mathcal{S}()$
#s61[a]	0.6	0.7	0.5	0.8	0.6	0.64
#s62[b]	0.5	0.5	0.5	0.8	0.7	0.6

[a] *Implement a mechanism that allows the server to send a trigger message to the device via a low energy communication channel.*
[b] *Have the device listening for these triggering messages and immediately establish communication with the server when it receives such a message.*

4.2.2. The occurrence of the core attributes within the specified thresholds

In our findings, the results reveal that 93% of the core characteristics of the various IoT patterns are generally found in the mid-section of the document represented by the range $[i, j]$. 4% of the attributes were located in the top part of the documents or within the range of $[i - 1]$. We also found that 3% of the attributes in the documents were located in the bottom part of the documents, represented by the range $[j + 1]$.

4.2.3. The overall performance of each independent feature

The overall performance of the five (5) features is presented in Table 5. We used recall and precision to measure the performance of these features. In this study, Recall determines how many sentences are returned that meet the criteria for a specific function or feature. In other words, Recall represents the totality of results for each feature on a scale of [0, 1]. Recall is measured as in (13). On the other hand, Precision measures the ratio of satisfactory results in proportion to the totality of results. Precision is calculated as in (14).

$$Recall = \frac{\#t_p}{\#t_p + \#f_p} \tag{13}$$

$$Precision = \frac{\#t_p}{\#t_p + \#f_p} \tag{14}$$

For instance, the Precision for the function $t()$ with regard to the number of declarative sentences in a given document is the number of how many sentences are terminated by the full stop symbol (.).

Table 5 shows that sentence type is the best performing metric for attributes extraction with a 100% accuracy score or a precision value of 1. Contextually, this means that all identified sentences from which attributes were extracted were terminated by a full stop. Lexical units, on the other hand, as a metric were the worst performing feature, demonstrating that several sentences that meet the lexical unit criteria were not classified as core IoT pattern attributes.

Table 5. Feature performance

Features	Precision
#type $t()$	1
#position $\rho()$	0.9
#length $\ell()$	0.7
#headings $h()$	0.65
#lexical units $u()$	0.56

4.2.4. Correlations between features

The correlations between the five metrics used in our attributes extraction method are shown in Table 6. The frequency and direction of association between any two features are measured by these correlations. We used Spearman's rank correlation of the form:

$$r_s = 1 - \frac{6\sum D^2}{n(n^2-1)} \quad (15)$$

The function (15) helps us to determine the degree to which two given features rate the same set of sentences. In Table 6, the symbol * marks those features that are significantly correlated. From these findings, we observe that $\rho()$ and $h()$ are highly correlated. The high correlation between these two features is expected given that the set of sentences that were identified as attributes were mainly under section headers, which coincided with sentences positioned in the range [i, j] of each document. From Table 6, we also observe that the other features have relatively low correlations. This means that these features are independent of each other.

Table 6. Rank correlation coefficients between features

Rank Correlations Coefficients					
Features	$t()$	$\rho()$	$\ell()$	$h()$	$u()$
#type $t()$	-	0.35	0.4	0.5	0.38
#position $\rho()$	0.6	-	0.45	0.89*	0.41
#length $\ell()$	0.25	0.4	-	0.3	0.55
#heading $h()$	0.38	0.47	0.51	-	0.49
#lexical units $u()$	0.4	0.55	0.37	0.43	-

4.3. Evaluation of relations embeddings

In this section, we present the evaluation results to demonstrate the performance of our vector space model for relations embeddings. For this task, a multi-threaded architecture is used to separately train progenitor relationships and their thesaurus-based synonyms from words taken from the rest of the document. In simple terms, relationships are trained in a specific assigned thread, and all the other generic words from the document are trained using a separate thread. In this case, words from both threads jointly and asynchronously update the output relations embeddings and determine the final relations vectors. In both threads, we set the symmetric context window size to 20 (i.e., 10 words to the right and 10 words to the left of the target word). We use differential learning to manipulate and control each thread's contribution to the final word embeddings. Specifically, we control the output of the final word embeddings by controlling the behaviour of the stochastic gradient descent function.

