A Semantic Similarity Model to Compare Heterogeneous Data Sources to Augment Engineering Data with New Failure modes in Automotive Industry

Dnyanesh Rajpathak[1] and John Cafeo[2]

[1,2]*Advanced Analytics Center of Excellence, Chief Data and Analytics Office, General Motors, Warren, Michigan, 48092-2031, USA*

(dnyanesh.rajpathak, john.cafeo)@gm.com

ABSTRACT

In industry, the capture of symptoms and failure modes during a fault event provides valuable information for effectively improving the product quality. An ontology-based semantic similarity system is employed enabling automatic comparison of engineering data (in the form of design failure mode effect analysis, DFMEA) with field repair data collected during the product warranty period to discover new symptoms and failure modes to augment the DFMEA.

The complexity of engineering data and the overwhelming volume of field repair data makes the task of identifying new symptoms and failure modes an impractical one from first principles. While the engineering data is structured, technical in nature, e.g. Seat Belt comfort per GMUTS, the field repair data is unstructured consisting of different noises. Some examples of noises observed in the field repair data include abbreviations, inconsistent use of vocabulary ('seat buckle is damaged' vs 'buckle unlatching'), misspellings, etc. Not surprisingly, text mining and semantic similarity are gaining serious attention due to their ability to link heterogeneous data sources and discover knowledge assets latent in text.

In our approach, initially the key constructs (e.g. components (parts), symptoms, failure modes) from the data are annotated by using the domain ontology. From these constructs *pairs of terms* and *pairs of tuples* are constructed to compute term-to-term and tuple-to-tuple semantic similarity respectively. Finally, text-to-text semantic similarity is calculated by combining term-to-term and tuple-to-tuple similarity scores.

The proposed method is implemented as a prototype tool and its performance is validated by using real-life data from the automobile domain. On average, our system has achieved the F1 score of 0.78 and 0.75 in discovering synonym and new symptoms respectively, whereas it achieved an F1 score of 0.72 and 0.68 in discovering synonym and new failure modes respectively. On average, the fault detection and the fault isolation rates improved from 0.51 to 0.86 and 0.50 to 0.92 respectively.

1. INTRODUCTION

Design failure mode and effect analysis (DFMEA) related to complex systems (e.g. turbines, automotive, aerospace, power plants) captures the critical engineering information (Sutrisno & Lee, 2011). It includes information related to key components, their fault free boundary conditions, functional information, symptoms, and failure modes in the event of fault, their severity, and frequency. Benedittini, Baines, Lightfoot, and Greenough (2009) reports that the complex architecture of modern vehicles typically involves several sensors, software, and electrification. Abdallah, Feron, Hellestrand, Koopman, and Wolf (2010) states that with such rapid growth of technology and its inclusion in modern vehicles leads to potential complicated faults due to inter-system interactions and communications. Any deviation of a complex system from its fault free state into a faulty state requires an in-depth fault diagnosis to detect the root-causes. Typically, the field repair data (henceforth verbatim) is collected throughout the warranty period of vehicles. It captures important component information observed during the fault event along with symptoms and failure modes. To improve the fault diagnostics and product quality, it is crucial that the new symptoms and failure modes are discovered in time by comparing field data with the engineering data.

The task of identifying new symptoms and failure modes from the first principles is an impractical one primarily due to the complexity of engineering data coupled with the overwhelming volume of verbatim data. Typically, the engineering data is structured in nature and it uses technical vocabulary, e.g. "unstable electric contact", "Seat Belt comfort per GMUTS", whereas the verbatim data captured in free-flowing English and due to the lack of controlled

vocabulary it is unstructured in nature. Generally, it gives rise to vocabulary mismatches ('seat buckle is damaged' vs 'buckle unlatching') and different types of noises are observed in the data, e.g. abbreviated text, misspellings, inclusion of additional white space, run-on-words.

Not surprisingly, text mining (Hearst, 1999) is gaining serious attention due to its ability to automate the process of knowledge discovery by training a machine. The semantic similarity on the other hand facilitates automatic comparison and linking of high-volume, heterogeneous data sources. In the past, Atamer (2004), Wirth, Kramer, and Peter (1996), Price and Taylor (1997) have proposed different systems to compare Failure Modes and Effect Analysis data and field experience. Different systems were developed by Rajpathak, Chougule, and Bandyopadhyay (2010) and Rajpathak (2013) to analyze the warranty data to provide an early indication of product abnormalities. However, limited efforts have been invested to compare and relate field data with DFMEAs. In Figure 1, we depict the scope of this work toward bridging the gap in literature.

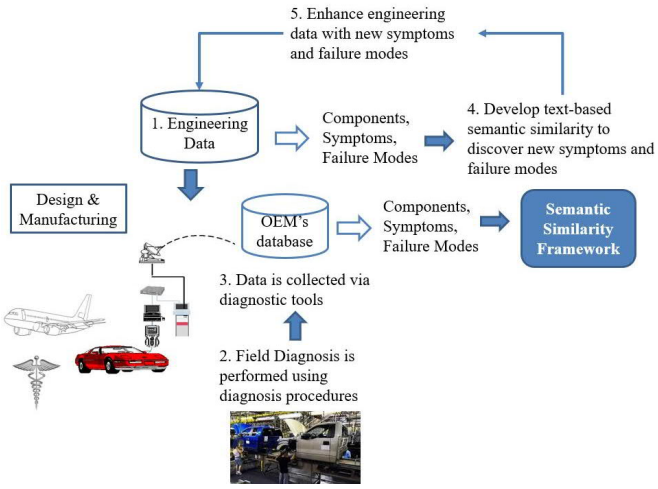

Figure 1. Scope of the ontology-based semantic-similarity framework.

