Word Vector Embeddings and Domain Specific Semantic based Semi-Supervised Ontology Instance Population

Vindula Jayawardana[#1], Dimuthu Lakmal[#2], Nisansa de Silva[#3], Amal Shehan Perera[#4], Keet Sugathadasa[#5], Buddhi Ayesha[#6], Madhavi Perera[*7]

***Abstract*— An ontology defines a set of representational primitives which model a domain of knowledge or discourse. With the arising fields such as information extraction and knowledge management, the role of ontology has become a driving factor of many modern-day systems. Ontology population, on the other hand, is an inherently problematic process, as it needs manual intervention to prevent the conceptual drift. The semantic sensitive word embedding has become a popular topic in natural language processing with its capability to cope with the semantic challenges. Incorporating domain specific semantic similarity with the word embeddings could potentially improve the performance in terms of semantic similarity in specific domains. Thus, in this study we propose a novel way of semi-supervised ontology population through word embeddings and domain specific semantic similarity as the basis. We built several models including traditional benchmark models and new types of models which are based on word embeddings. Finally, we ensemble them together to come up with a synergistic model which outperformed the candidate models by 33% in comparison to the best performed candidate model.**

***Keywords* — Ontology, Ontology Population, Word Embeddings, Word2vec, Semantic Similarity.**

I. Introduction

In various computational tasks in many different fields, the use of ontologies is becoming increasingly involved. Many of the research areas such as knowledge engineering and representation, information retrieval and extraction, and knowledge management and agent systems [1] have incorporated the use of ontologies to a greater extent. As defined by Thomas R. Gruber [2], an ontology is a "formal and explicit specification of a shared conceptualization". Due to the evolving ability of ontologies to overcome limitations in traditional natural language processing methods, the popularity of using ontologies in modern computation tasks are getting increased day by day. For an example, text classification [3], [4], word set expansions [5], linguistic information management [6], [7], and information extraction [8], [9] emphasize the growing popularity of the ontology based computations and processing.

According to Carla Faria et al. [10], ontology population looks for instantiating the constituent elements of an ontology like properties and non-taxonomic relationships. However, most of the time, ontology populations are done by domain experts and knowledge engineers as a manual process, which is both time consuming and expensive. As majority of the world's knowledge is encoded in natural language text, automating the population of these ontologies using results obtained from Natural Language Processing (NLP) based analysis of documents has recently become a major challenge for NLP applications [11].

In this study, we propose a novel way for semi-supervised instance population of an ontology using word vector embeddings. Word Embeddings could be identified as a collective name for a set of language modelling and feature learning techniques in natural language processing. The basic idea behind word embedding is based on the concept where words or phrases from the vocabulary are mapped to vectors of real numbers. We use these vectors as a method of arriving at instance population in an ontology. For this purpose, we built an iterative model based on the class representative vector for ontology classes [12]. In our implementation, we built multiple models based on different methodologies. In one model we assigned membership to natural language tokens by distance to the representative vectors. In another, we used dissimilar exclusion method to identify the membership. Set expansion as described by [5], was used in another model for the purpose of ontology population. Finally, we used two semi-supervised models based on k-means clustering and hierarchical clustering. As each model outputs a set of candidate words for a given class, we then collaborate with domain experts and knowledge engineers to identify the performance of each model and to build an ensemble model as the final resultant model. We intend to demonstrate the use of domain specific semantic similarity in defining the similarities between instances and classes.

As allowed by the nature of the defined models, we use the domain specific semantic similarity measure [13] as the distance measure.

Semi-supervised learning falls between unsupervised learning (without any labelled training data) and supervised learning (with completely labelled training data). It has been observed that many machine leaning approaches elucidate considerable improvement in learning accuracy, when unlabelled data is used in conjunction with a small amount of labelled data.

Manuscript received on 09th Dec, 2017. Recommended by Dr. M. G. N. A. S. Fernando on 12th June, 2018.
This paper is an extended version of the paper "Semi-Supervised Instance Population of an Ontology using Word Vector Embedding" presented at the ICTer 2017.

Vindula Jayawardana Dimuthu Lakmal, Nisansa de Silva, Amal Shehan Perera, Keet Sugathadasa and Buddhi Ayesha are from Department of Computer Science & Engineering, University of Moratuwa. (*vindula.13@cse.mrt.ac.lk, kjtdimuthu.13@cse.mrt.ac.lk, nisansadds@cse.mrt.ac.lk, shehan@cse.mrt.ac.lk, keetmalin.13@cse.mrt.ac.lk, buddhiayesha@cse.mrt.ac.lk*).
Madhavi Perera is from University of London International Programmes, University of London, UK (*madhaviperera58@gmail.com*).

The legal context contains jargon which is complex and most of the time impossible to store in mind; whether it be an average person or a paralegal, given that it consists terminology derived from ancient Latin terms, as well as various distinctive terminology depending on the category of laws and the geographical settings of practice. Therefore, knowing them manually is rather an impossible task which drove us to select the legal domain for this study of semi-supervised ontology population.

The rest of this paper is organized as follows: In Section II we review previous studies related to this work. The details of our methodology for semi-supervised instance population of an ontology using word vector embeddings is introduced in Section III. In Section IV, we demonstrate that our proposed methodology produces superior results outperforming traditional approaches. Finally, we conclude and discuss some future works in Section V.

II. Background and Related Work

The following sections depict the background of this study and other related studies.

A. Ontologies

Ontologies are mainly used to organize information as a form of knowledge representation in many areas. As defined by Thomas R. Gruber [2], "ontologies are an explicit and formal specifications of the terms in the domain and the relations among them". Ontologies have been expanding out from the realm of Artificial-Intelligence to domain specific tasks such as: Linguistics [4], [5], [14]–[16], Law [12], Medicine [6], [7], [9]. Ontologies have become common on the semantic iteration of the World-Wide Web [17]. An ontology may model either the world or a part of it as seen by the said area's viewpoint [5].

The basic ground units of an ontology are the *Individuals* (instances). By grouping these *Individuals* which can either be concrete objects or abstract objects, the structures called *classes* are built. A *class* in an ontology is a representation of a concept, type, category, or a kind. However, these definitions may be altered depending on the domain of the ontology. Often these *classes* form taxonomic hierarchies among them by subsuming, or being subsumed by, another class.

B. Word Vector Embeddings

As first proposed by Tomas Mikolov et al. [18] word embedding systems, are a set of natural language modelling and feature learning techniques, where words from a domain are mapped to vectors to create a model that has a distributed representation of words. Word2vec[1] [19], GloVe [20], and Latent Dirichlet Allocation (LDA) [21] are the leading Word Vector Embedding systems. However, due to the flexibility and ease of customization, we picked word2vec as the word embedding method for this study.

