NLP-Based Approach to Semantic Classification of Heterogeneous Transportation Asset Data Terminology

Tuyen Le, S.M.ASCE[1]; and H. David Jeong, A.M.ASCE[2]

Abstract: The inconsistency of data terminology has imposed big challenges on integrating transportation project data from distinct sources. Differences in meaning of data elements may lead to miscommunication between data senders and receivers. Semantic relations between terms in digital dictionaries, such as ontologies, can enable the semantics of a data element to be transparent and unambiguous to computer systems. However, because of the lack of effective automated methods, identifying these relations is labor intensive and time consuming. This paper presents a novel integrated methodology that leverages multiple computational techniques to extract heterogeneous American-English data terms used in different highway agencies and their semantic relations from design manuals and other technical specifications. The proposed method implements natural language processing (NLP) to detect data elements from text documents and uses machine learning to determine the semantic relatedness among terms using their occurrence statistics in a corpus. The study also consists of developing an algorithm that classifies semantically related terms into three different lexical groups including synonymy, hyponymy, and meronymy. The key merit in this technique is that the detection of semantic relations uses only linguistic information in texts and does not depend on other existing hand-coded semantic resources. A case study was undertaken that implemented the proposed method on a 16-million-word corpus of roadway design manuals to extract and classify roadway data items. The developed classifier was evaluated using a human-encoded test set, and the results show an overall performance of 92.76% in precision and 81.02% recall. **DOI: 10.1061/(ASCE)CP.1943-5487.0000701.**

Author keywords: Heterogeneous data terminology; Data sharing; Semantic interoperability; Semantic relation; Natural language processing; Vector space model; Transportation data.

Introduction

The implementation of advanced technologies such as three-dimensional (3D) modeling, geographic information systems (GISs), mobile devices, or light detection and ranging (LiDAR) throughout the lifecycle of a transportation asset has enabled data to be increasingly available in digital format. Because of the fragmented nature of the transportation industry, lifecycle data are generated individually by project partners and are archived in their own repositories (Harrison et al. 2016). The efficiency of data sharing and integration is crucial to enhance data reusability which will translate into reduced data re-creation, enhanced productivity, and better decision making. Addressing the interoperability issue has been widely recognized as a pressing need to allow for computer-to-computer data exchange and seamless integration of heterogeneous data from multiple sources (Karimi et al. 2003; Gallaher et al. 2004; Bittner et al. 2005). The transportation sector, however, has not yet successfully facilitated a high degree of interoperability (Lefler 2014). To reuse digital data, much laborious work is required for finding, verifying, and transforming facility and project information from a certain format to one another (Gallaher et al. 2004).

Semantic interoperability is the highest level of interoperability that is concerned with the issue, whereby two computer systems may not share a common understanding of the same data item (Heiler 1995). In the fragmented civil infrastructure domain, names of things might vary across data sources. Polysemy and synonymy are two major linguistic obstacles to the semantic integration of a multitude of data sources (Noy 2004). Polysemy refers to cases in which a unique data term has distinct meanings in different contexts. The difference in meaning is a result of the diversity and temporary nature of definitions and the variation in data collection methods (Walton et al. 2015). For example, 'rail' can mean a transportation mode or a barrier structure. Synonymy, in contrast, is associated with the disparity of names for the same data across systems. For instance, the data element of roadway type is named 'functional system' in the Highway Performance Monitoring System (HPMS), but 'functional class' in the Highway Safety Information System (HSIS). Data integration in such a heterogeneous environment is highly problematic (Karimi et al. 2003). Polysemy may lead to a wrong match of two semantically different data items; and synonymy can cause a failure of aggregating similar elements. Explicitly specifying the semantic equivalence or relatedness between data terminologies becomes critical to proper integration of disparate data (Ouksel and Sheth 1999).

Previous studies on semantic similarity and relatedness between data items lie in the development of data libraries, taxonomies, and ontologies. A semantic resource specializes the meaning of terms through their lexical relations with each of other. Examples in this area include the civil engineering thesaurus (Abuzir and Abuzir 2002), the e-Cognos ontology (Wetherill et al. 2003), and the buildingSMART data dictionary (buildingSMART 2016). As shown in the literature review, their coverages are still limited, especially in the transportation sector, in spite of years of efforts because of the

[1]Ph.D. Candidate, Dept. of Civil, Construction and Environmental Engineering, Iowa State Univ., Ames, IA 50011. E-mail: ttle@iastate.edu

[2]Associate Professor, Dept. of Civil, Construction and Environmental Engineering, Iowa State Univ., Ames, IA 50011 (corresponding author). ORCID: https://orcid.org/0000-0003-4074-1869. E-mail: djeong@iastate.edu

Note. This manuscript was submitted on May 22, 2016; approved on April 7, 2017; published online on July 29, 2017. Discussion period open until December 29, 2017; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

reliance on conventional methods, which are labor intensive and time consuming. To develop a knowledge base, developers are required to manually determine important terms and their relations by interviewing domain experts or examining technical documents. The shortage of such semantic resources has become a bottleneck for semantic integration. There is a need for an automated data classification method that will allow digital dictionaries to be quickly constructed for specific needs and to keep up with the growth of terms (Mounce et al. 2010).

To fulfill that demand, this study aims to propose a novel linguistic approach for automatically classifying the semantic relations among heterogeneous data elements associated with a transportation asset. The study leverages natural language processing (NLP) to extract key data items and their meanings by analyzing the statistical data of context words in technical documents. This process generates a vector space in which each point represents the semantics of a data item. The research also includes a new integrated classification algorithm that utilizes syntactic rules, cluster analysis, and word embedding to categorize related elements into three different lexical groups that are similar-to (synonymy), is-a (hyponymy), and part-of (meronymy). To demonstrate the success of the proposed method, the framework was implemented on a corpus of roadway design manuals. A *Java* package and several data sets resulting from the study can be found at CeTermClassifier (2017).

Background

Natural Language Processing

NLP is a research area developing techniques that can be used to analyze and derive valuable information from natural languages like text and speech. Some of the major applications of NLP include language translation, information extraction, and opinion mining (Cambria and White 2014). These applications are embodied by a rich set of NLP techniques ranging from syntactic processing such as tokenization (breaking a sentence into individual tokens) (Webster and Kit 1992; Zhao and Kit 2011), part-of-speech (POS) tagging (assigning tags, e.g., adjective, noun, and verb, to each token of a sentence) (Toutanova et al. 2003; Cunningham et al. 2002), and dependency parser (identifying relationships between linguistic units) (Chen and Manning 2014), to the semantic level, for instance word sense disambiguation (Lesk 1986; Yarowsky 1995; Navigli 2009). NLP methods can be classified into two main groups: (1) rule-based and (2) machine learning (ML)–based methods. Rule-based systems, which rely solely on hand-coded syntax rules, are not able to fully cover all human rules (Marcus 1995); their performance, therefore, is relatively low. The ML-based approach is independent of languages and linguistic grammars (Costa-Jussa et al. 2012) because patterns can be quickly learned from even unannotated training examples. Thanks to its impressive out-performance, NLP research is shifting to statistical ML-based methods (Cambria and White 2014).

Vector Representation of Word Semantics

Measuring semantic similarity, which is an important NLP-related research topic, aims at determining how much two linguistic units (e.g., words, phrases, sentences, concepts) are semantically alike. For example, a 'railway' might be more similar to a 'roadway' than to a 'train.' The state-of-the-art methodology for this task can be divided into two categories that are (1) thesaurus-based methods and (2) vector space models (VSM) (also known as word embedding) (Harispe et al. 2013). The former approach relies on a hand-coded digital dictionary [e.g., WordNet (Princeton University 2017)] which formally structures terms in a network of semantic relations. In this method, the semantic similarity between a given pair of words can be measured based on the distance between them in the hierarchical structure. The method is an ideal solution if digital dictionaries are available. However, digital dictionaries are typically handcrafted; they are, therefore, not available to many domains (Kolb 2008). The latter technique assesses the meaning of words or phrases by analyzing their occurrence frequency in natural language text documents. VSM outperforms the dictionary-based method, especially in terms of time saving, because a semantic model can be automatically obtained from a text corpus, and corpus collecting is much easier than manually constructing a digital dictionary (Turney and Pantel 2010).

VSM estimates semantic similarity based on the distributional model, which represents the meaning of a word through its context (co-occurring words) in a corpus (Erk 2012). The distributional model stands on the distributional hypothesis that states that two similar terms tend to occur in the same context (Harris 1954). The output of this approach is a vector space, in which each numeric vector represents a word in the vocabulary. The similarity between semantic units in this model can be represented by the Euclidean distance between the corresponding points (Erk 2012).