Our ontology based semantic similarity model first identifies components (parts), symptoms, and failure modes mentioned in the verbatim and engineering data. During the fault event, additional indicators like the error codes and scanned values of operating parameters are also collected. In this work, a dependency is established between either a single or multiple failure modes[1], say f_1, f_2, …, f_i with a single or multiple symptoms, say s_1, s_2, …, s_j to generate meaningful constructs. These constructs are then used to construct *pairs of terms* and *pairs of tuples* (i.e. pairs of multi-term phrases) that are used to calculate *term-to-term* and *tuple-to-tuple* semantic similarity respectively. Finally, the *term-to-term* and *tuple-to-tuple* semantic similarity scores are combined to calculate *text-to-text* semantic similarity to discover new symptoms or failure modes from the verbatim data. In specific we make the following contributions – 1. A principled approach is proposed to compare industrial scale heterogeneous data, which overcomes the existing limitation of having to rely on the first principles; 2. Our hierarchical semantic similarity model successfully handles multi-term phrases, which helps us to overcome the key limitation of relying on single term phrases for computing semantic similarity. We successfully combine bottom-up *term-to-term* and *tuple-to-tuple* scores to perform a robust and realistic comparison between two data sources via *text-to-text* semantic similarity; and 3. In literature, e.g. Atamer (2004) the relationship between failure modes is identified only using those failure modes that are anticipated at a design stage. Since our approach discovers synonymous as well as new symptoms and failure modes, it enhances the relationship between symptoms and failure modes.

The relevant literature is reviewed in the next section. In section 3, we first introduce our domain knowledge model in terms of the domain ontology. Then, we discuss in detail how the collocates are identified from the DFMEA and verbatim data. These collocates are used as the candidates to compute semantic similarity between any two documents. Next, our hierarchical semantic similarity model is discussed in detail. In Section 4, we discuss the experiments and finally in section 5 we conclude our paper by reiterating the key contributions.

2. STATE-OF-THE-ART

The work in DFMEA can be broadly divided into two areas (Sutrisno & Lee, 2011): 1. Automation and modification of DFMEA construction and 2. Enhancement of DFMEA using model failure phenomenon and its combination with other quality tools. The work related to DFMEA automation and modification is the most relevant to our proposal. Zhang and Guogi (2013) proposed different MATLAB/Simulink models to process DFMEAs along with the nominal system behavior. It combines digital components (software and hardware) with the mechanical models, i.e. the environment in which the system runs. This model is augmented with the fault models for the digital and mechanical systems to create the Extended System Model. Papadopoulous, Parker, and Grante (2004) proposed a tool for the automatic synthesis of FMEAs. It builds upon their earlier work of synthesizing the fault trees in which the FMEAs are built from engineering diagrams and they are augmented with the component failures. The overall effectiveness of DFMEA is studied by Carlson (2012) and four broad factors are identified for the successful application of FMEA. One specific factor reports that a critical mistake in failing to make FMEAs an effective tool is a "disconnect between FMEA and field lessons (failure modes) learned." Although, a need to discover new failure modes is pointed

[1] Failure mode: The term failure mode is generally used to indicate the root-causes associated with the faults observed particularly in the engineering systems. However, in our work we do not necessarily restrict ourselves to a specific set of systems while using the notion of a failure mode.

out, no indication is given about how such discovery can be made. Tso, Tai, Chau, and Alkalai (2005) have reported their experience in developing a design-for-safety workbench, *risk assessment and management environment (RAME)* for microelectronic avionics systems. RAME consists of a test-reporting/failure-tracking system and it is an off-the-shelf data mining tool, a knowledge base, and a fault model that permits systematic learning from the prior projects to automate FMEA. Finally, Atamar (2004) reports a case study for a turbofan engine that illustrates how the case-based reasoning is used to assess an overlap of FMEA symptoms with its counterpart in a case base.

In literature, the term semantic similarity is interchangeably used with semantic relatedness. While the former one assigns a metric to the terms that are members of two text data and it calculates their content similarity, the latter one handles the notions of antonym and meronymy (e.g. the *part-of* relation). The resultant metric provides a number between 0 and 1, where 0 denotes no similarity between two text data and 1 denotes identical text data. Several techniques have been proposed in the literature to identify semantic similarity and they can be broadly classified into the following categories – Deerwester, Dumais, Furnas, Landauer, and Harshman (1990) proposed latent semantic indexing (LSI), point-wise mutual information (Turney, 2001), second-order co-occurrence pointwise mutual information (Islam & Inkpen 2008), corpus based semantic representation (Grefenstette, 1994), (Lund & Curt 1996), (Landauer & Susan 1997), (Schütze, 1997), Griffiths, Steyvers, and Tenenbaum (2007), (Padó & Lapata 2007), (Lenci, 2008), (Turney & Pantel, 2010), and ontology-structure based measures (Rada et al., 1989) (Wu & Palmer, 1994) (Leacock & Chodorow, 1998) (Bullinaria & Levy, 2007). The LSI model (Deerwester et al., 1990) analyzes relationships between different documents by using the terms contained in them. A term-document matrix is constructed in which the rows represent words and the columns represent number of documents in which the words appear. The singular value decomposition is used to reduce the dimensionality and the cosine similarity between two vectors is calculated to determine how close two words are with each other on the scale of 0-1. In the point-wise mutual information (PMI) model (Turney, 2001), the notion of co-occurrence between the phrases reported in two text data, say problem and choice_i are used to calculate their co-occurrence probability. The statistically independent phrase probability is given as $p = (problem\&choice_i)$, while the dependent ones is calculated as, $\frac{p(problem\&choice_i)}{p(problem)p(choice_i)}$. Grefenstette, (1994), (Lund & Curt, 1996), (Landauer & Susan, 1997), (Schütze, 1997) and (Padó & Lapata, 2007) proposes models in which the semantic similarity between two phrases is determined in terms of the attributional similarity. Medin, Goldstone, and Gentner (1990) proposed the relational similarity. Islam and Inkpen (2008) proposed a new corpus-based model, referred to as the Second Order Co-occurrence PMI (SOC-PMI) to calculate the semantic similarity between two target words, say W1 and W2. In SOC-PMI model, a context window is used to collect the co-occurring words with W1 and W2 and the common words are retained. The frequency of common words is calculated. Finally, the PMI of common words is aggregated to calculate relative semantic similarity score.