Word2vec is a neural network with two layers, which uses a large corpus of text as an input and outputs a vector space, typically of several hundred dimensions for the given corpus of text. Word2vec trains neural network to reconstruct the linguistic contexts of words utilizing either of two methods: continuous bag-of-word (CBOW) or continuous skip-gram. In continuous bag-of-words method, the model predicts the current word from a windows of surrounding context words. In the continuous skip-gram method, the model uses the current word to predict the surrounding window of the context words. Word2vec can be adapted to provide similar terms for an input term and facilitate vector operations with a high degree of accuracy.

Word2vec has been used in many areas due to its capability in coping up with the challenge of preserving the semantic sensitivity of a given context. It has been used in sentiment analysis [22]–[25] and text classification [26]. Gerhard Wohlgenannt et al. [27]'s approach to emulate a simple ontology using word2vec and Harmen Prins [28]'s usage of word2vec extension: node2vec [29], to overcome the problems in vectorization of an ontology, are two major works that have been carried out in relation to ontologies with the use of word2vec. More recently there have been successful studies on using word2vec on the legal domain [12], [13].

C. Word Set Expansion

Word lists that contain closely related sets of words is a critical requirement in machine understanding and processing of natural languages. Creating and maintaining such closely related word lists is a complex process that requires human input and is carried out manually in the absence of tools [5]. The said word-lists usually contain words that are deemed to be homogeneous in the level of abstraction involved in the application. Thus, two words W1 and W2 might belong to a single word-list in one application, but belong to different word-lists in another application. This fuzzy definition and usage is what makes creation and maintenance of these word- lists a complex task.

De Silva et al. [5] describe a supervised learning mechanism which employs a word ontology to expand word lists containing closely related sets of words. This study has been an extension of their previous work [15], which was done to enhance the refactoring process of the RelEx2Frame component of OpenCog AGI Framework, by expanding concept variables used in RelEx. The expected outcome of the project has a significant effectiveness on applications which fit into real life IT solutions which are related to natural language domain. Mainly AI related applications which require English language processing would benefit from the project. Moreover, output of the project can be utilized in a vast area of applications related to English language such as chat applications, text critiquing, information retrieval from the web, question answering, summarization and translations, rather than focusing on specific area of applications of English language.

D. Ontology Population

Being a knowledge acquisition task, ontology population is inherently a complex activity. Ontology population has been approached by using techniques such as rule based and machine learning. SPRAT [30] combines aspects from traditional named entity recognition, ontology-based information extraction, and relation extraction, in order to identify patterns for the extraction of a variety of entity types and relations between them, and to re-engineer them into concepts and instances in an ontology.

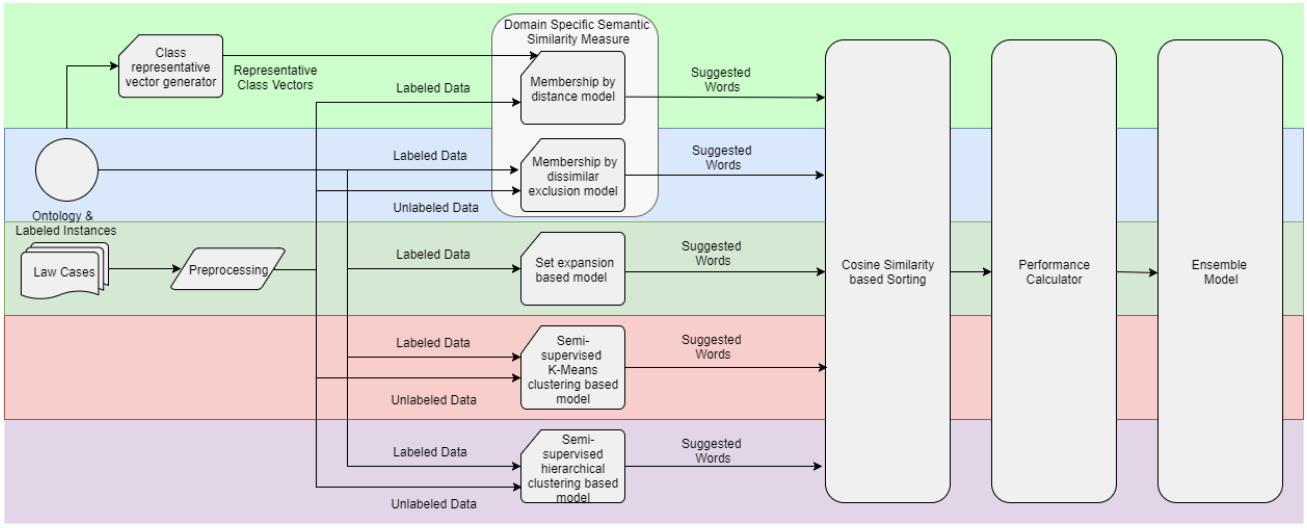


Fig 1. Flow of semi-supervised instance populating of an ontology using word vector embedding

Since majority of world's knowledge concentrated in natural language text, it is vital to take the knowledge extracted from natural language analysis, into account when populating an ontology in any given domain. Natural language analysis frameworks such as GATE have been introduced with the aim of facilitating NLP application development. In GATE, natural language processing tasks such as tokenization, POS tagging, or chunking are supported by integrating existing components into complex application pipelines. Nevertheless, exporting results of GATE natural language analysis into ontology still requires high degree of human intervention. Rene Witte et al. [11] have implemented a GATE processing resource namely *OWLExporter* that empowers automation of ontology population from text for an existing application pipeline. It yields a number of novel features such as exporting sentences, noun and verb phrase chunks, and integrating reasoning support for conference chains, to overcome said issues with ontology population using GATE. Moreover, it allows language engineers to create ontology population systems without requiring extensive knowledge of ontology APIs.

However modern-day researches are more focused on semi supervised ontology population due to the nature of less manual intervention.

E. Domain Specific Semantic Similarity

In almost all Natural Language Processing (NLP) tasks such as Information Retrieval, Information Extraction, and Natural Language Understanding (NLU) [8], semantic similarity measurements based on linguistic features are a fundamental component. Methods that treat words as independent atomic units are not sufficient to capture the expressiveness of language [19]. A solution to this is word context learning methods [15], [31]. Another solution is lexical semantic similarity based methods [4]. Both of these approaches try to capture semantic and syntactic information of a word.

Lexical Semantic similarity of two entities is a measure of the likeness of the semantic content of those entities. This likeness of the semantic content of the entities are most commonly calculated with the help of topological similarity existing within an ontology such as WordNet [32]. Wu and Palmer proposed a method to give the similarity between two words in the 0 to 1 range [33]. In comparison, Jiang and Conrath proposed a method to measure the lexical semantic similarity between word pairs using corpus statistics and lexical taxonomy [34]. Hirst & St-Onge's system [35] quantifies the amount that the relevant synsets are connected by a path that is not too long and that does not change direction often. In [4], the strengths of each of these algorithms were evaluated by means of the tool WS4J3.