The conventional method to construct a VSM is to use the word-context matrix, which shows how frequently a word appears in the context of another word in a given text corpus. These raw data of frequencies are used to estimate the co-occurrence probabilities. This statistical process results in a matrix in which each row is a vector representation. Pointwise mutual information (PMI) (Church and Hanks 1990) or its variant, positive PMI (PPMI) is a popular method to calculate co-occurrence probabilities. A more advanced approach uses machine learning to train representation vectors. The two leading state-of-the-art ML-based word embedding techniques are named Word2Vec and Glove. The Word2Vec model (Mikolov et al. 2013), which is a neural network model, learns vector representation of words from their surrounding words. Mikolov et al. (2013) proposed two opposite network architectures, including continuous bag-of-words (CBOW) and skip-gram. CBOW predicts a word given a set of context words, whereas skip-gram aims to predict the context of a given word. The training objective of both models is to minimize the overall prediction error. Glove or Global Vectors (Pennington et al. 2014) trains on the global co-occurrence matrix with the objective that the probability of co-occurrence between two words equals the dot product of their vector representations. There are conflicting recommendations on the wining model in the literature. The authors of Glove argue that their model outperforms Word2Vec. However, a number of independent benchmarking experiments provide an opposite suggestion. For example, a comparative study by Levy et al. (2015) on the accuracy in various tasks and golden standards reveals that the skip-gram in Word2Vec is superior to Glove in most of the experiments, especially on similarity evaluation. The best precision of skip-gram is 0.793, whereas Glove achieves the highest score of 0.725. The outperformance of Mikolov's models on the similarity task is confirmed in another benchmarking study (Hill et al. 2015), in which this model is also the winner.

The VSM approach has been progressively implemented in recent NLP-related studies in the construction industry. Yalcinkaya and Singh (2015) utilized VSM to extract principle research topics related to BIM from a corpus of nearly 1,000 paper abstracts. This approach was also used for information retrieval to search for text documents (Lv and El-Gohary 2015) or computer-aided design (CAD) documents (Hsu 2013). The increasing number of successful use cases in the construction industry has evidently

demonstrated that the VSM method can be successfully implemented to tackle the issue of semantic interoperability in sharing digital data across the lifecycle of a highway project.

Related Studies

A popular solution to semantic interoperability is to develop taxonomies, ontologies, or other forms of digital dictionaries that can provide machine-readable definitions of domain concepts. A plethora of such semantic resources have been developed for the highway industry. However, conventional development methods require significant human efforts on knowledge retrieval, and ontology construction and validation. The pioneer in this line of research is the e-Cognos ontology (Wetherill et al. 2003; Lima et al. 2005), which formulates the execution process of a construction project as an explicitly interactive network of the following principal concepts: actors, resources, products, processes and technical topics. The ontology developers of this project reviewed existing taxonomies and construction specific documents, and interacted with the end users to identify relevant concepts and their semantic relations. Industry experts were invited to validate e-Cognos' concept names and relations. Because the introduction of e-Cognos, plenty of other ontologies have been built for various aspects of a highway project, for instance, construction taxonomy (El-Diraby and Kashif 2005; El-Diraby et al. 2005), freight ontology (Seedah et al. 2015a), and the ontology of urban infrastructure products (Osman and Ei-Diraby 2006). These studies also relied on domain experts (El-Diraby and Kashif 2005; El-Diraby et al. 2005; Osman and Ei-Diraby 2006) or existing knowledge bases (Seedah et al. 2015a) to construct their semantic products. The limitations on time and resources of the traditional knowledge-based methodology have created a bottleneck in semantic interoperability. In addition, existing ontologies primarily focus on concept description and neglect the heterogeneity of concept names. Therefore, there is a need to develop a data-driven method that can automate the process of formulating domain concepts and also incorporate term diversity into ontologies.

Another line of research on semantic interoperability targets at the heterogeneity of concept names rather than concept description. A few frameworks to assist developers in precisely mapping data labels from heterogeneous sources have been introduced for various construction sectors. In the building sector, buildingSMART proposed a novel framework, namely the International Framework for Dictionaries (IFD) or ISO 12006-3 (ISO 2007) for developing a multilingual data schema in which each concept can have multiple names in different languages. With IFD, the identity of a concept is defined by a global unique ID (GUID) instead of its name; hence, an IFD-based exchange mechanism is able to avoid data mismatches owing to name inconsistency (Hezik 2008). The buildingSMART Data Dictionary (bSDD) (buildingSMART 2016) is the first digital library of building concepts organized in IFD format. Each concept in bSDD consists of a set of synonymy names not only in English but also in computer-coded languages (e.g., IFC) and in other human languages (e.g., French, Norwegian). Therefore, a complete bSDD would enable digital data regardless of languages to be sharable and unambiguously reusable. Yet, its size remains limited because the identification of these sets of synonyms is laborious and time intensive. In the transportation sector, there has been a shortage of research efforts on the heterogeneity of data names at the database level until recently. Seedah et al. (2015b) proposed a role-based classification schema (RBCS) to classify data in freight databases. RBCS defines nine distinct groups of roles that are time (year, month), place (city name, population), commodity (liquid, value), link (roadway name, width), mode (truck, rail), industry (company name, sales), event (accident, number of fatalities), human (officer, driver age), and unclassified. Seedah et al. (2015b) argue that once data elements across separate databases are categorized using this standard system, it becomes easier for practitioners to identify the semantic relatedness between items. However, even if RBCS is successfully applied to all freight databases, much more effort is still needed to further specify the relation type (e.g., synonym, functional-related) between two data elements in the same category.

In attempts to reduce laborious work on defining concepts, a few researchers have sought to propose semiautomated and automated methods for identifying semantic relations among technical terms. Abuzir and Abuzir (2002) developed the ThesWB system which utilizes hand-coded syntax patterns to detect lexical relations between civil engineering terms from HTML web pages. The performance of ThesWB was not reported, but it is not likely to be high because rule-based approaches are repeatedly criticized for not being able to capture all the variant ways to present relations among terms in natural language (Marcus 1995; Navigli and Velardi 2010). Rezgui (2007) suggested a more sophisticated approach that is based on the statistics of word occurrence in domain text documents rather than predefined rules. This method implements term frequency-inverse document frequency (TF-IDF) to evaluate the importance degree of a keyword to the examined domain. The method computes the relatedness between a given pair of important keywords using the metric clusters measure, which estimates the association based on the distance between them in the text. These potential relationships are then validated and categorized by domain experts. Because Rezgui's (2007) methodology detects relations between words occurring in the same sentence, equivalent terms which are used interchangeably could not be captured. In another study, Zhang and El-Gohary (2016) proposed a machine learning–based methodology for identifying the semantic relation between a new concept and the existing IFC entities. This algorithm was reported to achieve an average precision of nearly 90%. The algorithm identifies potentially related concepts based on the predefined lexical relations provided in WordNet (Princeton University 2017). Because WordNet is a generic lexicon that lacks concepts in many construction sectors, including the civil infrastructure, this algorithm would not be scalable well on matching terms in those domains.

As shown in the literature review, there are numerous research efforts in developing ontologies for the highway sector. However, existing ontologies are mainly hand-coded through the manual processes of knowledge acquisition and translation into a digital format. Relying on this traditional approach has created a bottleneck in facilitating semantic interoperability. A few efforts have been made to automate the process of constructing or extending existing semantic resources. The most rigorous methodology in the state of the art is the one developed by Zhang and El-Gohary (2016) that has a high level of accuracy. One limitation of this algorithm is the reliance on a semantic resource; therefore, it would not be well applicable to such domains as civil infrastructure and transportation that are beyond the vocabulary scope of existing lexical databases. Thus, it is essential to develop an automated method that can allow for fast development of domain lexicons and also reduce dependence on other existing semantic resources.

NLP-Based Methodology to Classification of Heterogeneous Data Terms

The goal of this research is to propose an NLP-based methodology that can automate the process of extracting diverse data elements

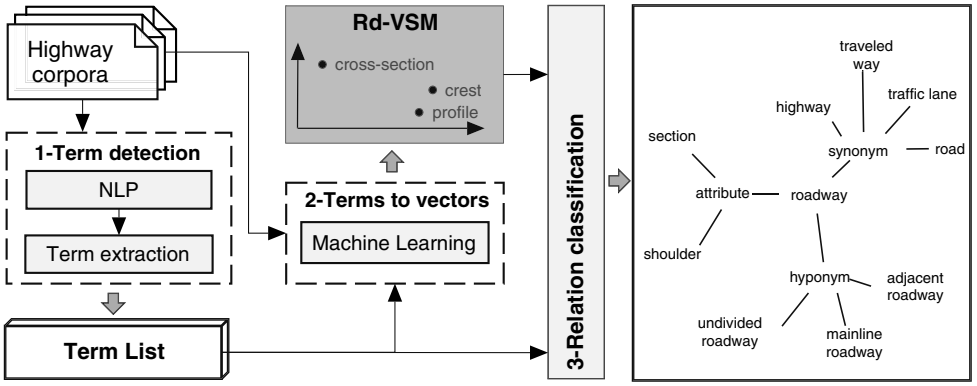

Fig. 1. Overview of the proposed methodology

and their semantic relations from American-English technical guideline documents. Fig. 1 shows that the proposed method consists of three major stages that are to (1) utilize NLP techniques to extract multiword data items from a domain text corpus; (2) implement machine learning to generate vector representation of the extracted terms; and (3) design an algorithm integrating various linguistic patterns, clustering, and semantic vectors to detect the semantic relation between a given pair of terms. The succeeding sections discuss these phases in detail.