$$\downarrow \Theta_j \coloneqq \Theta_j - \alpha \cdot \frac{\partial}{\partial \Theta_j} J(\Theta) \quad (16)$$

We set a small learning rate, α = 0.01, for training progenitor relationships and their synonyms. To marginalise the contribution of all the other words in the documents, we set a higher learning rate, $\alpha = 0.5$, to reduce optimisation of the thread to converge to a good local minimum. Intuitively, this results in less contribution from the generic words.

This approach allows us to spot a progenitor relationship if it presents in the text, or if any of its synonyms is used in the contents of the document. Using cosine similarity, we obtained a score of 86% success rate which shows the model's capability to recognise numerous various expressions used by the authors to define relationships between the IoT patterns. If the progenitor relationship (or any of its synonyms) does not appear in the contents of the document, we select the word with the highest cosine similarity to either the progenitor relationship or any of its synonyms.

4.4. Graph construction experiments

From our data set, we managed to get approximately 34 109 extraction graphs with nearly 42 812 vertices labels. Our main task was to collectively construct a set of knowledge graphs and evaluate them based on the existing 10 384 target knowledge graphs in our benchmark database. Table 7 above shows the results of evaluating knowledge graph construction. In the table, the Average Confidence of Extractions represent the candidate facts stored in the database against the graphs that are inputted into the database for a potential match. In this study, we use confidence levels to specify a range of values that are likely to contain the true population mean on the basis of a sample. Through our experiments, we found that 71% or 0.715 extracted graphs were likely to contain correct knowledge graphs. Figure 2 shows re-produced excerpts of some knowledge graphs generated using our proposed process. The graphs are extracted from the Neo4j graph platform and have been reproduced for clear visibility. The direction of the edges in our graphs is based on the chronological appearance of the entities within the document. In other words, the entity that appears first in a given document is the source node and the entity that appears in the later section of the

document is the target node. Ontological rules are said to be satisfied if the extracted graph contains no violations of ontological validations and inference. The results reveal that 87% or 0.874 of the extraction graphs satisfied the pre-defined ontological rules. The results also show that only 63% or 0.633 of the knowledge graphs were linked successfully by virtue of the semantic rules.

Table 7. Evaluating knowledge graph construction

Indicator	Performance
Average Confidence of Extractions	0.715
Satisfaction of Ontological Rules	0.874
Satisfaction of Semantic Rules	0.633

Figure 2. Excerpts of some IoT patterns and their relationships

5. CONCLUSION

Current classification schemes for organising the IoT patterns are generally manual. In the literature, the use of machine learning techniques for intelligent classification schemes in this area have been given little attention. In this paper, we have proposed a multifaceted process that is data-driven and uses machine learning techniques to mine knowledge graphs that outlines the interoperability of the IoT patterns. We used a combination of techniques to extract entities, their attributes, and the relationships between them from a

large set of documents. While knowledge graphs are amongst the technologies with the fastest growing curves, at the moment, there is no accepted technique for mining these knowledge graphs from natural text. The results reveal that topic modelling, attributes extraction and word embeddings can be used successfully to capture essential knowledge required to model the IoT pattern relationships using knowledge graphs. The findings of this study contribute towards closing a research gap in the IoT paradigm, particularly with regard to building an automatic-generated repository for the IoT patterns. This repository can be used as a reference point for the construction of the IoT reference architecture, and can also assist the IoT practitioners with quick, seamless retrieval of these patterns.

APPENDIX

A. Semi-supervised guided LDA model

The approach followed in our study is to allow the user to direct the topic discovery process by providing seed information at the level of topic and word selection. In simple terms, the user is aware of the topics that represent the set of documents, and also understands the words that must be allocated to each of those topics. In section 2.2 of this study, we have described in some details the process of extracting these words. Statistically, the seeded topics are given a 10% boost and the seeded words are given the same seed confidence toward the seeded topics. Intuitively, the algorithm is as algorthm 1.