Several corpus-based semantic similarity models are proposed in the literature, e.g. Griffiths, Steyvers, and Tenenbaum (2007), (Padó & Lapata, 2007), (Lenci, 2008), and (Turney & Pantel, 2010). In a corpus-based semantic similarity model, the meaning of linguistic expressions is characterized in term of the distributional properties. It is referred to as the Distributional Semantic Model (DSM). The DSM model relies on a variation of the distributional hypothesis (Miller & Charles 1991) to calculate semantic similarity. It makes use of the attributional and relational similarity (Turney, 2001) along with unstructured DSM and structured DSM. In unstructured DSM, the distributional data is represented by the unstructured co-occurring relations between an element and a context. The lexical collocates within a certain distance from a word are collected. The structured DSM collects the co-occurrence statistics in the form of corpus-based triples and the context is said to be linked to a word if a lexico-syntactic relationship exists among them. However, as shown by Lenci (2008) the use of co-occurrence statistics without using a domain model leads to an incompatible semantic space problem. In other words, the constructs even when not related to each other are shown to be similar with each other simply based on the common underlying distribution. In Rada, Mili, Bicknell, and Blettner (1989), the 'Distance' metric is proposed to compute the average minimum path length over all pair wise combinations of concepts between two sub-concepts. In Wu and Palmer (1994), a semantic similarity between two concepts, say C1 and C2 is calculated by using WordNet (Miller 1995). In Leacock and Chodorow (1998), the similarity between two concepts is measured based on the shortest path between two concepts in a 'is-a' hierarchy of a taxonomy. In Bullinaria and Levy (2007), a semantic similarity model makes use of the co-occurrence statistics between two words, say W1 and W2 and a word window of a specific size is used to collect the co-occurring context information.

3. ONTOLOGY-BASED SEMANTIC SIMILARITY MODEL

Figure 2 shows different components involved in our model. In our model, the key constructs, such as components (parts), symptoms, and failure modes are identified from the DFMEA and the verbatim data. As it can be seen in figure 2, the domain ontology is used to identify the key constructs reported in the data. Next, we compute criticality of each key construct in terms of term frequency inverse document frequency (tf*idf) (Spärck Jones, 1972). The constructs above a specific threshold values are then used to first calculate *term-to-term* and *tuple-to-tuple* semantic similarity scores. The *text-to-text* semantic similarity combines the results of *term-to-term* and *tuple-to-tuple*. Finally, the rules

are derived to determine whether the symptoms or failure modes are synonym to each other, or it is a new knowledge. In the next section, we will discuss the domain ontology and its internal structure.

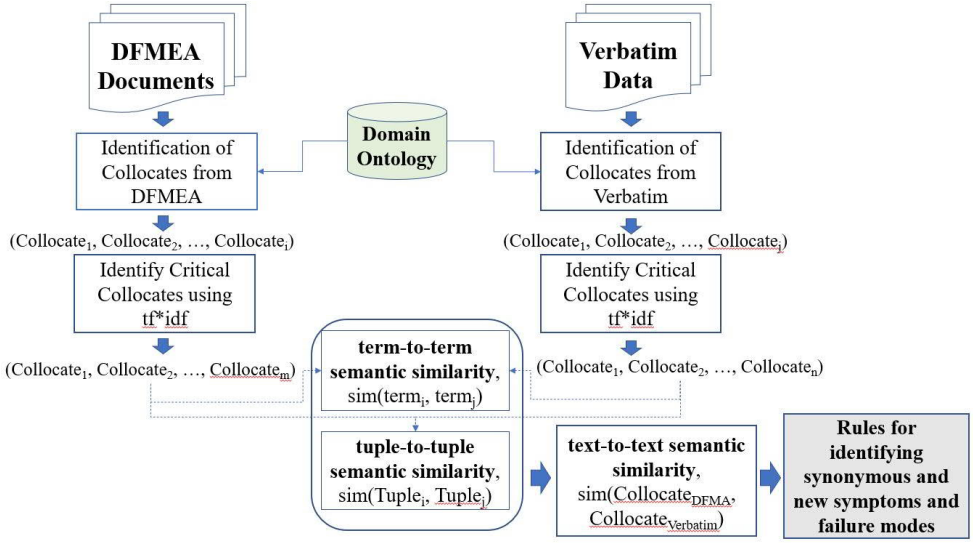

Figure 2. The different components involved in the ontology-based semantic similarity model.

3.1. Domain ontology as a knowledge model

The domain ontology consists of three key elements, $DFMEA_{onto} = (C_i, C_{i\_subconcept}, I_{ci})$, where C_i represents a finite set of classes, $C_{i\_subconcept}$ represents sub-classes of C_i commonly observed in our domain, and I_{ci} represents the class instances extracted from the data to populate C_i and $C_{i\_subconcept}$. Our ontology does not make subscription to any specific automotive applications and, therefore, as discussed by Fausto, Maurizio, and Ilya (2007) it is a descriptive lightweight ontology.

The C_i consists of the following classes: $(\texttt{Component}_i, \texttt{Symptom}_j, \texttt{Failure Mode}_k) \in C_i$. The class $\texttt{Component}_i$ is used to formalize a set of engineering artifacts in our domain, e.g. `transmission`, `clutch`, `fuelTank`. Any deviation of a system from its normal operating state is considered as a faulty state. The engineering faults are acquired in terms of the symptoms and they are formalized by using the class $\texttt{Symptom}_j$. The failure modes represent the root causes of the observed symptoms and they are are formalized by using the class $\texttt{Failure Mode}_k$.

More specific domain knowledge is formalized in terms of the class-subclass hierarchy, $C_{isubconcept}$. The classes at the top-level in a class hierarchy are more generic, whereas the ones further down in a hierarchy are more specific in nature. For example, the class $\texttt{Symptom}_j$ is further specialized into two subclasses - `faultCodeSymptom` and `textSymptom`. The former class is used to formalize fault or diagnostic trouble codes (e.g. P0100) observed during fault, whereas the latter one is used to formalize the text symptoms reported by the customers, e.g. ‘rattling noise’, ‘vehicle hard start’.