However, Semantic similarity measures built for general use do not perform well within specific domains. Law and Medical [36] fields are the fields which suffer from this issue drastically. Therefore, Sugathadasa et al. [13] have introduced a domain specific semantic similarity measure that has been created by the synergistic union of word2vec, a word embedding method that is used for semantic similarity calculation and lexicon based (lexical) semantic similarity methods. According to Sugathadasa et al., while for word context learning, word embedding method, word2vec [12], [19]. has been used, number of lexical semantic similarity measures [33]–[35] have been used to augment and improve the results.

F. Semi Supervised Ontology Population

Although supervised machine learning methodologies have showed promising results when it comes to information extraction, they accumulate more cost for training since they require vast number of labelled training data. As a solution, semi-supervised machine learning methodologies have been introduced, requiring considerably less amount of labelled training data.

Carlson et al. [37] proposed a semi-supervised learning model to populate instances of a set of target categories and relations of an ontology by providing seed labelled data and a set of constraints which couples classes and relationships of an ontology. Semi-supervised algorithms tend to show unacceptable results due to 'semantic drift' and constraints have been introduced to overcome the issue. Carlson et al. have used 'Bootstrapping' method for semi-supervised learning which starts with a small number of labelled data and grows labelled data iteratively, which are chosen from a set of candidates, which is classified using the current semi-supervised model. Three types of constraints have been introduced by Carlson et al. to conform mutual exclusion, type checking, and text features.

Carlson et al. [38] have expanded coupled semi-supervised learning [37] to never-ending language learning (NELL); an agent that runs forever to extract information from the web and populate them continuously into a knowledge base. A prototype of the system that they have implemented is able to extract noun phrases related to various semantic categories, and semantic relations between categories. Its information extracting ability increases day by day which is evidenced by the ability to extract more information from previous day's text sources more accurately. Input ontology in the system was included with seed instances for each ontology class and then sub systems which consist of previously described coupled semi supervised methodologies extract candidate instances and relationships from the text corpus. Knowledge Integrator of the system chooses strongly supported sets of instances and relations from the candidate set, as new beliefs of the system.

Zhilin Yang et al. [39] have presented a semi supervised learning methodology based on graph embeddings. The system consists of two main sections namely 'transductive' and 'inductive'. The 'transductive' approach predicts instances which are already observed in the graph in the training period. In 'inductive' approach, predictions can be made on unobserved instances in the training period. A probabilistic model was developed to learn node embeddings to generate edges in a graph.

Jie Liu et al. [40] have proposed a method of similarity aggregation using SVM is to classify weighted similarity vectors which are calculated using concept name and properties of individuals of ontologies.

III. Methodology

We discuss the methodology used in this study in this section. Each of the following subsections describe a step of our process. An overview of the methodology we propose is illustrated in Fig. 1.

A. Ontology Creation

For the ontology creation, we focused on the consumer protection law of the United State legal system as the domain of interest and created a legal ontology. This legal ontology was developed by based on Findlaw [41] as the reference. The ontology creating process was an iterative process where, upon adding parts of legal domain knowledge to the ontology, a validation phase is run by the domain experts. However, to improve the clarity of this paper, we extract a sub-ontology from it and use it to explain the methodology to make the process simple and intuitive to understand. In selecting a part of the ontology, we mainly focused on more sophisticated relationships and taxonomic presences. An overview of the selected part of the ontology is illustrated in Fig. 2. After the creation of sub-ontology, we manually populated the ontology with seed instances for each ontology class. For this phase as well, we incorporated the domain experts' knowledge and the collaboration of knowledge engineers.

B. Training word Embeddings

The word embeddings method used in this study was built using a word2vec model. We obtained a large legal text corpus from Findlaw [41] and built a word2vec model using the corpus. The reason for selecting word2vec word embedding for this study is the success demonstrated by other studies such as [12] and [13] in the legal domain that uses word2vec as the word embedding method. The text corpus consisted of legal cases under 78 law categories. In creating the legal text corpus, we used the Stanford CoreNLP for preprocessing the text with tokenizing and sentence splitting. Fig. 3(a) illustrates the Natural Language Processing pipeline we used in pre- processing the text corpus.

Following are the important parameters we specified in training the model.

- size (dimensionality): 200
- context window size: 10
- learning model: CBOW
- min-count: 5
- training algorithm: hierarchical softmax

C. Deriving Representative Class Vectors

Ontology classes are sets of homogeneous instance objects that can be converted to a vector space by word vector embeddings. A methodology to derive a representative vector for ontology classes, whose instances were mapped to a vector space is presented in [12]. We followed the same approach and started by deriving five candidate vectors which are then used to train a machine learning model that would calculate a representative vector for each of the classes in the selected sub-ontology shown in Fig. 2. In the following sections, we describe in-depth, the manner in how we used this derived class vectors in our proposed methodology.

D. Instances Corpus for Ontology Population

In order to perform semi-supervised ontology population, we used legal cases from Findlaw [41] to create an instances corpus. We performed *Stanford CoreNLP* based pre-processing on the raw text with tokenizing and sentence splitting to generate the instance corpus. This legal corpus was used in the subsequent models for the purpose of ontology population.

E. Domain Specific Semantic Similarity Measure

In order to measure the domain specific semantic similarities, we used the methodology proposed by Sugathadasa at el. [13]. Fig. 3(b) indicates the high-level overview of building the domain specific semantic similarity model as per [13]. Depending upon the nature of the models we train, we intend to use this trained model in subsequent actions.

F. Candidate Model Building

Based on the aforementioned components, we built five candidate models for semi-supervised instance population of the ontology. The five models are illustrated below:

- Membership by distance model (M_1)
- Membership by dissimilar exclusion model (M_2)
- Set expansion based model (M_3)
- Semi-supervised K-Means clustering based model (M_4)
- Semi-supervised hierarchical clustering based model (M_5)

In the subsequent sections, we use the *IndexOf* function as defined by Equation 1. Here, X is the index of element y in S.