Multiword Data Element Extraction

Technical documents such as design manuals, guidelines, and specifications are great sources of domain data elements that occur as technical terms. Linguists argue that a technical term is either a noun [e.g., road, annual average daily traffic (AADT)] or a noun phrase (NP) (e.g., right of way, sight distance) that frequently occurs in domain text documents (Justeson and Katz 1995). The meaning of a multiword term may not be directly interpreted from the meaning of its constituents; therefore, it must be treated as an individual word. As mentioned, a multiword term must be an NP; thus, NPs are good multiword term candidates. To detect this type of term, the corpus is first scanned to search for NPs, followed by assessing their importance to the domain. The process of extracting multiword terms is discussed in detail next.

NP Extraction

Fig. 2 illustrates how NPs are extracted from a natural language sentence based on the Apache OpenNLP (2017) library. This process includes the following steps:

1. Word tokenizing: In this step, the text corpus is broken down into individual units (also called tokens) (OpenNLP 2017). Tokenizing is to separate punctuation marks, for instance periods, commas, semicolons, and parentheses, from words. The tokenizer is capable of distinguishing between marks in acronyms (e.g., r.o.w., r/w) and punctuation symbols; these kinds of words will remain in the corpus.
2. POS tagging: The purpose of this step is to determine the part of speech tag (e.g., NN-noun, JJ-adjective, and VB-verb) for each unit of the tokenized corpus obtained from the previous step. A full set of POS tags can be found in the Penn Treebank (Marcus et al. 1993).
3. Noun phrase detection: This phase aims to collect NPs using syntactic rules. Table 1 presents the used patterns that are reformulated from the one suggested by Justeson and Katz (1995) for better human readability. The tagged corpus is thoroughly scanned to collect sequences matching those patterns. This study assumes that sequences of more than six words are not likely to be a technical term; therefore, they are automatically discarded. In addition, to reduce the discrimination between syntactic variants of the same term, the collected NPs need to be normalized. This study considers the following two types of syntactic variation:
   - Type 1: Plural forms, for example 'roadways' and 'roadway'. Stemming is a popular process to reduce words to their stems. Overstemming and understemming are two common errors. Overstemming refers to the removal of true suffixes (e.g., 'divided highway' → 'divide highway'); understemming occurs when predefined rules fail to handle irregular forms, for instance 'foot' and 'feet'. Despite the fact that none of the existing algorithms can completely eliminate these errors, they are good enough to not degrade the overall performance of NLP applications (Jivani 2011). This study implements the *PlingStemmer* (Suchanek et al. 2006), which stems an English noun to its singular form, to normalize plural nouns in the corpus. One advantage of this algorithm is the utilization of both syntactic rules and dictionaries.

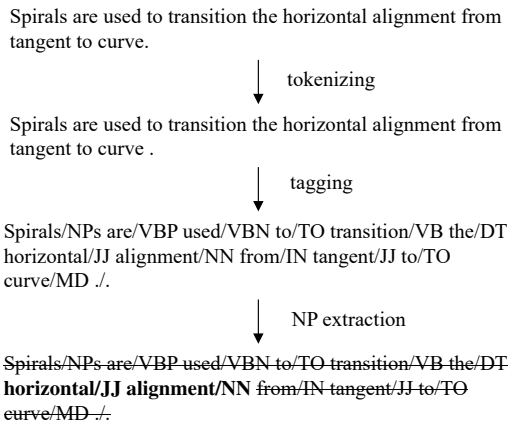

Fig. 2. Linguistic processing procedure to detect NPs

Table 1. Term Candidate Filters

Pattern	Examples
(Adj|N)*N	Road, roadway shoulder, vertical alignment
(Adj|N)*N Prep (Adj|N)*N	Right of way, type of roadway

Note: Prep = preposition; | and *, respectively, denote 'or', and 'zero or more'.

Dictionaries are to verify the outcomes from purely pattern-based stemming and allow for the inclusion of irregular plural nouns; therefore, stemming errors can be reduced. Furthermore, because only nouns are impacted, mis-stemming on such terms as 'divided highway' can be prevented.

- Type 2: Prepositional noun phrases, for example 'type of roadway' and 'roadway type'. To normalize this type of variation, the form with a preposition is converted into the non-preposition form by removing the preposition and reversing the order of the remaining portions. For example, 'type of roadway' will become 'roadway type.' However, blindly applying normalization will create unreal instances because not every prepositional NP is paraphrasable. 'Right of way' is one example of such non paraphrasable NPs. Therefore, this study implements paraphrasing for only those NPs whose reversed form also exists in the extracted list.

The instances obtained from the preceding process may include errors. To eliminate incorrectly extracted sequences, the following two discard criteria are used. First, a valid NP must not contain any minimal stop word. The minimal stop list consists of frequent words and phrases that carry obviously no meaning for a technical term, including determiners (e.g., another, any, particular), coordinating conjunctions (e.g., nor, or, and), comparative adjectives (e.g., largest, longest, best), and stop phrases (e.g., lack of, set of, kind of). The list is called a minimal stop list to distinguish it from the large stop list commonly used in NLP applications. The second constraint for filtering out bad NPs is occurrence frequency. This study assumes that instances that are not a randomly combined sequence appear at least twice in the corpus. Items that appear only once are eliminated. This hypothesis might be not applicable for a small corpus (e.g., 10,000 words) because the frequency of true NPs tends to be low.

NP Ranking and Term Selection

Multiword term definition varies among authors, and there is a lack of formal and widely accepted rules to determine if an NP is a multiword term (Frantzi et al. 2000). There are a number of methods for estimating termhood (the degree that a linguistic unit is a domain-technical concept), such as TF-IDF (Sparck Jones 1972; Salton and Buckley 1988), C-Value (Frantzi et al. 2000), and Termex (Sclano and Velardi 2007). Of these methods, Termex outperforms others on the Wikipedia corpus, and C-Value is the best on the GENIA medical corpus (Zhang et al. 2008). One notable observation from these studies is that C-Value is more suitable for term extraction from a domain corpus rather than a generic one. For this reason, C-Value has been used in various studies in the biomedical field, for instance works conducted by Ananiadou et al. (2000), Lossio-Ventura et al. (2013), and Nenadić et al. (2002). Because the methodology proposed in this paper aims to extract data elements from highway guidelines and manuals which are domain specific documents, C-Value would be the most suitable for the termhood determination task. The C-Value measure, as formulated in Eq. (1), suggests that the longer an NP is, the more likely that is a term; and the more frequently it appears in a domain corpus, the more likely it will be a domain term

$$\text{C-Value}(a) = \begin{cases} \log_2|a| \cdot f(a), & \text{if } a \text{ is not nested} \\ \log_2|a| \left[f(a) - \dfrac{1}{P(T_a)} \sum_{b \in T_a} f(b) \right], & \text{otherwise} \end{cases} \quad (1)$$

where a = candidate noun phrase; $|a|$ = length of noun phrase a; f = frequency of a in the corpus; T_a = set of extracted noun phrases that contain a; and $P(T_a)$ = size of T_a set.

The C-Value measure is used to compute termhood for every term candidate generated from the previous stage. This process results in a data set of terms along with their C-Value scores. These term candidates are ranked by C-Value.

To automatically remove candidates that are unlikely to be a domain term, a C-Value threshold can be used as an acceptance limit. However, choosing a proper absolute threshold is challenging because it typically depends on the corpus size. A high limit can help to significantly reduce bad candidates, but real terms that appear at the bottom owing to their low frequency will be excluded. Manual evaluation of the entire sorted list would avoid the removal of real terms with low C-Values, but it might be too laborious especially for large corpora. To minimize both laborious work and the number of true terms wrongly discarded, this study adopts a relative cut-off policy proposed by Lopes and Vieira (2015) that is based on the optimal trade-off point between a wrong discard of true domain terms and the wrong inclusion of irrelevant ones. The policy suggests that the bottom 85% of the ranked list should be discarded.