Algorithm 1: A Guided LDA Algorithm

1. **For each** k **= 1, ..., T**
   (a) **choose** a regular topic. $\phi_k^r \sim Dir(\beta_r)$
   (b) **choose** a seed topic $\phi_k^s \sim Dir(\beta_s)$.
   (c) **choose** $\pi_k \sim Beta(1,1)$.

2. **For each document** d, choose $\theta_d \sim Dir(\sigma)$.
   - For each token i = 1, ..., N_d:
   (a) **select** a topic $Z_i \sim Mult(\theta_d)$, apply seed confidence μ to the seeded topic.
   The probability will be higher for the seeded topic, $p(z|w,d)\sigma\, z \epsilon\, d$.
   (b) **select** a word $W_i \sim Mult(\phi_{zi})$apply seed confidence μ to the seeded words.
   The probability p of seeded words W belonging to topic Z will be higher, $p(w|z,d)\sigma\, w \epsilon\, z\, p$

For a detailed description of how to incorporate lexical priors into topic models, the reader is encouraged to refer to Jagarlamudi [60] for additional details.

B. Optimisation regression algorithm

To learn the correct weights for the attributes' extraction features, we change regression weights after every iteration according to the gradient. As stated in section 2.2 of this paper, we initialise the model with equal weights for all features. We select a constant initialisation scheme to allow features to have an identical influence on the cost and identify better performing features in the initial round of performance. This allows us to determine the allocation of the initialised weights for the features. In this case, the weights are still random but differ in terms of range and magnitude depending on the performance of the initial phase. This provides a controlled initialisation that is more accurate and performs faster while also giving a more efficient gradient descent.

Algorithm 2: Optimisation Regression Algorithm

Optimisation Linear Regression (D**,** Number of iterations**)**
  Initialise weights $\omega = (\omega_0, \omega_1, \omega_2, \ldots, \omega_d)$
    For $i = 1{:}1$ - Number of iterations
      do
        select a data point $D_i = (x_i, y_i)$ from D
        set $\alpha = \frac{1}{i}$
        update weight vector
        $\omega \leftarrow \omega + \alpha\big(y_i - f(x_i, \omega)\big)x_i$
      End for
      return weightsω

REFERENCES

[1] M. Ganesan and N. Sivakumar, "A survey on IoT related patterns," *International Journal of Pure and Applied Mathematics*, vol. 117, no. 19, pp. 365-369, 2017.

