IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 9 May 2024, accepted 31 May 2024, date of publication 11 June 2024, date of current version 21 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3412929

RESEARCH ARTICLE

A Model for Quantifying the Degree of Understanding in Cross-Domain M2M Semantic Communications

RICKY YUEN-TAN HOU, (Senior Member, IEEE), GANG LIU, JEFFERSON FONG, HUI ZHANG, (Senior Member, IEEE), AND SEON-PHIL JEONG
Guangdong Provincial Key Laboratory IRADS, BNU-HKBU United International College, Zhuhai 519087, China
Department of Computer Science, BNU-HKBU United International College, Zhuhai 519087, China

Corresponding author: Ricky Yuen-Tan Hou (rickyhou.hk@gmail.com)

This work was supported by Guangdong Provincial Key Laboratory of ''Interdisciplinary Research and Application for Data Science'' (IRADS) under Grant 2022B1212010006 and Grant R0400001-22.

ABSTRACT This paper addresses the problem of semantic communications (SemComs) in intelligent machine-to-machine (M2M) applications. Although M2M applications may employ other languages as the communication medium, natural languages are commonly used as the medium between machines and robots. One favorable characteristic of using natural languages is that it allows humans to inspect communication contents easily, which caters to the needs of security and quality of service for M2M communication. Currently, no exact solutions are available for quantifying and measuring the understanding of M2M communication. This paper identifies three specific challenges in the field: inconsistent knowledge base (KB), cross-domain interpretation, and a measure for understanding the meaning of messages. We propose a model to address these challenges in two steps. First, we propose an evidence-based shared-KB communication model for cross-domain meaning interpretation using Dewey Decimal Classification. Second, we propose a measure to quantify the understanding level through a two-stage validation between the sender and receiver. Real-life datasets and numerical experiments are used to evaluate the model's performance. The results show that the degree of understanding (DoU) can be successfully measured by observing the performance of the sender and receiver under the same conditions. The proposed method can effectively improve mutual understanding between the two machines.

INDEX TERMS Cross-domain interpretation, degree of understanding, machine-to-machine, semantic communications, two-stage validation.

I. INTRODUCTION

With advanced technologies like the Internet of Things and artificial intelligence, communication applications go beyond the classic communication theory. Decades ago, Weaver [1] pointed out that communication consists of three levels of goals. The first is on bit transmission. The other two are the meaning interpretation of transmitted symbols and the effectiveness of semantic communications (SemComs). The latter two are the next steps in classic communications [2]. SemCom concerns the process of transmitting the meaning of messages from a sender to its receiver [3], [4], [5], [6], which could be conducted between different parties, such as Human-to-Human (H2H), Machine-to-Human (M2H), and Machine-to-Machine (M2M). M2H communication conveys the meanings between humans and machines. On the other hand, M2M communication concerns the interactions among machines for effective task execution [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Tang.

A. TYPICAL SEMANTIC COMMUINCATION SYSTEM

The typical SemCom system is shown in Fig. 1 [3], [7], [8]. A sender transmits messages to a receiver. At first, the sender encodes the messages at the semantic- and bit- levels to combat semantic or physical noise over a channel. A KB is usually included in the process of semantic encoding. Then,

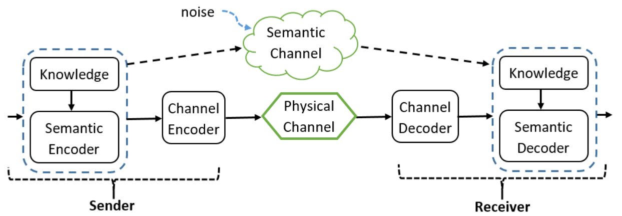

FIGURE 1. Main components of a typical SemCom system.

the encoded signal propagates from the sender to the receiver via a physical channel. The receiver decodes the received signals at both the bit- and semantic- levels for reconstructing the transmitted message. A human being or intelligent machine consumes the messages and executes the task accordingly. In the existing models, text is the mainstream data source supported in SemComs. However, image and speech can also be supported [4], [7].

There are two types of channels in SemCom systems. The first is the physical channel, which deals with data transmission at the bit level. The second is the semantic channel, in which semantic noise needs to be addressed during transmission because it might cause interpretation errors or misunderstandings.

Compared to traditional communication, SemCom aims to maximize the expected faithfulness in representing the observed worlds and minimize the amount of data transmitted [7]. As a result, the sender and receiver must perform various highly intelligent algorithms [3], [4] in addition to the functions of traditional communication terminals.