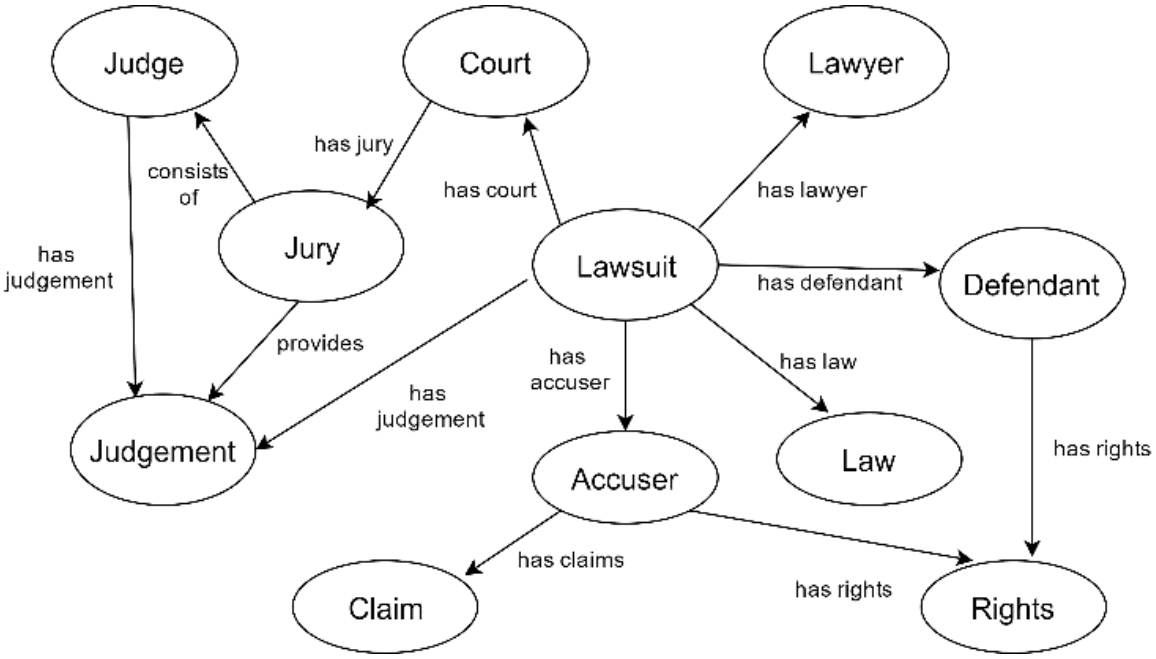


Fig. 2. Ontology sub-section used for the population

$$X = indexOf(y, S) \quad (1)$$

1) *Membership by Distance Model (M_1):* In this model, the candidate vectors for the ontology are generated from the instance corpus based on the minimum distance to the representative class vector derived in Section III-C. Given an instance i which has the vector embedding X_i, Equation 2 describes which class the particular instance belongs to.

$$C_{M1} = \left\{ indexOf(c_j, C) \left| \underset{c_j \in C}{\operatorname{argmax}} \{ distance(X_i, c_j) \} \right. \right\} \quad (2)$$

Here, the set C denotes the set of representative class vectors. C_{M1} is the selected class index of the instance i out of class set C. distance (X_i, c_j) is a function which provides domain specific semantic similarity between the given instances. In measuring the semantic similarity between the given instance and derived ontology class vector, we could encounter a situation where the derived ontology class vector may not be in the vector space model. In such a situation, semantic similarity was taken by identifying the closest vector available in the vector space to the derived ontology class vector and then taking domain specific semantic similarity between the identified vector and the instance vector. Here, the closet vector to the given ontology class vector was found based on the cousin similarity.

2) Membership by dissimilar exclusion model (M_2): In this model, we use word2vec based dissimilar exclusion method in identifying the membership of a particular instance to a given class. This is a utilization of an internal method of word2vec where given a set of members, it would return the member that should be removed from the set-in order to increase the set cohesion. For example, given the set of instances: *breakfast*, *cereal*, *dinner* and *lunch*, the word2vec dissimilar exclusion method would identify the instance *cereal* as the item that should be removed from the set to increase the set cohesion. We define this method as shown in Equation 3, where *S* is the set provided and *e* is the member selected to be excluded.

$$e = \text{Exclusion}(S) \quad (3)$$

Here the Exclusion(S) is defined as below. For a given n number of words, we obtain word embeddings of them using word2vec. Let W_v denote the word vectors of n words.

$$W_v = [v_1 \quad v_2 \quad v_3 \quad \dots \quad v_n] \quad (4)$$

Now we take for each word vector, the average distance from the rest of the word vectors as per the Equation 5. The i^{th} word will have zero distance from itself so there is no need to explicitly remove the i^{th} element from the sum.

$$d_i = \frac{\sum_{j=1}^{n} \text{Distance}(v_i, v_j)}{n-1} \quad (5)$$

Here, d_i denotes the average distance from the rest of the other word vectors for the word vector i. Distance (v_i, v_j) function performs the distance calculation based on domain specific semantic similarity measure as per [13].

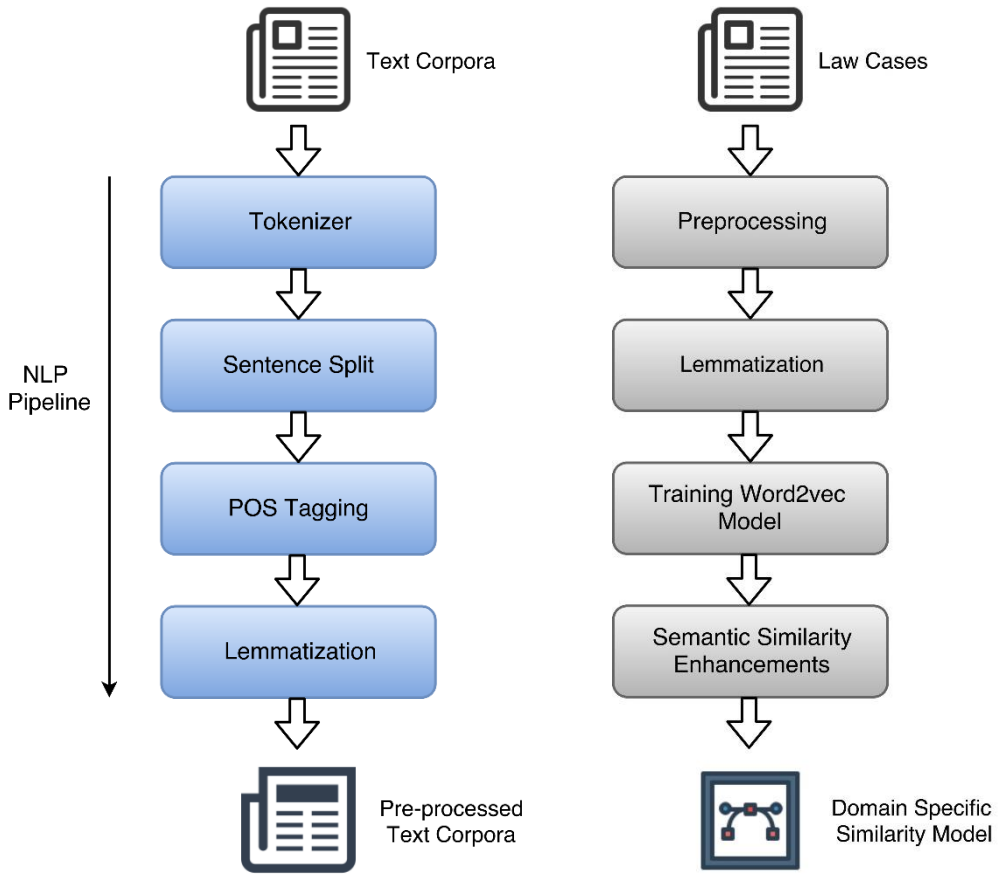
Upon defining the d_i as per Equation 5, we then define *D* as per Equation 6.