[2] S. Qanbari *et al.*, "IoT design patterns: computational constructs to design, build and engineer edge applications," *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2016, pp. 277-282, doi: 10.1109/IoTDI.2015.18.
[3] G. S. Chandra, "Pattern language for iot applications," *Pattern Languages of Programs Conference*, pp. 1-8, 2016.
[4] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017, Art. no. 9324035, doi: 10.1155/2017/9324035.
[5] H. Fu, G. Manogaran, K. Wu, M. Cao, S. Jiang, and A. Yang, "Intelligent decision-making of online shopping behavior based on internet of things," *International Journal of Information Management*, vol. 50, pp. 515-525, 2020, doi: 10.1016/j.ijinfomgt.2019.03.010.
[6] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076-8094, 2019, doi: 10.1109/JIOT.2019.2920987.
[7] M. b. M. Noor and W. H. Hassan, "Current research on internet of things (IoT) security: A survey," *Computer networks*, vol. 148, pp. 283-294, 2019, doi: 10.1016/j.comnet.2018.11.025.
[8] L. Reinfurt, U. Breitenbücher, M. Falkenthal, F. Leymann, and A. Riegg, "Internet of things patterns," *Proceedings of the 21st European Conference on Pattern Languages of Programs*, 2016, pp. 1-21, Art. no. 5, doi: 10.1145/3011784.3011789.
[9] W. Shi, Y. Guo and Y. Liu, "When flexible organic field-effect transistors meet biomimetics: A prospective view of the internet of things," *Advanced Materials*, vol. 32, no. 15, 2020, Art. no. 1901493, doi: 10.1002/adma.201901493.
[10] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724-4734, 2018, doi: 10.1109/TII.2018.2852491.
[11] M. Koster, "Design patterns for an internet of things: A design pattern framework for iot architecture," 2014. [Online]. Available: https://community.arm.com/iot/b/blog/posts/design-patterns-for-an-internet-of-things.
[12] T. Kliegr and O. Zamazal, "Lhd 2.0: A text mining approach to typing entities in knowledge graphs," *Journal of Web Semantics*, vol. 39, pp. 47-61, 2016, doi: 10.1016/j.websem.2016.05.001.
[13] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic web*, vol. 8, no. 3, pp. 489-508, 2017, doi: 10.3233/SW-160218.
[14] J. Pujara and S. Singh, "Mining knowledge graphs from text," *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 789-790, doi: 10.1145/3159652.3162011.
[15] V. Sithole and L. Marshall, "Attributes extraction for fine-grained differentiation of the internet of things patterns," *Proceedings of the South African Institute of Computer Scientists and Information Technologists 2019*, 2019, pp. 1-10, Art. no. 9, doi: 10.1145/3351108.3351118.
[16] J. De Loof *et al.*, "Internet of things-architecture iot-a deliverable d1. 5-final architectural reference model for the iot v3.0," *IoT-A (257521)*, pp. 1-499, 2013.
[17] S. Rubin, "Knowledge discovery and dissemination of text by mining with words," *Publication of US8285728B1*, 2012, uS Patent 8,285,728.
[18] P. Qi, T. Dozat, Y. Zhang, and C. D. Manning, "Universal dependency parsing from scratch," *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 2018, pp. 160-170.
[19] E. Kiperwasser and Y. Goldberg, "Simple and accurate dependency parsing using bidirectional LSTM feature representations," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 313-327, 2016, doi: 10.1162/tacl_a_00101.
[20] B. G. Banik and S. K. Bandyopadhyay, "Novel text steganography using natural language processing and part-of-speech tagging," *IETE Journal of Research*, vol. 66, no. 3, pp. 384-395, 2020, doi: 10.1080/03772063.2018.1491807.
[21] C. Wang, "Performance evaluation of English part-of-speech tagging based on typical parameter smoothing algorithm," *Journal of Physics: Conference Series*, vol. 1533, 2020, Art. no. 022032, doi: 10.1088/1742-6596/1533/2/022032.
[22] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1-20, 2020, doi: 10.1109/TKDE.2020.2981314.
[23] J. M. Giorgi and G. D. Bader, "Towards reliable named entity recognition in the biomedical domain," *Bioinformatics*, vol. 36, no. 1, pp. 280-286, 2020, doi: 10.1093/bioinformatics/btz504.
[24] S. I. Nikolenko, S. Koltcov, and O. Koltsova, "Topic modelling for qualitative studies," *Journal of Information Science*, vol. 43, no. 1, pp. 88-102, 2017, doi: 10.1177/0165551515617393.
[25] J. M. Luo, H. Q. Vu, G. Li, and R. Law, "Topic modelling for theme park online reviews: analysis of disneyland," *Journal of Travel & Tourism Marketing*, vol. 37, no. 2, pp. 272-285, 2020, doi: 10.1080/10548408.2020.1740138.
[26] V. Gangadharan and D. Gupta, "Recognizing named entities in agriculture documents using LDA based topic modelling techniques," *Procedia Computer Science*, vol. 171, pp. 1337-1345, 2020, doi: 10.1016/j.procs.2020.04.143.
[27] J. Büschken and G. M. Allenby, "Sentence-based text analysis for customer reviews," *Marketing Science*, vol. 35, no. 6, pp. 953-975, 2016, doi: 10.1287/mksc.2016.0993.
[28] X. Sun, S. Sun, M. Yin, and H. Yang, "Hybrid neural conditional random fields for multi-view sequence labeling," *Knowledge-Based Systems*, vol. 189, 2020, Art. no. 105151, doi: 10.1016/j.knosys.2019.105151.
[29] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence* Review, vol. 53, pp. 5455-5516, 2020, doi: 10.1007/s10462-020-09825-6.