Data Element Vector Space Model

This phase aims at converting the vocabulary of a domain corpus into a vector space that presents the semantics of a term as a vector. This study uses the unsupervised Word2Vec model (Mikolov et al. 2013) to learn representation vectors. As discussed previously, Word2Vec and Glove are the two leading state-of-the-art word embedding techniques. Word2Vec is usually outperforms Glove, despite the fact that there is a lack of conclusive evidence in the literature for the superiority of one over the other. Because the objective of this research is not to propose an optimized embedding method, Word2Vec was arbitrarily selected for the vector representation learning task in the proposed classifier.

In the Word2Vec model, a training data point is corresponding to a target word and its context words in the corpus. Consider the sentence, "The minimum [bike lane width on a roadway with no curb and gutter] is 4 feet," with 'roadway' as the target word. Surrounding words are captured using a context window, indicated by the brackets, which limits how may words appear to the left and to the right of the target word. In the example, the context of the term 'roadway' with the window size of 5 will be {bike, lane, width, on, a, with, no, curb, and, gutter}. Any contextual word in the stop list (frequent words in English with little meaning, such as a, an, and the) will be neglected, and the context set becomes {bike, lane, width, curb, gutter}.

Before data collection, an additional step is needed to handle the issue related to multiword terms. Because document scanning is on a word-by-word basis, the tokenized and stemmed corpus resulted from the NP extraction phase must be adjusted so that multiword data elements can be treated as single words. To meet that requirement, white spaces within a multiword term are replaced by hyphens to connect its individual words into a single unit. For instance, 'vertical alignment' becomes 'vertical-alignment.'

This study trains vector representation using both CBOW and skip-gram network types of Word2Vec. Fig. 3 illustrates these learning networks, in which V and N respectively denote the size of the corpus vocabulary and the hidden layer. In CBOW, context words are at the input layer and target words are at the output layer, whereas skip-gram reverses the role of the data components. Word2Vec encodes a word as a *one-hot* vector in which only one element at the index of the word in the vocabulary is set to one, and all other items are zero. For example, the one-hot vector of

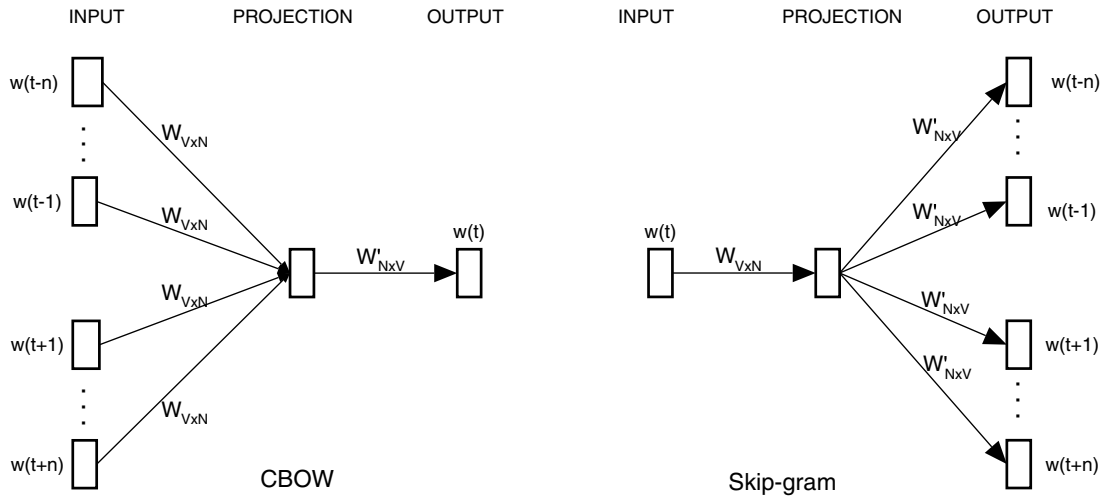

Fig. 3. Word2Vec neural network structures

the kth word in the vocabulary with the size of $\mathbf{V}$ will be $\{x_1 = 0, x_2 = 0, \ldots, x_k = 1, \ldots, x_V = 0\}$. The outcome of this machine learning process is a set of $\mathbf{N}$-dimensional representation vectors each of which is corresponding to a row in the learned parameter matrix, $\mathbf{W}_{\mathbf{N}\cdot\mathbf{V}}$. The similarity between a pair of vectors represents the similarity in context between their corresponding words and can be measured by the angle between word representation vectors [Eq. (2)] or the distance between word points [Eq. (3)]

$$\text{cosine\_similarity} = \frac{A \cdot B}{||A||\,||B||} \tag{2}$$

$$\text{dis\_similarity} = \sqrt{(xA_1 - xB_1)^2 + (xA_2 - xB_2)^2 + \cdots + (xA_n - xB_n)^2} \tag{3}$$

where n = vector dimension which is also the hidden layer size.

The learning model includes three major parameters—*frequency threshold*, *hidden layer size*, and *context window size* (Table 2). Frequency threshold is used in this phase to eliminate from the training data those input words that are unimportant to the domain. As discussed earlier in the NP extraction stage, low-frequency words are unlikely to be a technical term. Words with the occurrence below a threshold will be excluded from the input vocabulary. Radim (2014) suggests a frequency limit ranging from 0 to 100 depending on the corpus size, where 0 means to accept everything. Setting this parameter high can enhance the accuracy, but many true technical terms would be out of vocabulary. The second important parameter is layer size, which determines the number of nodes in the hidden layer. This parameter highly affects the training accuracy and processing time. A larger layer size is better in terms of accuracy, but this will be paid off by the running time. A reasonable figuration for this parameter is from tens to hundreds (Radim 2014). The final major parameter, context window size, decides how many context words to be considered. Google recommends a size of 10 for the skip-gram model. In the present experiments, these parameters are subject to be changed so that the best model can be achieved. The selection of an optimal parameter setting is discussed later in "Implementation and Performance Evaluation."

Table 2. Skip-Gram Model Parameters

Parameter	Value
Frequency threshold	0–100
Hidden layer size	100–500
Context window size	5–15

Semantic Relation Classification Algorithm

This section explains a designed classification algorithm for automated identification of semantic relations among data terms. This study focuses on the following three semantic relations: similar-to (synonymy), is-a (hyponymy), and part-of (meronymy). The similar-to relation refers to a pair of terms that share similar meanings. Because very few instances have exactly the same meaning, in this study, the similar-to category also includes near synonyms that can be used interchangeably to a certain extent (Inkpen and Hirst 2006). For example, two terms, 'highway' and 'street,' are equivalent in the context in which geometry is the only attribute considered. Another type to be detected is the is-a tag, which relates to concept–superconcept pairs, for instance 'highway-facility.' Finally, part-of is associated with instances in which a concept represents a component (or a property) of another concept, e.g., 'shoulder-road' and 'volume-traffic.'

Terms that relate to each other through one of the aforementioned semantic relations are expected to have a high similarity score. Thus, a collection of nearest terms generated by the vector space model is an excellent source of semantically related terms. To support automated detection of relation type, this study designs a classifying algorithm of which the pseudocode is shown in Algorithm 1. Given a pair of the target t and a near term n, the algorithm returns one of the following tags: similar-to, is-a, part-of, and nonrelated. First, a surfacing rule-based checking is performed. The rule in this study is that if the target word t (e.g., 'road') is the head noun of a near term n (e.g., 'public road'), a triple (n is-a t) is correspondingly harvested. In cases in which t (e.g., 'road') matches the modifier component of n (e.g., 'road facility'), the modifier is eliminated from n. Second, the algorithm detects the relation between pairs ($n - t$) by checking its occurrence in a syntactically related pair data set. The syntactic resource consists of is-a and part-of term pairs that are extracted from the input corpus using a minimally supervised training method (explained in the next section). The algorithm also considers *reverse* is-a (hypernym) and

reverse part-of (whole-of) when the input pair in reverse order exists in the syntactic resources. If the input pair does not belong to those categories, it is temporarily tagged as similar-to. Clustering is then applied on the temporary similar-to list after being sorted by similarity in descending order. Because similar terms tend to have a high similarity score, accepting only items occurring in the top c clusters helps to eliminate other nonrelated terms. Subsequent sections discuss in detail the collection of is-a and part-of instances and the clustering of similar-to items.