$$D = [d_1 \quad d_2 \quad d_3 \quad \dots \quad d_n] \quad (6)$$

Finally, we identify e, the member selected to be excluded as Equation 7.

$$e = \left\{ indexOf(d_j, D) \left| \underset{d_j \in D}{\operatorname{argmin}} \{ d_j \} \right. \right\} \quad (7)$$

Here, S_j is the seed set of class *j*. If the value $E_{i,j}$ gets evaluated to *TRUE* we declare that instance *i* should belong to class *j* under model M_2. We used the Equation 10 to decide whether the instance *i* should belong to class *j*.



(a) NLP Pipeline for Text Pre-processing

(b) High-level overview of building domain specific semantic similarity model pipeline

Fig. 3. Graphical illustrations of process pipelines

$$E_{i,j} = \begin{cases} 1 & if\ e \in S_j\ where\ e = \text{Exclusion}(S_j \cup X_i) \\ 0 & Otherwise \end{cases} \quad (8)$$

When using the aforementioned method in identifying the membership of an instance, there is a possibility of getting more than one class for a given instance as a possible parent class. Hence;

$$C_{M2} = \{k | 0 < k \leq N\} \quad (9)$$

Here in Equation 9, C_{M2} is the set of classes for a given instance i and N is the total number of classes we have in the ontology.

3) Set Expansion Based Model (M_3): For the purpose of set expansion based model, we selected the algorithm presented in [5] which was built on the earlier algorithm described in [15]. The rationale behind this selection is the fact that as per [5], WordNet [32] based linguistic processes are reliable due to the fact that the WordNet lexicon was built on the knowledge of expert linguists.

In this model, the idea is to increase the ontology class instances based on a WordNet hierarchy-based expansion. Simply put, it discovers the WordNet *synsets* pertaining to the seed words and proceeds up the hierarchy to find the minimum common ancestors for each of the senses of the words. Next the most common word sense is selected by majority. The relevant rooted tree is extracted and the gazetteer list of that rooted *synset* tree is created. The gazetteer list is subjected to set subtraction of the original seed set. The set intersection of the remaining set with the candidate word set is declared to be the word set assigned to the given class. However, it should be noted that as we showed in model M_2, after running the set expansion algorithm, one candidate instance may be tentatively assigned to more than one class. Fig. 4 illustrates the flow

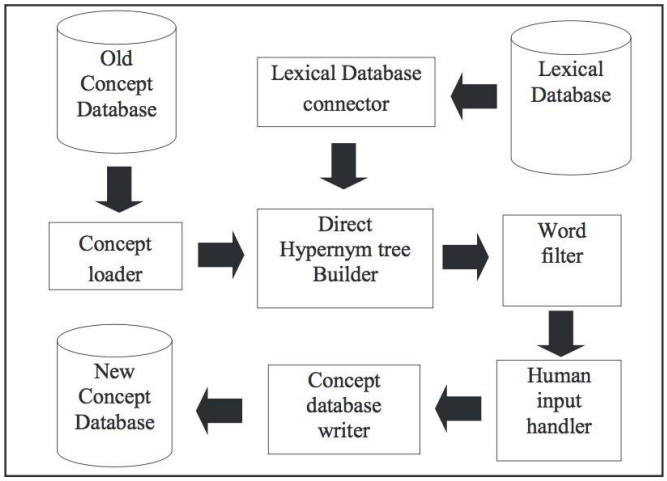


Fig. 4. Flow diagram for the simplified architecture for concept expanding using WordNet. (synonym) [5]

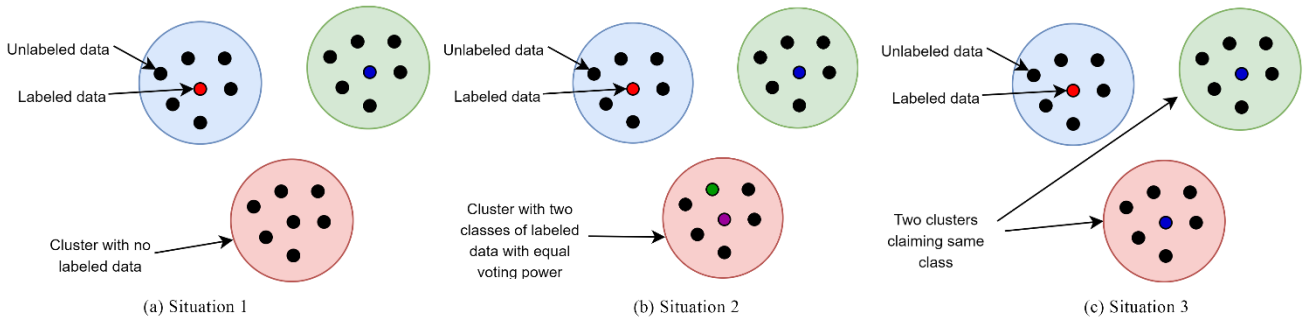


Fig. 5 Three situations in semi-supervised k-means clustering based model

for the simplified architecture of the concept expanding using WordNet as per the algorithm we used [5].

4) Semi-Supervised K-Means Clustering Based Model (M_4): Out of the models proposed in this study so far, this model is the first semi-supervised model. First, the seed instances are put together with the unlabelled data from instance corpus. Let Nlabeled be the number of labelled (seed) instances and Nunlabeled be the total number of unlabelled instances. Thus, by mixing up the labelled and unlabelled data, we get a total of Nlabeled + Nunlabeled number of instances. Next, all the instances are used to run the k-means algorithm where k is selected to be the same as the number of classes in the ontology.