Finally, the instances I_{ci} acquired from the data are used to instantiate the specific classes. For example, the concept $\texttt{Symptom}_j$ can be instantiated with the literal observations, such as ‘`vehicleRunsRough`’, ‘`windowNotRollingUp`’, ‘`P0100`’ and so on. These instances are used to identify key constructs reported in the data and they are used to compute the *term-to-term*, *tuple-to-tuple*, and *text-to-text* semantic similarity scores.

3.2. Identification of collocates from the DFMEA data

The class instances (cf. Section 3.1) are used to tag the phrases in DFMEAs. However, the term identification is a non-trivial exercise mainly due to the vocabulary mismatch. In many cases, at a surface level the phrases in DFMEA, say $Phrase_{DFMEA}$ and the class instances, say $Phrase_{instance}$ may share very few or no terms with each other, e.g. ‘not $working_{DFMEA}$’ and ‘intermittent $open_{instance}$’, but they refer to the same fault. In literature, different approaches are proposed to handle the vocabulary mismatch problem, e.g. Shekarpour, Marx, Auer, and Sheth (2017) uses the stemming approach, the morphology approach (Krovetz, 1993), LSI (Deerwester et al., 1990), the translation model (Berger & Lafferty, 1999), and the query expansion (Lavrekno & Croft, 2001).

We develop a context-based query expansion probability model to handle the vocabulary mismatch problem. Different steps involved in our model are discussed below:

Step 1. $Phrase_{DFMEA}$ and $Phrase_{instance}$ are treated as the queries, say Q_i and a common set of documents (D_c), say C related to each query Q_j are identified such that $Phrase_{DFMEA} \wedge Phrase_{instance} \in Q_j$ as shown in Eq. (1) and (2).

$$R := \{r \in C : Relevant\,(r, Q_i)\} \quad (1)$$

$$D_c = R_{Phrase_{DFMEA}} \cap R_{Phrase_{instance}} \quad (2)$$

Step 2. Next, the terms co-occurring with $Phrase_{DFMEA}$ and $Phrase_{instance}$ in D_c are collected. The notion of co-occurrence is established by applying a word window of five terms (determined empirically) on the either side of $Phrase_{DFMEA}$ and $Phrase_{instance}$ in a data in which they are reported. The co-occurring terms, $Term_{DFMEA}$ and $Term_{instance}$ within a specific word window are collected.

Step 3. In the event of fault, a dominant set of failure modes, f_i are captured from the verbatim data. Along with the context information collected in Step 2, the f_i are used as a complimentary context information. Next, the probability $P(f_i \mid Term_{DFMEA})$ and $P(f_i \mid Term_{instance})$ is estimated from the aggregate data to determine their similarity with each other. For the sake of brevity, we only show the calculations of $P(f_i \mid Term_{DFMEA})$ and the calculations of $P(f_i \mid Term_{instance})$ can be realized on the same lines.

$$P(f_i|Term_{DFMEA}) = arg_{f_i}^{Term_{DFMEA}} maxP(f_i|Term_{DFMEA}) \quad (3)$$

$$= arg_{f_i} max \frac{P(Term_{DFMEA}|f_i)P(f_i)}{P(Term_{DFMEA})} \quad (4)$$

$$= arg_{f_i} maxP(Term_{DFMEA}|f_i)P(f_i) \quad (5)$$

The tagged terms make up our context, say C and we make Naïve Bayes assumption that these terms are independent of each other, which gives us Eq. (6).

$$P(C|f_i) = P = (\{Term_{DFMEA}|Term_{DFMEA}\ in\ C\}|f_i) = \prod_{Term_{DFMEA} \in C} P(f_i|Term_{DFMEA}) \quad (6)$$

The $P(Term_{DFMEA}|f_i)$ and $P(f_i)$ in Eq. (5) is calculated by using Eq. (7).

$$P(Term_{DFMEA}|f_i) = \frac{f(Term_{DFMEA}, f_i)}{f(f_i)} \text{ and } P(f_i) = \frac{f(Term'_{DFMEA}, f_i)}{f(Term'_{DFMEA})} \quad (7)$$

where,

$f(Term_{DFMEA}, f_i)$ = number of co-occurrences of $Term_{DFMEA}$ with f_i;

$f(Term'_{DFMEA}, f_i)$ = number of co-occurrences of $Term'_{DFMEA}$ out of the scope and do not co-occur with f_i;

$f(Term'_{DFMEA})$ = number of co-occurrences of the terms $Term'_{DFMEA}$ out of the scope with respect to the f_i counted in the aggregate data.

In cases where the estimated probabilities, $P(f_i \mid Term_{DFMEA})$ and $P(f_i \mid Term_{instance})$ are greater than 0.92, such $Term_{DFMEA}$ and $Term_{instance}$ are considered to be similar to each other, represented as $Collocate_{DFMEA}$.

3.3. Identification of collocates from the verbatim data

To identify the relevant collocates from the verbatim data, we use $Collocate_{DFMEA}$ (cf. Section 3.2) and they are used to cluster the verbatim data:

Step 1. Let, $V_i = (v_1, v_2, v_3, \ldots, v_j)$ represent a set of verbatim data and $Collocates_{DFMEA}$ represents a set of terms identified from DFMEA.