[30] P. Yao *et al.*, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641-646, 2020, doi: 10.1038/s41586-020-1942-4.
[31] D.-X. Zhou, "Universality of deep convolutional neural networks," *Applied and computational harmonic analysis*, vol. 48, no. 2, pp. 787-794, 2020, doi: 10.1016/j.acha.2019.06.004.
[32] N. Somu, M. R. Gauthama Raman, and K. Ramamritham, "A hybrid model for building energy consumption forecasting using long, short term memory networks," *Applied Energy*, vol. 261, 2020, Art. no. 114131, doi: 10.1016/j.apenergy.2019.114131.
[33] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, 2020, Art. no. 132306, doi: 10.1016/j.physd.2019.132306.
[34] C. Nobata, S. Sekine, M. Murata, K. Uchimoto, M. Utiyama, and H. Isahara, "Sentence extraction system assembling multiple evidence," *in NTCIR*, pp. 1-6, 2001.
[35] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264-285, 1969, doi: 10.1145/321510.321519.
[36] A. Kanapala, S. Pal, and R. Pamula, "Text summarization from legal documents: a survey," *Artificial Intelligence Review*, vol. 51, no. 3, pp. 371-402, 2019, doi: 10.1007/s10462-017-9566-2.
[37] T. Jayalakshmi and A. Santhakumaran, "Statistical normalization and back propagation for classification," *International Journal of Computer Theory and Engineering*, vol. 3, no. 1, pp. 1793-8201, 2011.
[38] A. Afshari, M. Mojahed, and R. M. Yusuff, "Simple additive weighting approach to personnel selection problem," *International Journal of Innovation, Management and Technology*, vol. 1, no. 5, pp. 511-515, 2010, doi: 10.7763/IJIMT.2010.V1.89.
[39] W. Croft, "Parts of speech as language universals and as language-particular categories," *Empirical Approaches to Language Typology*, pp. 65-102, 2000, doi: 10.1515/9783110806120.65.
[40] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," *Thirtieth Aaai conference on artificial intelligence*, 2016, pp. 1955-1961.
[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
[42] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling wordembedding method," *arXiv preprint arXiv:1402.3722*, 2014.
[43] K. Singh *et al.*, "Towards optimisation of collaborative question answering over knowledge graphs," *arXiv preprint arXiv:1908.05098*, 2019.
[44] Y.-L. Chi, "Rule-based ontological knowledge base for monitoring partners across supply networks," Expert Systems with Applications, vol. 37, no. 2, pp. 1400-1407, 2010, doi: 10.1016/j.eswa.2009.06.097.
[45] E. F. Aminu, I. O. Oyefolahan, M. B. Abdullahi, and M. T. Salaudeen, "A review on ontology development methodologies for developing ontological knowledge representation systems for various domains," *International Journal of Information Engineering & Electronic Business*, vol. 12, no. 2, pp. 28-39, 2020, DOI:10.5815/ijieeb.2020.02.05.
[46] S. Lovrencic and M. Cubrilo, "Ontology evaluation-comprising verification and validation," *Central European Conference on Information and Intelligent Systems. Faculty of Organization and Informatics Varazdin*, 2008, pp. 1-7.
[47] G. Saldanha, "Principles of corpus linguistics and their application to translation studies research," *Tradumàtica: traducció i tecnologies de la informació i la comunicació*, no. 7, pp. 1-7, 2009.
[48] F. Tian, B. Gao, D. He, and T.-Y. Liu, "Sentence level recurrent topic model: letting topics speak for themselves," *arXiv preprint arXiv:1604.02038*, 2016.
[49] D. Surian, D. Q. Nguyen, G. Kennedy, M. Johnson, E. Coiera, and A. G. Dunn, "Characterizing twitter discussions about hpv vaccines using topic modeling and community detection," *Journal of medical Internet research*, vol. 18, no. 8, 2016, Art. no. e232, doi: 10.2196/jmir.6045.
[50] M. Reisenbichler and T. Reutterer, "Topic modeling in marketing: recent advances and research opportunities," *Journal of Business Economics*, vol. 89, no. 4, pp. 327-356, 2019, doi: 10.1007/s11573-018-0915-7.
[51] L. Liu, L. Tang, W. Dong, S. Yao, and W. Zhou, "An overview of topic modeling and its current applications in bioinformatics," *SpringerPlus*, vol. 5, no. 1, 2016, Art. no. 1608, doi: 10.1186/s40064-016-3252-8.
[52] W. Peng and T. Sun, "Method and system for identifying a key influencer in social media utilizing topic modeling and social diffusion analysis," *Science On, uS Patent 8,312,056*, 2012.
[53] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," *SOMA '10: Proceedings of the First Workshop on Social Media Analytics*, 2010, pp. 80-88, doi: 10.1145/1964858.1964870.
[54] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith, "Interactive topic modeling," *Machine learning*, vol. 95, no. 3, pp. 423-469, 2014, doi: 10.1007/s10994-013-5413-0.
[55] X. Sun, X. Liu, B. Li, Y. Duan, H. Yang, and J. Hu, "Exploring topic models in software engineering data analysis: A survey," *2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2016, pp. 357-362, doi: 10.1109/SNPD.2016.7515925.
[56] Q. Mei, D. Cai, D. Zhang, and C. Zhai, "Topic modeling with network regularization," *WWW '08: Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 101-110, doi: 10.1145/1367497.1367512.
[57] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," *WWW '08: Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 111-120, doi: 10.1145/1367497.1367513.
[58] H. M. Wallach, "Topic modeling: beyond bag-of-words," *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 977-984, doi: 10.1145/1143844.1143967.
[59] S. E. Dosso, "Quantifying uncertainty in geoacoustic inversion. i. a fast gibbs sampler approach," *The Journal of the Acoustical Society of America*, vol. 111, no. 1, pp. 129-142, 2002, doi: 10.1121/1.1419086.