Once the k-means clustering is finished, primary class cluster assignment for cluster L is done by voting of seed instances according to Equation 10, where C is the set of ontology classes, c_j is the j^{th} class from C, y_i is the i^{th} instance from L, and d_i is defined according to Equation 11.

$$C_l = \left\{ indexOf(c_j, C) \,\middle|\, \underset{c_j \in C}{\operatorname{argmax}} \left\{ \sum_{y_i \in L} d_i \right\} \right\} \tag{10}$$

$$d_i = \begin{cases} 1 & \text{if } y_i \in c_j \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

At this point, it should be noted that there can be three situations where it is possible to not get a c_l value assigned to some class L by Equation 10 without ambiguity: (1) L not having any seed instances to vote. (2) L has multiple seed instances but the majority voting ended in a tie. (3) Two (or more) clusters, claim the same class. These three situations are illustrated in Fig. 5. To solve these problems we defined Equation 12, which selects the unassigned class that is closest to an unassigned cluster. Here, an unassigned cluster L' is considered. C' is the set of representative class vectors of unassigned classes. $C_{l'}$ is the selected class index of the cluster L'.

$$C_{l'} = \left\{ indexOf(c_j, C) \,\middle|\, \underset{c_j \in C'}{\operatorname{argmax}} \left\{ \sum_{x_i \in L'} \left\{ \frac{X_i . c_j}{|X_i||c_j|} \right\} \right\} \right\} \tag{12}$$

The first problem to be solved is the problem of L having multiple seed instances, but the majority voting ending in a tie. In this case the C' of Equation 12 is limited to the set intersection of tied classes and unassigned classes. Next, the problem of Two (or more) clusters, claiming the same class is solved. In this case C' of Equation 12 is limited to the contested class. These steps are repeated until there is an iteration where there are no new assignments. Finally, all the remaining unassigned classes are put in C' and Equation 12 is executed repetitively with tie breaking, done with precedence until all the clusters are uniquely assigned to some class.

5) Semi-Supervised Hierarchical Clustering Based Model (M_5): The next model we used is a semi-supervised method based on hierarchical clustering. Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. We built a model which creates such hierarchy of clusters using the word embeddings taken from the word2vec model of the entire corpus similar to the process in Section III-F4. In this model, we extracted the slice of hierarchical clusters such that the number of clusters in the slice is equal to the number of classes in the sub-ontology. Next, the cluster-class assignment was done similar to the process in Section III-F4. Fig. 6 symbolizes this process in a nutshell.

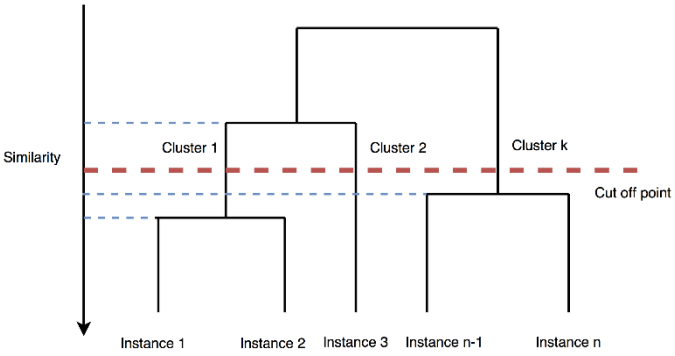


Fig. 6. Hierarchical clustering based approach

G. Model Accuracy Measure

After building the aforementioned models, we evaluated the accuracy of each model. As each model outputs an unordered set of suggested words, we sorted them using the Neural Network trained according to the methodology proposed in [13]. Upon completing the sorting, we applied a threshold to select the best candidates. Finally, we measured each model's accuracy as below. For this task, we involved domain experts and knowledge engineers. For a given model M_i in the context of class j:

$$\text{Precision}_{M_{i,j}} = \frac{W_{M_{i,j}} \cap W_j}{W_{M_{i,j}}} \tag{13}$$

$$\text{Recall}_{M_{i,j}} = \frac{W_{M_{i,j}} \cap W_j}{W_j} \tag{14}$$

Here, $W_{M_{i,j}}$ denotes words by the model M_i and W_j denotes the set of the words proposed by domain experts in to be the golden standard for class j. The model precision and recall of M_i was calculated by averaging the class values for precision and recall for those models.

$$F1_{M_i} = 2.\frac{\text{Precision}_{M_i} . \text{Recall}_{M_i}}{\text{Precision}_{M_i} + \text{Recall}_{M_i}} \quad (15)$$

H. Ensemble Model

Next, we came up with an ensemble model based on the models identified earlier. In the task of creating the ensemble model, we allocated a candidate weight for each model based on each model's *F1* measure as calculated in the previous step.

Let M_i be a model out of the models and let $F1_i$ be the F1 measure of model M_i. Hence, with the models in consideration, weight of the model W_i is calculated as shown in Equation 16, where p is the total number of models.

$$W_i = \frac{F1_i}{\sum_{i=1}^{p} F1_i} \quad (16)$$

As identified above, upon calculating the weight of each model, we created the ensemble model as shown in Equation 17. Given an unlabelled instance Y, let $M_{ensemble}$ be a p x n matrix where n denotes the number of classes in the ontology and p denotes the number of basic models. Each column of the matrix corresponds to a class in the ontology and each row corresponds to a model while each $m_{i,j}$ is derived from Equation 18.

$$M_{ensemble} = \begin{bmatrix} m_{1,1} & \cdots & m_{1,n} \\ \vdots & \ddots & \vdots \\ m_{p,1} & \cdots & m_{p,n} \end{bmatrix} \quad (17)$$

$$m_{i,j} = \begin{cases} 1 & \text{if } Y \in M_i \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Let $M_{weights}$ be the p length vector which defines the weights of each model calculated by Equation 16.

$$M_{weights} = [w_1 \quad w_2 \quad w_3 \quad \ldots \quad w_p] \quad (19)$$

Then we calculate the total score vector for the instance Y by,

$$S = M_{weights} . M_{ensemble} \quad (20)$$

Here, S is the score vector of size n where element i in the vector denotes the total score for instance Y for the membership in Class C_j. Next, we selected the class with the highest membership score as the parent class of instance Y. It is illustrated in Equation 21.

$$C_{M_{ensemble}} = \left\{ indexOf(S_{C_i}, S) \;\middle|\; \underset{S_{C_i} \in S}{\text{argmax}} \{S_{C_i}\} \right\} \quad (21)$$

With that, we get the final class of the instance Y. Hence, we populate that selected class with the instance Y.

IV. Results

In this section, we illustrate the results we obtained through our proposed methodology for semi supervised instance population of an ontology using word vector embeddings as the basis. We intend to illustrate and compare the results we obtained with domain specific semantic similarity incorporated and without incorporating it. It should be noted that domain specific semantic similarity was incorporated only in the models, membership by distance model(M1) and membership by dissimilar exclusion model(M2).