Algorithm 1. Semantic Relation Classification Algorithm

1: **Inputs**: term t, list of nearest terms N, list of part-of pairs P, list of is-a pairs I
2: **Outputs**: list of *Parts*, list of *Wholes*, list of *Hyponyms*, list of *Hypernyms*, list of *Synonyms*
3: **Procedure** Term classification procedure
4: **for all** $n \in N$ **do**
5: $x \leftarrow$ pair n: t
6: $h \leftarrow \text{headOf}(n); m \leftarrow \text{modifierOf}(n)$
7: **if** $h = t$ **then**
8: add x to *Hyponyms*
9: **else**
10: **if** $m = t$ **then**
11: $n \leftarrow h$
12: **if** $n{:}t \in P$ **then**
13: add x to *Parts*
14: **else if** $t{:}n \in P$ **then**
15: add x to *Wholes*
16: **else if** $n{:}t \in I$ **then**
17: add x to *Hyponyms*
18: **else if** $t{:}n \in I$ **then**
19: add x to *Hypernyms*
20: **else**
21: add x to *Synonyms*
22: Clusters $\leftarrow K$-mean(Synonyms)
23: Synonyms $\leftarrow$ instances in top c clusters of *Clusters*

Part-of and is-a Instance Extraction

Using syntactic patterns like those developed by Hearst (1992) is a popular method for automated detection of lexical relations. This method is straightforward in that instances can be quickly captured and can yield a high precision. However, a typical issue of using predefined rules is the low recall because generic patterns are usually ignored (Pantel and Pennacchiotti 2006). Generic patterns are those that are applicable to multiple types of relations. For instance, the pattern 'X of Y' can be found in both part-of (e.g., 'shoulder of roadway') and is-a (e.g., 'facility of highway'). In addition, existing patterns are usually induced from generic corpora and might not be well applicable for a domain corpus. This study adopts a widely used minimal-supervised technique proposed by Pantel and Pennacchiotti (2006) to learn reliable patterns for is-a and part-of relations from the highway corpus. The selection of this particular method results from its computational efficiency and recall improvement because more patterns can be discovered from domain-specific texts. The pattern learning is an iterative procedure of the following steps: (1) pattern induction, (2) pattern ranking/selection, and (3) instance extraction.

Pattern learning starts with extracting word sequences connecting the constituents of each pair instance for a certain relation (e.g., part-of). To initiate the first iteration, seed pairs, which are found by examining engineering glossaries from various state DOTs, are used. For example, with the seed 'median-roadway' of part-of, one extracted sequence is 'roadway without a median,' which correspondingly yields a pattern 'WHOLE without a PART.' Along with *good* chains, *bad* chains (e.g., 'of the roadway when median is') are also collected. Similar to the NP extraction task, a frequency threshold of 2 is used to reduce random sequences. The reliability of a pattern p in P patterns collected is measured as the average association with all instances in I using the following:

$$r_\pi(p) = \sum_{i \in I} \frac{\frac{pmi(i,p)}{\max_{\forall i \in I} pmi(i,p)} * r_l(i)}{I} \tag{4}$$

where $r_l(i)$ = instance reliability score which is defined later in Eq. (6). The reliability of initial seed pairs is set to 1. The association between instance i and pattern p, $pmi(i,p)$, is based on their occurrence frequencies as follows:

$$pmi(i,p) = \log \frac{|x,p,y|}{|x,*,y||*,p,*|} \tag{5}$$

where the asterisk represents a wildcard.

The patterns induced in Step 1 are ranked according to their reliability scores, and only the top k are accepted, in which k is set to 1 in the first iteration and increases by 1 over each iteration. The algorithm runs until k meets a given desired number of patterns, τ, which is 5 for all experiments in this study.

In Step 3, instances of related pairs are extracted from the corpus using those patterns accepted in Step 2. The reliability of an instance i is measured based on an equation analogous to the pattern reliability, as in

$$r_l(i) = \sum_{p \in P} \frac{\frac{pmi(i,p)}{\max pmi} * r_\pi(p)}{P} \tag{6}$$

Subsequent iterations will use the top m instances extracted for the pattern induction phase. In these experiments, $m = 100$. At the last iteration when τ patterns are induced, the extracted pairs are accepted as lexical-syntactic resources that will be used by the relation classifier.

Similar-to Instance Clustering

In this phase of the classification algorithm, the system implements cluster analysis on the temporary list to separate similar-to terms from other 'non-related' items. This study uses a k-means clustering algorithm (MacQueen 1967) to split the list into multiple clusters according to their similarity scores with the target word. The objective of k-means clustering, as illustrated in Eq. (7), is to minimize the sum of squared distances between words and the corresponding cluster centroid

$$\arg \min_C \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{7}$$

where μ_i = mean of points in the cluster C_i; and k = number of clusters. Because similar words tend to have a higher similarity score than other nonrelated words, items in the top clusters are more likely to be similar to the target word. Those terms beyond the top c clusters are unlikely to be a similar term; they are, thus, removed from the temporary similar-to list and are classified as non-related. Because increasing k would provide a better separation of near words, the value of k was chosen as high as the total similar-to candidates divided by 2 in these experiments.

Implementation and Performance Evaluation

This section presents an implementation case study on classifying roadway transportation data terms using the domain text. An empirical comparison between the proposed model and several baseline methods are also discussed.

Experiment Setup

Experiments were performed on a highway corpus composed of 48 engineering manuals and guidelines from 30 state DOTs. The content in a manual document in the civil engineering field is commonly presented in various formats such as plain text, tables, and equations. Because the structures of words in tables and equations are not yet supported by the state-of-the-art NLP techniques, they were removed from the text corpus. The removal may slightly reduce the corpus size, and accordingly affects the training data set. However, because sequences of words in tables and equations are not organized in the formal structure of a sentence, many unreal noun phrases would be captured when applying NP patterns on those features. The final plain text corpus consists of nearly 16 million words. This data set was utilized to extract multiword technical terms, which were then trained and transformed into representation vectors.

In this study, a *Java* prototype was built to assist researchers in implementing the proposed methodology to extract heterogeneous domain data elements and their semantic relations from plain text technical documents. The implementation procedure was according to the phases described in the proposed methodology. Specifically, the plain text roadway corpus was first fed into the system to generate a bag of roadway data elements, a data set of their representation vectors, and a collection of syntactically related pairs. This was followed by an evaluation of the semantic classifier algorithm and a comparison to several baseline models. The classifier was also tested with different parameter settings.

To evaluate the system performance, a test data set consisting of 22,500 pairs was developed, of which, there are 332 related pairs of words (88 is-a, 176 part-of, and 68 similar-to) and 22,168 nonrelated instances. The vocabulary of the test pairs was extracted from 1,000 sentences randomly selected from the highway corpus. By manually reviewing the automatically generated terms from the test sentences, 150 domain technical terms that appear two or more times were collected. Three Ph.D. students in civil engineering, including the first author, worked as annotators who independently identified and labeled the semantic relations among 150 words in the test vocabulary. The annotators were asked to assign one of the following three tags to a certain semantically related pair: part-of, is-a, and similar-to. Other pair combinations among 150 words beyond those discovered and tagged by annotators were automatically assigned the non-related tag. The knowledge base WordNet and various DOT roadway transportation glossaries were used during the annotation process. As a result, 332 pairs that at least two annotators agreed upon were obtained for the validation purpose. For a given pair of terms, the system returns one of the following tags: is-a, part-of, similar-to, and non-related. In this study, the following three measures are used to evaluate the semantic classifier: precision, recall, and F-measure. Let S_i denote a set of true pairs labeled with relation i in the test set, and S_i' is a set of pairs classified as relation i by the system. The evaluation metrics for a certain relation are defined in Eqs. (8)–(10). The overall system performance is evaluated using the same equations, but is based on the total correctly classified pairs for all types of relations

$$\text{Precision}_i = \frac{S_i \cap S_i'}{S_i'} \tag{8}$$

$$\text{Recall}_i = \frac{S_i \cap S_i'}{S_i} \tag{9}$$

$$F_i = \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{10}$$

To evaluate the success of the system, experiments were conducted to compare the performance between the proposed classifier and two other baseline methods. The first baseline model is one that purely uses lexical patterns learned in this study to detect the semantic relation between a given pair of terms. Because this uses only rules, similar-to is not applicable. The second baseline method uses Word2Vec without integrating pattern features. This model is basically the same as the proposed method, but all near words generated by Word2Vec are accepted as similar terms. Therefore, the comparison with this baseline method was only on the similar-to relation.