A SemCom system is a knowledge-based system [4]. Both the sender and receiver rely on a separated or shared background KB for the meaning interpretation of messages [3], [4], [6], [9]. Depending on the types of KB, the performance metrics used to measure SemCom systems are diverse, such as semantic error based [4], [8], age of information (AoI), and value of information [8], [9], [10].

Numerous information representations are available for semantic communication, such as natural languages, propositional logic, graphic-based, and neural-network-based languages [11], [12]. In particular, neural network-based languages incorporate semantic information into deep networks by training networks to set appropriate connection weights. In addition to the weight assignment, choosing an appropriate network structure and size is also critical [13].

B. MAIN APPLICATIONS OF SEMANTIC COMMUNICATONS

Quantifying semantic interpretation is required in many applications, such as the Internet of Things (IoT) [4], [8], artificial intelligence (AI) [4], [8], robotics [8], the Web [9], [14], extended reality [8], and big data [15]. We highlight some of the major applications as follows.

1) AI AND IOT

The information consumption of intelligent machines and AI robots is growing unprecedentedly. A simple and general solution is needed for quick semantic information consumption and processing [16]. Semantic knowledge reasoning framework can provide these applications with more automated data quality management [17]. Besides textual-data applications, SemCom is also demanded by non-text and entangled-data applications. For example, to improve communication reliability in the aviation industry, the semantic meaning of radiotelephony communication is extracted and represented with a semantic vector, which is then used to verify the semantic consistency [18]. In visual applications, images are encoded with the GAN-based coding method to allow semantic exchange [19]; a semantic enhanced encoder-decoder framework [20] is proposed to recognize low-quality scene texts in images robustly.

2) BIG DATA

Achieving big data analytics' full potential requires reconciling data distribution and modeling principles. Some attempts have been made. For example, the semantic-empowered communication paradigm (SCP) is proposed to address the problem. Uysal et al. [21] pointed out that a significant performance improvement can be made with simple modification of SCP for the AoI applications. Kountouris and Pappas [22] used SCP to significantly reduce reconstruction errors and cost of actuation errors in data processing.

3) SEMANTIC WEB

Semantic web technology enhances machines' ability to self-manage data in different aspects [14], [23]. However, new methods are required to address the issue of meaning interpretation in changing environments when retrieving data from various sources [24].

C. CONTRIBUTIONS AND PAPER OUTLINE

Our paper aims to address the problem of Cross-domain M2M SemCom, and the contributions can be summarized as follows:

- We identify three specific challenges in the field: inconsistent KB, cross-domain interpretation, and a measure for understanding the meaning of messages.
- We propose a shared-KB communication model for cross-domain meaning interpretation using Dewey Decimal Classification. Our proposed model differs from the existing solutions: (1) It embeds semantic information in the message to enable accurate cross-domain interpretation. (2) It supports evidence-based communication by measuring the DoU at both ends. (3) A unified reasoning framework is adopted to minimize the semantic errors in the process.
- We propose a measure to quantify the DoU through a two-stage validation between the sender and receiver, differentiating from the existing methods: it is a two-level hierarchical approach for narrowing the search space effectively during reasoning.

To the best of our knowledge, the proposed model is the first attempt to explore the application of an existing library knowledge management system in domain classification. Regarding the characteristics and benefits of the proposed model, we highlight the following.

- This model uses a feedback-based and evidence-based method to improve the mutual understanding of the sender and receiver iteratively in M2M communications, where decisions are made, and actions or activities are undertaken deriving from objective evidence.
- The model can quantify the DoU in sentence-based communications.
- The two-stage method can significantly reduce the search space.
- The model minimizes the cumulative errors of understanding in M2M communications.
- The model can be extended to other communication contexts, such as M2H and H2M applications.

The remaining parts of this paper are organized as follows. Section II formulates the problem. Section III highlights related work and research challenges. Section IV describes the design of a universal model for cross-domain SemComs. Section V explains the steps to convey meaning from the sender to the receiver at the word level. Section VI proposes a method to measure the DoU at the sentence level (SDoU). Section VII gives a solution for optimizing the objective function. Section VIII evaluates the performance of the proposed method. Section IX concludes this paper.

II. PROBLEM FORMULATION

This study adopts natural languages as the communication medium among machines and robots. A more detailed discussion of the language used is given in the Related Work section. The goal is to minimize the semantic understanding difference between the sender and receiver, and quantify the mutual understanding of transmitted messages. To quantify the DoU, we propose a hierarchical approach in which a message is separated into sentences and words. The mathematical formulation is as follows.