In testing our ensemble model, we used another instance corpus. In this corpus, we subdivided in the order of 70%, 20%, and 10% as the training set, validation set, and test set respectively. Training set was used in training the models individually. Validation set was used to fine tune the models. Finally, testing test was used in verifying the accuracy of the models. We have reported our findings below in the Table 1, where we compare the individual models: membership by distance model (M_1), membership by dissimilar exclusion model (M_2), set expansion based model (M_3), k-means clustering based model (M_4), hierarchical clustering based model (M_5) and the ensemble model as a whole. In Fig. 7, we compare the precision, recall and F1 of each of the candidate models along with the ensemble model with the domain specific semantic similarity. In Fig. 8, we compare the performance of membership by distance model (M1) and membership by dissimilar exclusion model (M2) with and without domain specific semantic similarity.

TABLE I

COMPARISON OF PERFORMANCE OF MODELS WITHOUT DOMAIN SPECIFIC SEMANTIC SIMILARITY

	Precision	**Recall**	**F1**
M1	0.08	0.22	0.12
M2	0.15	0.36	0.21
M3	0.24	0.30	0.26
M4	0.07	0.20	0.10
M5	0.06	0.23	0.10
Mensembe	**0.51**	**0.63**	**0.56**

In defining the ensemble model, Equation 22 defines the calculated weights of each model in the order of models *M1* to *M5* without domain specific semantic similarity incorporated. Equation 23 defines the calculated weights of each model in the order of models M1 to M5 with domain specific semantic similarity incorporated. These calculations are based on the above-mentioned training set.

$$M_{weights} = [0.15 \quad 0.27 \quad 0.33 \quad 0.13 \quad 0.12] \quad (22)$$

TABLE II

COMPARISON OF PERFORMANCE OF MODELS WITH DOMAIN SPECIFIC SEMANTIC SIMILARITY

	Precision	**Recall**	**F1**
M1	0.17	0.33	0.22
M2	0.27	0.45	0.34
M3	0.24	0.30	0.26
M4	0.07	0.20	0.10
M5	0.06	0.23	0.10
Mensembe	**0.65**	**0.70**	**0.67**

$$M_{weights} = [0.22 \quad 0.33 \quad 0.25 \quad 0.10 \quad 0.10] \quad (23)$$

It can be seen that, domain specific semantic similarity measures have improved the performance of membership by distance model (M1) and membership by dissimilar exclusion model (M2) by 10% and 13% respectively. Also with that performance change, the ensemble model has an improvement of 11% compared to the ensemble model we obtained without domain specific semantic similarity measures incorporated.

As can be seen on Table II, our ensemble model's F1 has been improved by 33%, compared to the best of the candidate models with semantic similarity measures being used. Hence, from the results obtained, as a proof of concept, we can demonstrate that word embeddings can be used effectively in semi-supervised ontology population.

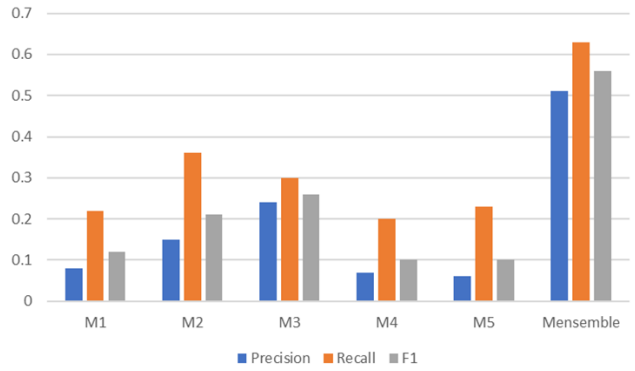


Fig. 7. Comparison of precision, recall and F1 of the models with domain specific semantic similarity

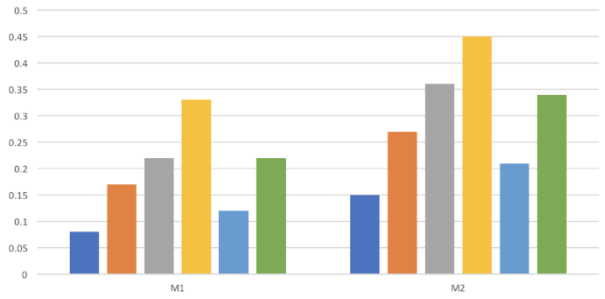


Fig. 8. Comparison of precision, recall and F1 of the models M1 and M2 with and without domain specific semantic similarity

V. Conclusion and Future Works

The methods and experiments presented in this journal paper on semi-supervised ontology instance population are extensions of our conference paper [42]. The methods and experiments on embedding semantic similarity measures to ontology assisted models in outperforming known benchmarks are implementations presented exclusively on this journal paper.

Through this work we demonstrated the use of word embeddings on semi-supervised ontology population. We mainly focused on semi-supervised population which basically falls between the supervised population and unsupervised population. The main motive behind making the process semi- supervised is to reduce the level of manual interventions in ontology populations while maintaining a considerable amount of accuracy. As shown in the results, our ensemble model outperforms the five individual models in populating the selected legal ontology. The findings in this study is mainly important in two ways as mentioned below.

Firstly, an important part of the ontology engineering cycle is the ability to keep a handcrafted ontology up to date. Through the semi-supervised ontology population, we can reduce the hassle involved in manual intervention to keep the ontology updated.

Secondly, there is novelty in the methodology proposed in our study. We proved that, since word embeddings map words or phrases from the vocabulary to vectors of real numbers based on the semantic context, a methodology based upon it can yield more sophisticated results when it comes to context sensitive tasks like ontology population. This indeed is a step up from the traditional information extraction based ontology population and maintenance processes, towards new horizons.

We can improve the methodology proposed, to yield better accuracy performances. For an example, we only considered the single word instances in populating the ontology using the defined models. However, in some of the scenarios, phrases also could be instances of ontology classes. Hence, it is important to convert phases to vectors and use them in the methodology as well. Also, as illustrated with models M4 and M5, we can perform more sophisticated semi-supervised ontology populations based on the concept of this study with more improvements. We keep them to be the future works of this study.

VI. Acknowledgement

The authors would also like to thank Menuka Warushavithana, Thejan Rupasinghe, Gathika Rathnayaka and Viraj Salaka from Department of Computer Science and Engineering of University of Moratuwa, Sri Lanka, for the immense assistance they provided in preparing this work.

References

[1] N. Guarino, "Formal Ontology and Information Systems," *Proceedings of FOIS'98, Trento, Italy*, 1998.

[2] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition, 5(2):199-220*, 1993.

[3] X.-Q. Yang, N. Sun, T.-L. Sun, X.-Y. Cao and X.-J. Zheng, "The application of latent semantic indexing and ontology in text classification," *International Journal of Innovative Computing, Information and Control*, vol. 5, pp. 4491-4499, 2009.

[4] N. H. N. D. de Silva, "SAFS3 algorithm: Frequency statistic and semantic similarity based semantic classification use case," *Advances in ICT for Emerging Regions (ICTer), 2015 Fifteenth International Conference on*, pp. 77-83, 2015.