[60] J. Jagarlamudi, H. Daumé, and R. Udupa, "Incorporating lexical priors into topic models," *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics*, 2012, pp. 204-213.

BIOGRAPHIES OF AUTHORS

Vusi Sithole is a PhD Candidate at the University of Pretoria, Department of Computer Science with a Master of Information Technology from the same University (2016). He obtained a bachelor's degree in Information Sciences from the same institution in 2009. His research interests are in fields of Software design patterns, Internet of Things, Machine Learning, Graph Theory, and Computational Linguistics. The focus of his PhD thesis is on establishing a multifaceted approach for organising patterns for the Internet of Things. This includes subjects such as attributes extraction, building concept hierarchies for the IoT patterns, definition formation for the patterns, and engineering a pattern locator technique for the known patterns. He has published several peer-reviewed papers in his areas of research and continues to do ground-breaking research to advance humanity.

Linda Marshall is a Senior Lecturer at the University of Pretoria, Department of Computer Science. She is the head of the Computer Science Education (SCEDAR) research group at the Computer Science Department, University of Pretoria. She has been the Director of the ACM ICPC South African Regional contest since 2001. Her research interests are in the areas of Generic Programming, Graph Comparison, Software Engineering and more particularly Computer Science Education. She has published extensively in these areas and continues to supervise numerous students undertaking research projects in many of these areas and other emerging fields such as Artificial Intelligence and Machine Learning.
Further info on her homepage: http://www.cs.up.ac.za/~lmarshall/