Step 2. Each verbatim, say v_k is represented in terms of the finite number of phrases, $v_k = (Phrase_1, Phrase_2, \ldots, Phrase_n)$ and each document is represented in terms of their frequency over the aggregate verbatim data, represented by $rv_k = (f_{Phrase1}, f_{Phrase2}, \ldots, f_{Phrasen})$. To reduce the data dimensionality, we only use rv_k to cluster the data. The verbatim data is clustered by using hierarchical agglomerative clustering (Kaufman & Rousseuw, 2005) into a set of clusters, $(C_{Phrase1}, C_{Phrase2}, C_{Phrase3}, \ldots, C_{Phrasen}) \in C$. To cluster the data, the $Phrase_k \in rv_k$ are sorted based on their frequency and in each iteration a repair verbatim with a mention of $Phrase_k$ is assigned to its own cluster. A phrase that is assigned to a cluster is removed from the sorted list. Next, the average pairwise proximity of all the pairs from $C_{Phrasek}$ and $C_{Phrasem}$ is calculated by using the average linkage (Kaufman & Rousseuw, 2005) by using Eq. 8. The two most similar clusters say $C_{Phrasek}$ and $C_{Phrasem}$ are merged and the distance between remaining clusters is updated.

$$D\left(C_{Phrase_k}, C_{Phrase_m}\right) = \frac{1}{N_{C_{Phrasek}} * N_{C_{Phrasem}}} \sum_{i=1}^{N_{C_{Phrasek}}} \sum_{j=1}^{N_{C_{Phrasem}}} d(x_i, y_j) \quad (8)$$

where,

$x_i \in C_{Phrase_k}$ and $y_j \in C_{Phrase_m}$;

$d(x_i, x_j)$ is the distance between objects $x_i \in C_{Phrase_k}$;

$y_j \in C_{Phrase_m}$ from C_{Phrase_k} and C_{Phrase_m};

$N_{C_{Phrasek}}$ and $N_{C_{Phrasem}}$ are the repair verbatim assigned to C_{Phrase_k} and C_{Phrase_m} respectively.

Step 3. Next, from each cluster the instances of symptoms and failure modes are tagged by using the domain ontology, represented as $Collocate_{Verbatim}$.

Now, we have the collocates that are identified from DFMEA, $Collocate_{DFMEA}$ and verbatim, $Collocate_{Verbatim}$. In the next section, we discuss how these collocates are used to calculate the semantic similarity.

3.4. Semantic similarity model to discover new symptoms and failure modes

In our model, the semantic similarity between $Collocate_{DFMEA}$ and $Collocate_{Verbatim}$ is computed to discover new symptoms and failure modes. Initially, we start with the word-based semantic similarity model (Mihalcea, Corley, and Strapparava, 2006), which is shown in Eq. (9).

$$sim^w(T_i, T_j) = \frac{1}{2}\left(\frac{\sum_{w \in T_i}(maxSim(w, T_j))}{\sum_{w \in T_i} idf(w)} + \frac{\sum_{w \in T_j}(maxSim(w, T_i))}{\sum_{w \in T_i} idf(w)}\right) \quad (9)$$

where,

$maxSim(w, T_j)$ is the maximum similarity between a word from T_i, i.e. $w \in T_i$ with all the relevant words from T_j. For example, while comparing two failure

modes a phrase from one failure mode is compared with all the phrases from other failure mode;

$idf(w)$ is the inverse document frequency of a specific word, w.

In (Mihalcea et al., 2006), the text-to-text semantic similarity is calculated between such two text phrases as the ones consists of single words. However, such model provides limiting results in our problem area and we extend Mihalcea et al. (2006) model to handle the text phrases also consisting of multiple word (also referred to as the 'collocates').

The term-to-term, tuple-to-tuple, and text-to-text semantic similarity model is discussed in the next section.

3.4.1. Term-to-term, tuple-to-tuple, and text-to-text similarity model

Step 1. Instead of selecting all the collocates identified from DFMEA and verbatim data, we compute the tf*idf of the collocates, $Collocate_{DFMEA}$ and $Collocate_{Verbatim}$. The ones with their frequency above 0.89 (determined empirically) are used for the semantic similarity comparison.

Step 2. Next, we collect the context information associated with the collocates from the corpus of verbatim. A word window of five unigrams is applied to collect the context information for $Collocate_{DFMEA}$ and $Collocate_{Verbatim}$. For each critical DFMEA collocate (e.g. "window not opening") the relevant components (parts), symptoms, and failure modes are collected from the corpus. From the collected context information, as shown follows the ordered set of tuple pairs are constructed: ($Collocate_{DFMEA}$ $Component_i$), ($Collocate_{DFMEA}$ $Symptom_j$), and ($Collocate_{DFMEA}$ $FailureMode_k$) $\in Tuple_m$.

Step 3. By using the same process described in Steps 1 and 2, we also construct the ordered set of tuple pairs for $Collocate_{Verbatim}$: ($Collocate_{Verbatim}$ $Component_p$), ($Collocate_{Verbatim}$ $Symptom_q$), and ($Collocate_{Verbatim}$ $FailureMode_r$) $\in Tuple_n$.

Step 4. Having constructed two context matrices $Tuple_m$ and $Tuple_n$, first we compute a *term-to-term* semantic similarity by using the terms that are member of these matrices by using Eq. (10).

$$sim(term_i, term_j) = log_2\left(1 + \frac{hits(term_i, term_j)^2}{hits(term_i).hits(term_j)}\right) \quad (10)$$

where,

hits($term_i$) and *hits*($term_j$) as well as *hits*($term_i$, $term_j$)[2] represents the number of times $term_i$ and $term_j$ and the binary tuple($term_i$, $term_j$) appear in the corpus.

Next, a *tuple-to-tuple* semantic similarity is calculated and this time instead of using the terms, our model utilizes the tuples from $Tuple_m$ and $Tuple_n$ to compute their similarity by using Eq. (11).

$$sim(Tuple_i, Tuple_j) = log_2\left(1 + \frac{hits(Tuple_i \& Tuple_j)^2}{hits(Tuple_i).hits(term_j)}\right) \quad (11)$$

where,

hits($Tuple_i$) and *hits*($Tuple_j$) represents the frequency of occurrence of the tuples in the corpus, whereas the *hits*($Tuple_i$ & $Tuple_j$) represents the number of times both $Tuple_i$ and $Tuple_j$ occurs in the documents in a corpus.