Output from Interim Steps

The first output from the system is a domain terminology set. There are almost 288,000 NPs extracted, of which more than 17,000 were accepted as technical terms after removing instances with stop words and applying the 15% cut-off policy. Table 3 shows the distribution of terms by sequence length along with the top five examples for each category. As shown, the majority are bigrams (65.62%), whereas lengthy NPs account for a relatively small portion in the corpus, i.e., 1.75 and 0.39%, respectively, for 5 and 6 grams. Using this terminology data set, the corpus was modified by

Table 3. Total Number of Extracted Terms

N-gram	Count	Percentage	Top 5 (C-Value)
Bigrams	11,446	65.62	Sight distance (9,701); design speed (9,376); traffic control (6,142); cross section (5,280); clear zone (4,837)
Trigrams	4,421	25.35	Right of way (7,945); traffic control device (3,188); contract unit price (2,836); left turn lane (1,976); portland cement concrete (1,930)
4-grams	1,180	6.76	Right of way line (1,147); uniform traffic control device (924); highway right of way (907); portland cement concrete pavement (737); right of way acquisition (564)
5-grams	306	1.75	Two way left turn lane (303), mdt statewide integrated roadside vegetation (241); portable precast concrete barrier rail (163); right of way control section (149); effective modulus of subgrade reaction (130)
6-grams	68	0.39	Positional accuracy of as built record (65); right turn fixed object pedestrian night (46); bridge rehabilitation technique steel superstructure reference (46); air void of compacted bituminous mixture (38); continuous two way left turn lane (38)
Total	17,443	100	

Note: C-Values are in parentheses.

Table 4. Patterns Learned and Examples of Pairs Extracted

Relation	Seeds	Patterns learned	Extracted pairs
X part-of Y	Alignment::roadway	X (of|at|in) (a|an|the) Y	Curb::roadway
	Median::roadway	Y (with|with no|without) (a|an) X	Sidewalk::bridge
	Ramp::interchange	Y ('s|where) X	Radius::horizontal curve
	(Total seeds: 10)	—	(Total pairs: 30,423)
X is-a Y	Highway::facility	X (, NP)*(,)? (and|or) other Y	Cracking::damage
	Culvert::drainage facility	Y (,)? such as (NP,)* X	Bridge::structure
	Sign::traffic control device	Y, including (NP,)* X	Crane::equipment
	(Total seeds: 7)	—	(Total pairs: 8,339)

Note: NP = noun phrase; | = or; * = zero or more; ? = zero or one.

connecting the tokens in the multiple-word terms with the minus sign to ensure that they are treated as single tokens.

The system was then applied on the modified corpus to extract lexical pairs. Table 4 shows the patterns learned and examples of instances harvested for the part-of and is-a relations. To collect pairs related through the two relations, 10 and 7 seeds, respectively, were used. Those seeds were obtained by reviewing various roadway transportation glossaries. As shown in Table 4, three groups of patterns were induced for each relation part-of and is-a. Using these patterns, approximately 30,000 part-of and 8,000 is-a pairs were collected.

Another important product generated by the system is a term space. Fig. 4 presents the vector space of roadway data elements derived from the word embedding training process when the parameters—frequency threshold, hidden layer size, and window size—were set to 5, 100, and 5, respectively. To present those high-dimensional vectors in a two-dimensional graph, principle component analysis (PCA) was used to reduce the dimension. Based on the distance between terms visualized in Fig. 4, the most related data elements for a certain data type can be quickly identified. For example, an inlet can be inferred to be more similar to an outlet, because they are grouped nearer to each other, than to a pavement. Table 5 shows a partial ranked list of the nearest terms of 'street' in order of similarity score.

Table 5. Examples of the Top Nearest Words

Target term	Nearest words	Cosine	Rank
Street	Highway	0.658	1
	Direct-access	0.583	2
	Collector-road	0.557	3
	Public-street	0.533	4
	Local-street	0.561	5
	Curb-extension	0.526	13
	On-street-parking	0.491	23

Table 6. Overall System Performance with Different Parameter Settings and Training Network Type

Model	Precision (%)	Recall (%)	F (%)
CBOW 5-100-5	92.76	81.02	86.50
CBOW 5-300-5	93.70	77.37	84.76
CBOW 50-100-5	84.44	85.71	85.07
Skip-gram 50-100-5	80.60	65.06	72.00
Skip-gram 50-100-15	76.15	54.82	63.75

System Performance

Before evaluating the system and comparing the performance with baseline methods, several experiments were carried out to identify the optimal value for three model parameters, frequency threshold, vector size, and context window size, and to select a better network type (CBOW or skip-gram) of the Word2Vec training model. To examine the effect of a certain parameter, its value in the standard setting was increased (5, 100, 5), while other parameters stayed unchanged. The training network type was also changed to determine the optimal setting. Table 6 shows the results from those experiments when the top synonym cluster parameter, c, was set to 2.

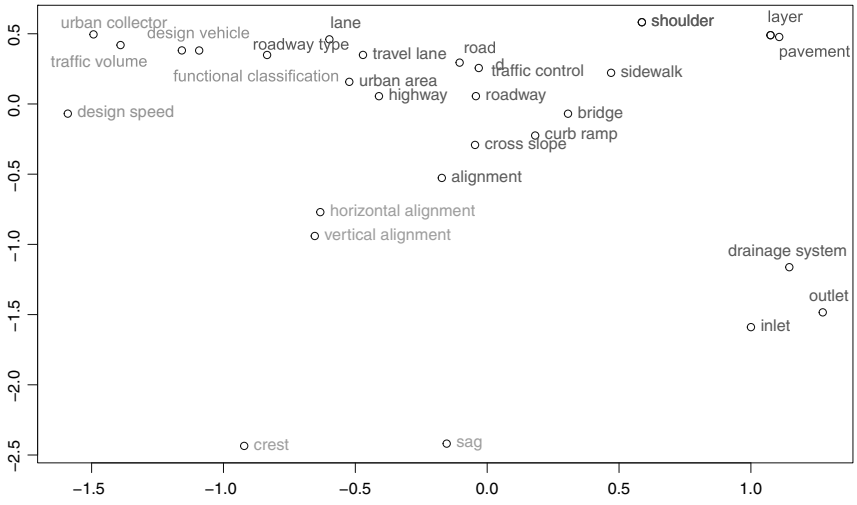

Fig. 4. PCAs representation of roadway term vectors

Table 7. System Performance

Model	Part-of			Is-a			Similar-to		
	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
Pattern only	80.00	95.45	87.05	81.93	77.27	79.53	—	—	—
CBOW only	—	—	—	—	—	—	70.0	61.76	65.63
CBOW + pattern	94.74	81.82	87.80	94.87	84.09	89.16	85.0	75.0	79.69

Note: P, R, and F = precision, recall, and F measure, respectively.

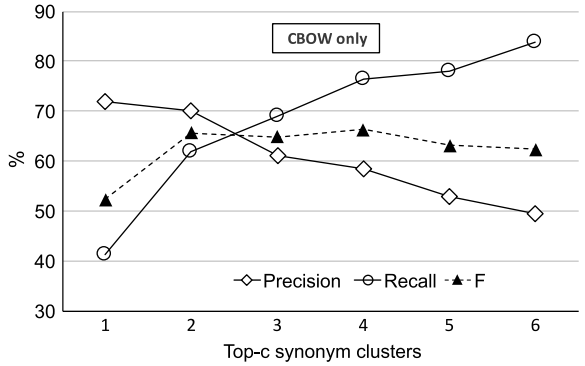

Fig. 5. Synonym detection performance for the CBOW model

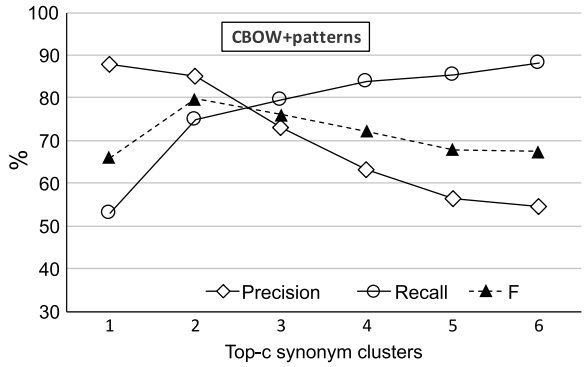

Fig. 6. Synonym detection performance for the CBOW + pattern model

The results indicate that neither increasing frequency threshold, hidden layer size, nor window size necessarily improves the system performance. In addition, CBOW shows its strong superiority to skip-gram in the classifying system. Thus, in the comparative testing with other baseline methods, the standard parameter set was used with the CBOW structure.

Table 7 shows the performance of the proposed method in comparison to the other two baseline models. The performance for similar-to in this table is in accordance with the best case (*F-score* reaching the maximum) when varying the number of top *c* clusters accepted (see Figs. 5 and 6, respectively, for CBOW and CBOW + Pattern models). The integration between syntactic patterns and semantic word vectors significantly improves both recall and precision for the is-a and similar-to relations (Table 7). A slight *F* enhancement is also observed for the part-of relation. Among those three relations, is-a has the best performance with a precision of nearly 95% and a recall of approximately 85%. These impressive figures yield a 14% *F* improvement over the pattern-based approach, in which a major contribution is from the precision. It is evident that once semantic relatedness is considered, incorrect instances matching the syntactic is-a patterns can be effectively eliminated. Detecting synonymy, which is the most challenging task, also achieves a relative *F-score* of 79.69% compared with 65.63% when solely using CBOW. With respect to part-of detection, the integrated method greatly enhances the precision from 80 to 94.74%; however, because of a considerable drop in recall, the overall *F* improvement is just 0.75%. This result indicates that the induced part-of patterns are highly reliable; thus, the inclusion of semantic features gives only a slight improvement.