[5] N. H. N. D. De Silva, A. S. Perera and M. K. D. T. Maldeniya, "Semi-supervised algorithm for concept ontology based word set expansion," *Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on*, pp. 125-131, 2013.

[6] J. Huang, F. Gutierrez, H. J. Strachan, D. Dou, W. Huang, B. Smith, J. A. Blake, K. Eilbeck, D. A. Natale, Y. Lin and others, "OmniSearch: a semantic search system based on the Ontology for MIcroRNA Target (OMIT) for microRNA-target gene interaction data," *Journal of biomedical semantics*, vol. 7, p. 1, 2016.

[7] J. Huang, K. Eilbeck, B. Smith, J. A. Blake, D. Dou, W. Huang, D. A. Natale, A. Ruttenberg, J. Huan, M. T. Zimmermann and others, "The development of non-coding RNA ontology," *International journal of data mining and bioinformatics*, vol. 15, pp. 214-232, 2016.

[8] D. C. Wimalasuriya and D. Dou, "Ontology-based information extraction: An introduction and a survey of current approaches," *Journal of Information Science*, 2010.

[9] N. de Silva, D. Dou, and J. Huang, "Discovering inconsistencies in pubmed abstracts through ontology-based information extraction," in Proceedings of the 8th ACM International

Conference on Bioinformatics, Computational Biology, and Health Informatics. ACM, 2017, pp. 362–371.

[10] R. G. Carla Fariaa, Ivo Serrab, "A domain-independent process for automatic ontology population from text," *Science of Computer Programming*, 2014.

[11] J. R. Rene Witte, Ninus Khamis, "Flexible Ontology Population from Text: The OwlExporter."

[12] V. Jayawardana, D. Lakmal, N. de Silva, A. S. Perera, K. Sugathadasa, and B. Ayesha, "Deriving a Representative Vector for Ontology Classes with Instance Word Vector Embeddings," *arXiv Prepr. arXiv1706.02909*, 2017.

[13] K. Sugathadasa, B. Ayesha, N. de Silva, A. S. Perera, V. Jayawardana, D. Lakmal, and M. Perera, "Synergistic union of word2vec and lexicon for domain specific semantic similarity," arXiv preprint arXiv:1706.01967, 2017.

[14] I. Wijesiri, M. Gallage, B. Gunathilaka, M. Lakjeewa, D. C. Wimalasuriya, G. Dias, R. Paranavithana and N. De Silva, "Building a WordNet for Sinhala," in *7th Global Wordnet Conference*, 2014, p. 100.

[15] N. de Silva, C. Fernando, M. Maldeniya, D. N. C. Wijeratne, A. S. Perera, and B. Goertzel, "SeMap-mapping dependency relationships into semantic frame relationships," in *17th ERU Research Symposium*, 2011, vol. 17.

[16] N. de Silva, D. Maldeniya, and C. Wijeratne, "Subject Specific Stream Classification Preprocessing Algorithm for Twitter Data Stream," *arXiv Prepr. arXiv1705.09995*, 2017.

[17] L. De Silva and L. Jayaratne, "Semi-automatic extraction and modeling of ontologies using Wikipedia XML Corpus," *Applications of Digital Information and Web Technologies*, 2009.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv Prepr. arXiv1301.3781*, 2013.

[20] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation.," in *EMNLP*, 2014, vol. 14, pp. 1532–1543.

[21] R. Das, M. Zaheer, and C. Dyer, "Gaussian LDA for Topic Models with Word Embeddings.," in *ACL (1)*, 2015, pp. 795–804.

[22] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1555–1565, 2014.

[23] B. Xue, C. Fu, and Z. Shaobin, "Study on sentiment computing and classification of sina weibo with word2vec," *International Congress on Big Data (BigData Congress) 2014 IEEE*, pp. 358–363, 2014.

[24] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and svm perf," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1857–1863, 2015.

[25] H. Liu, "Sentiment Analysis of Citations Using Word2vec," *arXiv Prepr. arXiv1704.00177*, 2017.

[26] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and word2vec for text classification with semantic features," in *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on*, 2015, pp. 136–140.

[27] G. Wohlgenannt and F. Minic, "Using word2vec to Build a Simple Ontology Learning System." [Online]. Available: http://ceur-ws.org/Vol-1690/paper37.pdf. [Accessed: 30-May-2017].

[28] H. Prins, "Matching ontologies with distributed word embeddings."

[29] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*, pp. 855–864, 2016.

[30] Diana Maynard Adam Funk and W. Peters, "SPRAT: a tool for automatic semantic pattern-based ontology population," 2009.

[31] H. T. Ng and H. B. Lee, "Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach," in *34th annual meeting on Association for Computational Linguistics*, 1996, pp. 40–47.

[32] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: An on-line lexical database," *Intarenational Journal of Lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[33] Z. Wu and M. Palmer, "Verbs Semantics and Lexical Selection," in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, 1994, pp. 133–138.

[34] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proc of 10th International Conference on Research in Computational Linguistics, ROCLING'97*, 1997.

[35] G. Hirst, D. St-Onge, and others, "Lexical chains as representations of context for the detection and correction of malapropisms," *WordNet An Electronic Lexical database*, vol. 305, pp. 305–332, 1998.

[36] D. E. Oliver, Y. Shahar, and others, "Representation of change in controlled medical terminologies," *Artificial Intelligence in Medicine*, vol. 15, no. 1, pp. 53–76, 1999.

[37] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr., and T. M. Mitchell, "Coupled Semi-Supervised Learning for Information Extraction," *WSDM '10 Proceedings of the third ACM International Conference on Web search and data Mining*, pp. 101–110, 2010.

[38] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr., and T. M. Mitchell, "Toward an architecture for never-ending language learning," *Proceeding AAAI'10 Proc. Twenty-Fourth AAAI Conference on Artificial Intelligence*, pp. 1306–1313, 2010.

[39] R. S. Zhilin Yang William W. Cohen, "Revisiting Semi-Supervised Learning with Graph Embeddings," *Proceedings of International Conference on Machine Learning*, 2016.

[40] J. Liu, L. Qin, and H. Wang, "An ontology mapping method based on support vector machine," *Proceedings of 8th International Confeence on Ontoogy Matching-Volume 1111*, pp. 225–226, 2013.

[41] "{FindLaw} Cases and Codes." [Online]. Available: http://caselaw.findlaw.com/. [Accessed: 18-May-2017].

[42] V. Jayawardana, D. Lakmal, N. de Silva, A. S. Perera, K. Sugathadasa, and B. Ayesha, "Semi-supervised instance population of an ontology using word vector embedding," in *Advances in ICT for Emerging Regions (ICTer), 2017 Seventeenth International Conference on*, 2017, pp. 1–7.