Finally, the *term-to-term* and *tuple-to-tuple* semantic similarity scores are combined to produce an aggregate *text-to-text* semantic similarity score (Eq. 12) between the DFMEA collocates, Col_{DFMEA} and the verbatim collocates, $Col_{Verbatim}$.

$$sim(Col_{DFMEA}, Col_{Verbatim}) = \frac{1}{2}\left[\frac{\sum_{Tuple_m \in Col_{DFMEA}}(maxsim(Tuple_m, Col_{Verbatim}).idf(Tuple_m))}{\sum_{Tuple_m \in Col_{DFMEA}} idf(Tuple_m)}\right] + \left[\frac{\sum_{Tuple_n \in Col_{Verbatim}}(maxsim(Tuple_n, Col_{DFMEA}).idf(Tuple_n))}{\sum_{Tuple_n \in Col_{Verbatim}} idf(Tuple_n)}\right] \quad (12)$$

The maxsim($Tuple_m$, $Collocate_{DFMEA}$) function in Eq. (12) is calculated by using Eq. (13) and the calculation of maxsim($Tuple_n$, $Collocate_{Verbatim}$) can be realized on the same lines.

$$maxsim(Tuple_m, Collocat_{DFMEA}) = max_j\{sim(Tuple_m, tuple_j)\}; tuple_j \in Verbatim_j \quad (13)$$

Step 5. The *text-to-text* semantic similarity score between Col_{DFMEA} and $Col_{Verbatim}$ is used to create a rule: **IF** the semantic similarity score between the symptoms or the failure modes in Col_{DFMEA} and $Col_{Verbatim}$ is greater than 0.89 **THEN** they are similar to each other (i.e. synonyms), **ELSE IF** the semantic similarity score is less than 0.89, but greater than or equal to 0.65 **THEN** they are related to each other, **ELSE IF** the semantic similarity score is less than 0.65 **THEN** they are not related to each other. The symptoms and failure modes that are not covered in the DFMEA data are reviewed by the subject matter expert (SME) and they are included in DFMEA.

4. EXPERIMENTS

The proposed approach has been implemented in a prototype tool. For our experiment, we selected the DFMEAs and the verbatim data associated with the following three systems: seatbelt, power window switches, and air induction system.

4.1. Evaluation of collocate identification method

In this experiment, we evaluated performance of the collocate identification method (cf. Sections 3.2 and 3.3). To conduct this experiment, we randomly selected 3830 data points

[2] It is important to remember that while computing $sim(term_i, term_j)$ and $sim(Tuple_i, Tuple_j)$ we populate these functions with ($component_i$, $component_j$), ($symptom_i$, $symptom_j$), and ($failure\ mode_i$, $failure\ mode_j$) that are member of $Tuple_m$ and $Tuple_n$ to compute their similarity.

related to three systems, seatbelt (system A), 3000 data points related to power window switches (system B), and 1313 data points related to the air induction system (system C). To avoid an inexact and biased comparison, the experimental data obeyed identical and independent distribution. The collocate identification algorithm was applied to identify candidate collocates and the final set of collocates discovered by our algorithm were presented to the SMEs for their manual certification. The results inspected by the SMEs were used to calculate Precision, Recall, and F1-score and they are shown in Figure 3.

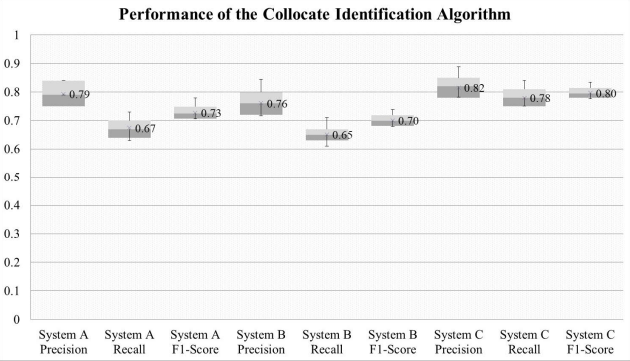

Figure 3. The result of collocate identification algorithm.

The recall rate in all the three systems were comparatively lower as compared to the precision rate. In other words, while the algorithm identified a smaller number of collocates from the data, it showed a high precision rate (i.e. low false positive rate) within newly discovered collocates. Having a high precision rate was one of the key requirements from the tool users because all newly discovered collocates were used in semantic similarity calculations. And ultimately the similar ones were used to augment DFMEAs and it was necessary to ensure that the collocates used for augmenting DFMEA were accurate. Moreover, the data generation mechanism was not in our control since the data was collected from different sources and it was captured by the different stakeholders. Given these multiple constraints, the average precision of 0.79, the recall of 0.7, and the F1-score of 0.74 was satisfactory in our business.

4.2. Evaluation of semantic similarity model

Here, we describe performance evaluation of the semantic similarity model discussed in Section 3.4 as it was used to discover new symptoms and failure modes from the verbatim data. The verbatim data was pre-processed to get rid of the special characters, additional white spaces, and run-on-words. The collocates were identified from the processed data by using the algorithms discussed in Sections 3.2 and 3.3. Again, to avoid an inexact and biased comparison the data followed identical and independent distribution. All the collocates identified from the DFMEA and verbatim data were used to computed *text-to-text* similarity. The results achieved by our model were presented to the SMEs for their evaluation. By using the results inspected by the SMEs the precision, recall, and F1-scores were calculated by using the equations (14), (15), and (16) respectively.

$$Precision = \frac{True\ Positives}{(True\ Positives + False\ Positives)} \quad (14)$$

$$Recall = \frac{True\ Positives}{(True\ Positives + False\ Negatives)} \quad (15)$$

$$F - Measure = \frac{2(Precision * Recall)}{(Precision + Recall)} \quad (16)$$

where,

True positives = the correct symptoms and failure modes correctly identified by the model.
False positives = the correct symptoms and failure modes incorrectly rejected by the model.
True negatives = the incorrect symptoms and failure modes correctly rejected by the model.
False negatives = the incorrect symptoms and failure modes not rejected by the model.

The results of this experiment are summarized in Table 1.

Table 1. The semantic similarity experiment results for all the three systems.