Research Findings, Implications, and Limitations

This paper provides many important contributions to the area of integrating transportation asset data. The disparity of data names and semantics is a major hurdle to merging disconnected transportation data sources. This study provides a novel linguistic methodology to assist in classifying heterogeneous data items using linguistic information in technical text documents. Specifically, this study contributes to the body of knowledge by (1) developing an NLP-based method for automated extraction of data types and their name variants from design manuals; (2) introducing a machine-learning approach that can learn the similarity in meaning among data items using their context words in texts; and (3) designing an algorithm that integrates syntactic rules, clustering, and word embedding to classify lexical relations among heterogeneous terms. The main merit of the study lies in the detection of linguistic inconsistency in naming the same data element. This capability enables data integration to precisely combine similar data even given different terms in different systems. Another key advantage is the use of only linguistic information in domain texts for semantic relatedness identification. By purely using the occurrence of data elements in domain documents, the classifying algorithm overcomes the limitations of costly handcrafted rules as used by Abuzir and Abuzir (2002) and Rezgui (2007), and eliminates the reliance on other existing dictionaries such as in the work by Zhang and El-Gohary (2016).

The present framework is not to completely eliminate human involvement, but it is expected to offer an enabling tool to assist researchers in developing supporting ontologies, taxonomies, and other forms of semantic resources with the inclusion of alternative names for a concept. Using the method presented in this paper, less effort is required because the only major requirement is collecting domain documents. Researchers may need to pay some effort toward validating the automatically generated data sets, but it is much less time-consuming than interviewing domain experts or manually examining written documents. Although the methodology has been tested only on a roadway corpus, it is generic and its applicability is broad. For example, the developed system can be implemented to extend the buildingSmart building data dictionary (buildingSMART 2016). The findings of this study would accelerate the process of removing the current bottleneck of

Table 8. Excerpts of Extracted Near-Synonym Set

Number	Synonym set
1	Highway; road; street
2	Crosswalk; crosswalk-line; pedestrian-crossing
3	Roundabout; traffic-circle; splitter-island
4	Traffic-island; refuge-island; pedestrian-refuge
5	Subbase; subgrade; base-course; base-layer
6	Grade-separation; at-grade-crossing; interchange; overpass

machine readable dictionaries, which are required for unambiguous data sharing, integration, and exchange.

In addition to theoretical implications, the outcome of this study offers practical value to the highway industry. The data sets resulting from the experiment in this study provide name variants and related items for more than 17,000 roadway data elements. For example, some of the alternative ways to present 'right of way' include 'row', 'r/w', or 'r.o.w.'. Several other examples of synonym sets generated from the system are shown in Table 8. The full library of terminology network generated from this study provides practitioners with suggestions on data keywords, their variations, and related data when finding data from external databases.

The current study has a number of limitations. The classifying algorithm covers only three types of semantic relations that are synonymy, hyponymy, and meronymy. Several other important relations that are not considered include siblings and functional associations, among others. The inclusion of these relations into the classifier would reduce incorrect synonym matching, which will enhance the precision value. In addition, this study only targets the synonymy issue; the polysemy obstacle is not yet addressed. Further research is needed to detect different senses of terms. Because a term that has multiple meanings would occur in different contexts, one potential solution is to cluster the instances of context words. A spread of contexts is a strong indication that a given term may refer to multiple things.

Conclusions

Data manipulation from multiple sources is a challenging task in transportation asset management because of the inconsistency of data terminology. The key contribution of this study is a novel approach for automated classification of semantic relations among heterogeneous data elements. In the proposed framework, machine learning was used to train the semantic similarity between technical terms. An algorithm was also designed to classify the nearest terms resulting from the semantic similarity model into distinct groups in accordance with their lexical relationships.

The developed system was tested and evaluated on a 16-million-word corpus of roadway design manuals collected from 30 state DOTs across the United States. The system performance was assessed by comparing automatically classified relations with those in a human-crafted gold standard. The result shows an overall performance of 92.76% in precision and 81.02% in recall. The best model is associated with the CBOW training structure and a parameter setting of 5, 100, and 5, respectively, for frequency threshold, hidden layer size, and window size. One area for future studies is to improve the recall score, which can be done by considering additional relation types. In addition, this paper focuses only on synonymy; research is needed to address the polysemy issue among data elements.

The proposed automated methodology for detecting semantic relations between data elements from texts is expected to significantly reduce human efforts in developing semantic resources for specific use cases in, but not limited to, the field of transportation asset management. Once digital data dictionaries become readily available, the level of semantic interoperability can be fully achieved in the construction industry.

Acknowledgments

This research was funded by the National Science Foundation (NSF) through Award NSF-CIS 420-60-83. The authors gratefully acknowledge NSF's support. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

Abuzir, Y., and Abuzir, M. O. (2002). "Constructing the civil engineering thesaurus (CET) using the ThesWB." *Computing in civil engineering*, ASCE, Reston, VA.

Ananiadou, S., Albert, S., and Schuhmann, D. (2000). "Evaluation of automatic term recognition of nuclear receptors from MEDLINE." *Genome Inf.*, 11, 450–451.

Apache OpenNLP. (2017). "OpenNLP." ⟨https://opennlp.apache.org/⟩ (Apr. 2, 2017).

Bittner, T., Donnelly, M., and Winter, S. (2005). "Ontology and semantic interoperability." *Large-scale 3D data integration: Challenges and opportunities*, CRC Press, Boca Raton, FL, 139–160.

buildingSMART. (2016). "buildingsmart data dictionary." ⟨http://bsdd.buildingsmart.org/⟩ (Mar. 15, 2016).

Cambria, E., and White, B. (2014). "Jumping NLP curves: A review of natural language processing research." *IEEE Comput. Intell. Mag.*, 9(2), 48–57.

CeTermClassifier. (2017). "GitHub." ⟨https://github.com/tuyenbk/CeTermClassifier⟩ (Jul. 15, 2017).

Chen, D., and Manning, C. D. (2014). "A fast and accurate dependency parser using neural networks." *Proc., 2014 Conf. on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 740–750.

Church, K. W., and Hanks, P. (1990). "Word association norms, mutual information, and lexicography." *Comput. Ling.*, 16(1), 22–29.

Costa-Jussa, M. R., Farrús, M., Mariño, J. B., and Fonollosa, J. A. (2012). "Study and comparison of rule-based and statistical Catalan-Spanish machine translation systems." *Comp. Inf.*, 31(2), 245–270.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). "GATE: A framework and graphical development environment for robust NLP tools and applications." *Proc., 40th Anniversary Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudburg, PA, 168–175.

El-Diraby, T., and Kashif, K. (2005). "Distributed ontology architecture for knowledge management in highway construction." *J. Constr. Eng. Manage.*, 10.1061/(ASCE)0733-9364(2005)131:5(591), 591–603.

El-Diraby, T., Lima, C., and Feis, B. (2005). "Domain taxonomy for construction concepts: Toward a formal ontology for construction knowledge." *J. Comput. Civil Eng.*, 10.1061/(ASCE)0887-3801(2005)19:4(394), 394–406.

Erk, K. (2012). "Vector space models of word meaning and phrase meaning: A survey." *Lang. Ling. Compass*, 6(10), 635–653.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). "Automatic recognition of multi-word terms: The C-value/NC-value method." *Int. J. Digital Libraries*, 3(2), 115–130.

Gallaher, M. P., O'Connor, A. C., Dettbarn, J. L., and Gilday, L. T. (2004). *Cost analysis of inadequate interoperability in the U.S. capital facilities industry*, U.S. Dept. of Commerce Technology Administration, National Institute of Standards and Technology, Gaithersburg, MD.

Harispe, S., Ranwez, S., Janaqi, S., and Montmain, J. (2013). "Semantic measures for the comparison of units of language, concepts or instances from text and knowledge base analysis." *arXiv 1310*, 1285.