System	Precision New Symptoms	Precision Synonym Symptoms	Precision New Failure Modes	Precision Synonym Failure Modes
Seat Belt	0.83	0.79	0.89	0.67
Power Window Switches	0.87	0.92	0.76	0.82
Air Induction System	0.71	0.82	0.67	0.73
System	**Recall New Symptoms**	**Recall Synonym Symptoms**	**Recall New Failure Modes**	**Recall Synonym Failure Modes**
Seat Belt	0.66	0.72	0.74	0.67
Power Window Switches	0.62	0.74	0.65	0.6
Air Induction System	0.89	0.72	0.63	0.65
System	**F1 New Symptoms**	**F1 Synonym Symptoms**	**F1 New Failure Modes**	**F1 Synonym Failure Modes**
Seat Belt	0.74	0.75	0.81	0.67
Power Window Switches	0.72	0.82	0.70	0.69
Air Induction System	0.79	0.77	0.65	0.69

Figure 4 shows the new and synonymous symptoms and failure modes discovered by our model from the data related to all the three systems.

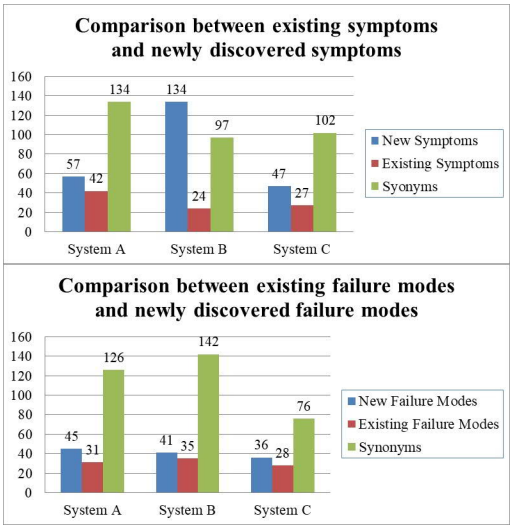

Figure 4. New and synonym symptoms and failure modes discovered by the semantic similarity model.

Finally, the newly discovered symptoms and failure modes were used in the testability and diagnosability metrics, such as *fault detection* and *fault isolation*. In both the metrices, we evaluated the fault detection and isolation rate before and after employing our proposed approach.

The *fault detection* (P_D) is the percent of faults detected by the new symptoms by observing new failure modes of a system. It was used to determine the fault coverage and assess undetected faults to determine acceptability. The P_D is computed by using Eq. (17).

$$P_D = \frac{[\sum_{i=1}^{m} D_i]}{m} \quad (17)$$

where, $D_i = 1$ if the fault i is detectable, the detection event $D_i = 1$, D_i has at least one non-zero element and m been number of associations between symptoms and failure modes. Figure 5 shows the comparative analysis of the fault detection rate. On average, the fault detection rate before employing our model was 51.6%, which improved significantly to 86.6% after employing our model. The low fault detection rate before using our proposed approach was due to the two key reasons: the use of inconsistent vocabulary and data incompleteness, which limited new knowledge discovery from the industrial scale data.

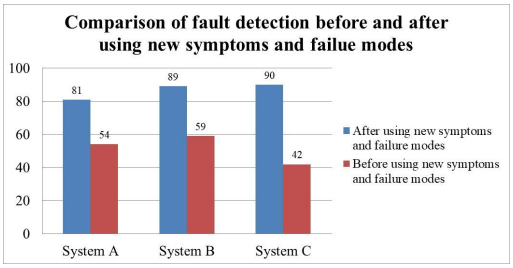

Figure 5. Improvement in the fault detection rate before and after using new symptoms and failure modes discovered by the proposed model.

The *fault isolation* (F_I) is the probability that newly discovered symptoms uniquely isolate system faults for the new failure modes discovered to be associated with a system. The percent fault isolation is computed as the percent of total faults that can be uniquely isolated using Eq. (18) for unweighted case.

$$F_I = \frac{[\sum_{i=1}^{m} ISO_i]}{m} \quad (18)$$

where, $ISO_i = 1$ if fault i is uniquely isolatable, 0 otherwise.

Figure 6 shows the improvement in the fault isolation rate after using newly discovered symptoms and failure modes by our model. In case of air induction system, the fault isolation rate was relatively lower, i.e. 88.79%. The closer analysis of the verbatim data related to air induction system revealed that not all the tests conducted by the field technicians were recorded in the verbatim database. Some of the test results were non-textual in nature and they were not included in this experiment. As a result, our algorithm only had a partial exposure to all the tests conducted in the field.

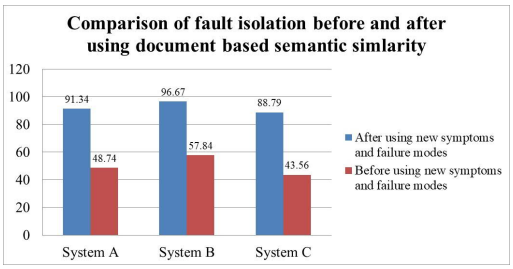

Figure 6. Improvement in the fault isolation rate before and after using new symptoms and failure modes discovered by the proposed model.

5. CONCLUSION

In this paper, a novel approach is proposed to automatically compare two heterogeneous unstructured text data sources, such as the DFMEA and field verbatim data to discover new symptoms and failure modes for in-time augmentation of DFMEAs. There is a limited effort in the existing state-of-the-art to automatically augment DFMEAs and our proposed model bridges this gap by making successful and practical proposal. In our approach, initially a domain ontology is used to identify key collocates from the DFMEA and verbatim data. While identifying the key collocates, we successfully tackle the vocabulary mismatch problem related to the two heterogenous data sources. In our hierarchical sematic similarity model, initially *term-to-term* and *tuple-to-tuple* semantic similarities are calculated. Our model handles multi-term phrases while calculating *tuple-to-tuple* semantic similarity. The two semantic similarity scores are uniquely combined to calculate final *text-to-text* semantic similarity. The performance of the prototype tool is validated by using three real-life systems – seatbelt, power window switch, and air induction. Our model yield 0.79, 0.7, and 0.74 precision, recall, and F1-score respectively. More importantly, the new symptoms and failure modes are discovered from all the three systems, i.e. seatbelt (57 symptoms and 45 failure modes), power window switch (134 symptoms and 41 failure modes), and air induction system (37 symptoms and 36 failure modes) using the proposed model. These new constructs are used to compute the fault detection and fault isolation rates. On average, the fault detection rate improved from 51.6% to 86.6%, whereas on an average the fault isolation rate improved from 50.0% to 92.3% after using newly discovered symptoms and failure modes. In summary, ours is a practical approach that can be used for the in-time augmentation of the DFMEA data to improve product quality.