Harris, Z. S. (1954). "Distributional structure." *Word*, 10(2–3), 146–162.
Harrison, F., Gordon, M., and Allen, G. (2016). "Leadership guide for strategic information management for state departments of transportation." *NCHRP Rep. 829*, National Academies Press, Washington, DC.
Hearst, M. A. (1992). "Automatic acquisition of hyponyms from large text corpora." *Proc., 14th Conf. on Computational Linguistics*, Vol. 2, Association for Computational Linguistics, Stroudsburg, PA, 539–545.
Heiler, S. (1995). "Semantic interoperability." *ACM Comput. Surv.*, 27(2), 271–273.
Hezik, M. (2008). "IFD library background and history." *The IFD Library/IDM/IFC/MVD Workshop*, Building Smart, VA.
Hill, F., Reichart, R., and Korhonen, A. (2015). "Simlex-999: Evaluating semantic models with (genuine) similarity estimation." *Comput. Ling.*, 41(4), 665–695.
Hsu, J.-Y. (2013). "Content-based text mining technique for retrieval of CAD documents." *Autom. Constr.*, 31, 65–74.
Inkpen, D., and Hirst, G. (2006). "Building and using a lexical knowledge base of near-synonym differences." *Comput. Ling.*, 32(2), 223–262.
ISO. (2007). "Building construction—Organization of information about construction works. Part 3: Framework for object-oriented information." *ISO 12006-3*, Geneva.
Jivani, A. (2011). "A comparative study of stemming algorithms." *Int. J. Comp. Tech. Appl.*, 2(6), 1930–1938.
Justeson, J. S., and Katz, S. M. (1995). "Technical terminology: Some linguistic properties and an algorithm for identification in text." *Nat. Lang. Eng.*, 1(1), 9–27.
Karimi, H. A., Akinci, B., Boukamp, F., and Peachavanish, R. (2003). "Semantic interoperability in infrastructure systems." *4th Joint Int. Symp. on Information Technology in Civil Engineering*, ASCE, Reston, VA, 42–42.
Kolb, P. (2008). "Disco: A multilingual database of distributionally similar words." *Proc., Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)-2008*, Berlin.
Lefler, N. X. (2014). "Roadway safety data interoperability between local and state agencies." *NCHRP Rep.*, National Academies Press, Washington, DC.
Lesk, M. (1986). "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone." *Proc., 5th Annual Int. Conf. on Systems Documentation*, Association for Computing Machinery, New York, 24–26.
Levy, O., Goldberg, Y., and Dagan, I. (2015). "Improving distributional similarity with lessons learned from word embeddings." *Trans. Assoc. Comput. Ling.*, 3, 211–225.
Lima, C., El-Diraby, T., and Stephens, J. (2005). "Ontology-based optimization of knowledge management in e-construction." *J. IT Constr.*, 10(21), 305–327.
Lopes, L., and Vieira, R. (2015). "Evaluation of cutoff policies for term extraction." *J. Braz. Comput. Soc.*, 21(1), 9.
Lossio-Ventura, J. A., Jonquet, C., Roche, M., and Teisseire, M. (2013). "Combining C-value and keyword extraction methods for biomedical terms extraction." *LBM'2013: 5th Int. Symp. on Languages in Biology and Medicine*, Database Center for Life Science, Tokyo.
Lv, X., and El-Gohary, N. M. (2015). "Semantic annotation for context-aware information retrieval for supporting the environmental review of transportation projects." *2015 Int. Workshop on Computing in Civil Engineering*, ASCE, Reston, VA, 165–172.
MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations." *Proc., 5th Berkeley Symp. on Mathematical Statistics and Probability*, Vol. 1, Oakland, CA, 281–297.
Marcus, M. (1995). "New trends in natural language processing: Statistical natural language processing." *Proc. Nat. Acad. Sci.*, 92(22), 10052–10059.
Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). "Building a large annotated corpus of English: The Penn Treebank." *Comput. Ling.*, 19(2), 313–330.
Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). "Efficient estimation of word representations in vector space." *arXiv* 1301, 3781.
Mounce, S., Brewster, C., Ashley, R., and Hurley, L. (2010). "Knowledge management for more sustainable water systems." *J. Inf. Technol. Constr.*, 15(11), 140–148.
Navigli, R. (2009). "Word sense disambiguation: A survey." *ACM Comput. Surv.*, 41(2), 1–69.
Navigli, R., and Velardi, P. (2010). "Learning word-class lattices for definition and hypernym extraction." *Proc., 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 1318–1327.
Nenadić, G., Spasić, I., and Ananiadou, S. (2002). "Automatic acronym acquisition and term variation management within domain-specific texts." *3rd Int. Conf. on Language Resources and Evaluation (LREC2002)*, European Language Resources Association, Paris, 2155–2162.
Noy, N. F. (2004). "Semantic integration: A survey of ontology-based approaches." *ACM Sigmod Rec.*, 33(4), 65–70.
Osman, H., and Ei-Diraby, T. (2006). "Ontological modeling of infrastructure products and related concepts." *Transp. Res. Rec.*, 1984, 159–167.
Ouksel, A. M., and Sheth, A. (1999). "Semantic interoperability in global information systems." *ACM Sigmod Rec.*, 28(1), 5–12.
Pantel, P., and Pennacchiotti, M. (2006). "Espresso: Leveraging generic patterns for automatically harvesting semantic relations." *Proc., 21st Int. Conf. on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 113–120.
Pennington, J., Socher, R., and Manning, C. D. (2014). "GloVe: Global vectors for word representation." ⟨http://www.aclweb.org/anthology/D14-1162⟩ (Mar. 7, 2017).
PlingStemmer [Computer software]. Cognitive Computation Group, Urbana, IL.
Princeton University. (2017). "About WordNet." ⟨http://wordnet.princeton.edu/wordnet/⟩ (Mar. 7, 2017).
Radim, R. (2014). "Word2vec tutorial." ⟨http://rare-technologies.com/word2vec-tutorial/⟩ (Mar. 3, 2017).
Rezgui, Y. (2007). "Text-based domain ontology building using Tf-Idf and metric clusters techniques." *Knowl. Eng. Rev.*, 22(04), 379–403.
Salton, G., and Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval." *Inf. Process. Manage.*, 24(5), 513–523.
Sclano, F., and Velardi, P. (2007). "Termextractor: A web application to learn the shared terminology of emergent web communities." *Enterprise interoperability*, Springer, London, 287–290.
Seedah, D. P., Choubassi, C., and Leite, F. (2015a). "Ontology for querying heterogeneous data sources in freight transportation." *J. Comput. Civil Eng.*, 10.1061/(ASCE)CP.1943-5487.0000548, 04015069.
Seedah, D. P., Sankaran, B., and O'Brien, W. J. (2015b). "Approach to classifying freight data elements across multiple data sources." *Transp. Res. Rec.*, 2529, 56–65.
Sparck Jones, K. (1972). "A statistical interpretation of term specificity and its application in retrieval." *J. Doc.*, 28(1), 11–21.
Suchanek, F. M., Ifrim, G., and Weikum, G. (2006). "Leila: Learning to extract information by linguistic analysis." *Proc., 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, Association for Computational Linguistics, Stroudsburg, PA, 18–25.
Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). "Feature-rich part-of-speech tagging with a cyclic dependency network." *Proc., 2003 Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, Association for Computational Linguistics, Stroudsburg, PA, 173–180.
Turney, P. D., and Pantel, P. (2010). "From frequency to meaning: Vector space models of semantics." *J. Artif. Intell. Res.*, 37(1), 141–188.
Walton, C. M., et al. (2015). *Implementing the freight transportation data architecture: Data element dictionary*, *Number Project NCFRP-47*, National Academies Press, Washington, DC.
Webster, J. J., and Kit, C. (1992). "Tokenization as the initial phase in nlp." *Proc., 14th Conf. on Computational Linguistics-Volume 4*, Association for Computational Linguistics, Stroudsburg, PA, 1106–1110.
Wetherill, M., Rezgui, Y., Lima, C., and Zarli, A. (2003). "Knowledge management for the construction industry: The e-cognos project." *J. Inf. Technol. Constr.*, 7(12), 183–196.

Yalcinkaya, M., and Singh, V. (2015). "Patterns and trends in building information modeling (bim) research: A latent semantic analysis." *Autom. Constr.*, 59, 68–80.

Yarowsky, D. (1995). "Unsupervised word sense disambiguation rivaling supervised methods." *Proc., 33rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 189–196.

Zhang, J., and El-Gohary, N. (2016). "Extending building information models semiautomatically using semantic natural language processing techniques." *J. Comput. Civil Eng.*, 10.1061/(ASCE)CP.1943-5487.0000536, C4016004.

Zhang, Z., Iria, J., Brewster, C., and Ciravegna, F. (2008). "A comparative evaluation of term recognition algorithms." *Proc., 6th Int. Conf. on Language Recourses and Evaluation*, European Language Resources Association, Paris.

Zhao, H., and Kit, C. (2011). "Integrating unsupervised and supervised word segmentation: The role of goodness measures." *Inf. Sci.*, 181(1), 163–183.