ACKNOWLEDGEMENT

The authors would like to thank GM's internal review committee and Dr. Jonathan Owen for providing their feedback and supporting this work.

REFERENCES

Abdallah, A., Feron, E. M., Hellestrand, G., Koopman, P., & Wolf, M. (2010). Hardware/Software Codesign of Aerospace and Automotive Systems. *Proceedings of the IEEE*, Vol. 98, no. 4, pp. 584-602

Atamer, A. (2004). Comparison of FMEA and Field-Experience for a Turbofan Engine with Application to Case-Based Reasoning. *Proceedings of the IEEE Aerospace Conference*, vol. 5, pp. 3354-3360

Benedittini, O., Baines, T. S., Lightfoot, H. W., & Greenough, R. M. (2009). State-of-the-art in Integrated Vehicle Health Management. *Journal of Aerospace Engineering*, vol. 223, no. 2, pp. 157-170

Berger, A. & Lafferty, J. (1999). Information retrieval as statistical translation. *Proceedings of the 22nd Annual international ACM SIGIR conference on research and development in information retrieval*, New York, USA, pp. 222-229

Bullinaria, J., & Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, vol. 39, pp. 510–526

Carlson, C. S. (2012). Lessons Learned for Effective FMEAs. *IEEE Proceedings of the Reliability and Maintainability Symposium*, pp. 280-285

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of American Society for Information Science*, vol. 41, no. 6, pp. 391-407

Fausto, G., Maurizio, M., & Ilya. Z. (2007). Encoding Classifications into Lightweight Ontologies. *Journal of Data Semantics*, vol. 8, pp. 57-81

Griffiths, T., Steyvers, M., & Tenenbaum, J. (2007). Topics in semantic representation. *Psychological Review*, vol. 114, pp. 211–244

Hearst, T., (1999). Untangling Text Data Mining. *Uni. of Maryland*, pp. 3-10

Islam, A., & Inkpen, D. (2008). Semantic Text Similarity using Corpus-Based Word Similarity and String Similarity. *ACM Transactions of Knowledge Discovery from Data*, vol. 2, no. 2, pp. 10:1-10:25

Kaufman, L., and P. Rousseeuw. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis*. New York, NY: John Wiley

Krovetz, R. (1993). Viewing morphology as an inference process. Proceedings of the 16th ACM SIGIR Conference, R. Korfhage et al. (Ed.), Pittsburgh, pp. 191-202

Landauer, T., & Susan D. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, vol. 104, no. 2, pp. 211–240

Lavrekno, V., & Croft, W. B. (2001). Relevance based language model. Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, New York, USA, pp. 120-127

Lenci, A. (2008). Distributional approaches in linguistic and cognitive research. *Italian Journal of Linguistics*, vol. 20, no. 1, pp. 1–31

Lund, K., & Curt, B. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, vol. 28, pp. 203-208

Medin D. L., Goldstone, R. L., & Gentner, D. (1990). Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. *Psychological Science*, vol. 1, pp. 64-69

Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. Proceedings. of the American Association for Artificial Intelligence (AAAI 2006), Boston, vol. 6, pp. 775-780

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, vol. 38, no. 11, pp. 39-41

Padó, S., & Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, vol. 33, no. 2, pp. 161–199

Papadopoulous, Y., Parker, D. & Grante, C. (2004). A Method and Tool Support for Model-based Semi-automated Failure Modes and Effects Analysis of Engineering Designs. *Proceedings of the 9th Australian Workshop on Safety Critical Systems and Software*, vol. 47, pp. 89-95

Price, C. & Taylor, N. (1997). Multiple Fault Diagnosis from FMEA. *Proceedings of AAAI-97/IM-97*, pp. 1052- 1057, Providence, RI

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and Application of a Metric on Semantic Net. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 1, pp. 17-30

Rajpathak, D., Chougule, R. & Bandyopadhyay, P. (2010). A Domain Specific Decision Support System for Knowledge Discovery Using Association and Text Mining. *International Journal of Knowledge and Information System*, vol. 31, no. 3, pp. 405-432.

Rajpathak, D. (2013). An Ontology Based Text Mining System for Knowledge Discovery from the Diagnosis Data in the Automotive Domain. *International Journal of Computers in Industry*, vol. 64, no. 5, pp. 565-580

Shekarpour, S., Marx, E., Auer, S., & Sheth, A. P. (2017). RQUERY: Rewriting Natural Language Queries on Knowledge Graphs to Alleviate the Vocabulary Mismatch Problem. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pp. 3936-3943

Spärck Jones, K. (1972). A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, vol. 28, pp. 11–21

Sutrisno, A. & Lee, T. J. (2011). Service reliability assessment using failure mode and effective analysis (FMEA): survey and opportunity roadmap. *International Journal of Engineering, Science and Technology*, vol. 3, no. 7, pp. p. 25-38

Tso, K. S., Tai, A. T., Chau, S. N., & Alkalai, L., (2005). On automating failure mode analysis and enhancing its integrity. *Proceedings. of the 1st IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2005)*, Changsha, Hunan, China, 12-14 December, pp. 287-194

Turney, P. D., (2001). Mining Web for synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, Freiburg, Germany, pp. 491-502

Turney, P., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, vol. 37, pp. 141–188

Wirth, B., Kramer, A., & Peter, G. (1996). Knowledge Based Support of System Analysis for Failure Mode and Effects Analysis. *Engineering Applications in Artificial Intelligence*, vol. 9, no. 3, pp. 219-229

Wu, Z., and Palmer, M. (1994). Verb semantics and lexical selection. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 133–138