We consider a message consisting of e sentences, $T = \{T^1, T^2, \cdots T^i, \cdots T^e\}$, and its corresponding meaning $M = \{M^1, M^2, \cdots M^i, \cdots M^e\}$. A sentence T^i consists of $|T^i|$ words or tokens (for simplicity of expression, we use the term "word" in this paper), such that $T^i = \left\{t_1^i, t_2^i, \cdots t_j^i, \cdots t_{|T^i|}^i\right\}$. The meaning of T^i is derived from a set of its word meanings, $M^i = \left\{m_1^i, m_2^i, \cdots m_j^i, \cdots m_{|T^i|}^i\right\}$, where m_j^i is the original meaning of word j in sentence i. During the transmission, suppose the interpretation of T^i by the sender and the receiver are M^{is} and M^{ir}, respectively. Intuitively, senders only send messages they fully understand. We assume $\left|M^i - M^{is}\right| \leq \Delta$, where sentence meanings are represented as vectors [25], and $\Delta \cong 0$ is negligible. In the literature, various sentence models can compute a representation vector for a given sentence [26], [27]. The meaning difference Δ is a user-defined parameter, which should be customized in applications. Table 1 is a list of major notations in this paper.

TABLE 1. List of notations.

Variables	Meanings
T	Message T
M	Meaning of message T
T^i	Sentence i of message T
M^i	Meaning of sentence i
t_j^i	Word j of sentence i
m_j^i	Meaning of word j in sentence i
M^{is}	Interpretation of sentence i at the sender
M^{ir}	Interpretation of sentence i at the receiver
T_j^i	Paraphrased version j for sentence i
T^{is}	Best paraphrased version for sentence i at the sender
T^{ir}	Best paraphrased version for sentence i at the receiver
T_0^i	Original version for sentence i

We formulate the quantification of DoU as a two-stage stochastic programming problem. The goal, denoted by F, is to minimize the misunderstanding between the sender and the receiver, which can be expressed mathematically as

$$F = min\, f(T) + g(T), \tag{1}$$

subject to

$$0 \leq f(T) \leq 1, \tag{2a}$$
$$0 \leq g(T) \leq 1, \tag{2b}$$

where the sub-functions $f(T)$ and $g(T)$ can be seen as the average of the misunderstanding of the sentences in message T: $f(T) = \frac{1}{e}\sum_{T^i \in T} f\left(T^i\right)$ and $g(T) = \frac{1}{e}\sum_{T^i \in T} g(T^i)$, and their values range from 0 to 1.

$f\left(T^i\right)$ and $g(T^i)$ are the misunderstandings of sentence T^i at the word and sentence levels, which can be further broken down. Let *sim*() be a function calculating the distance between two representation vectors of the sentences [26], [28]. Denote $sim_w\left(t_j^{is}, t_j^{ir}\right)$ as the meaning similarity for word t_j^i between the sender and the receiver. Similarly, let $sim_s\left(T^{is}, T^{ir}\right)$ be the meaning similarity for sentence T^i. Then quantifying the misunderstanding at the word and sentence levels can be expressed as follows

$$f\left(T^i\right) = 1 - \sum_{t_j^{is}, t_j^{ir} \in T^i} sim_w\left(t_j^{is}, t_j^{ir}\right), \tag{3a}$$
$$g\left(T^i\right) = 1 - sim_s\left(T^{is}, T^{ir}\right). \tag{3b}$$

The full expression of (1) is written as:

$$F = min\frac{1}{e}\left(\sum_{T^i \in T}\left(1 - \sum_{t_j^{is}, t_j^{ir} \in T^i} sim_w\left(t_j^{is}, t_j^{ir}\right)\right)\right) + \frac{1}{e}\sum_{T^i \in T}\left(1 - sim_s\left(T^{is}, T^{ir}\right)\right), \tag{4}$$

subject to

$$0 \leq sim_w \left(t_j^{is}, t_j^{ir}\right) \leq 1, \tag{5a}$$

$$0 \leq sim_s \left(T^{is}, T^{ir}\right) \leq 1, \tag{5b}$$

$$e, i, j \geq 1, e, i, j \in N, \tag{5c}$$

$$i \leq e. \tag{5d}$$

The challenge of this problem is to design suitable methods to quantify the two similarity scoring functions and to develop a method to find the optimal solution for the objective function efficiently.

III. RELATED WORK AND RESEARCH CHALLENGES

In this section, we highlight and discuss recent works closely related to our research in different aspects. Then, we identify three research challenges that will be addressed in the paper.

A. SYSTEM DESIGN

Weaver, Carnap, and Bar-Hillel were among the first to introduce the concept of SemComs [1], [29]. A shared knowledge framework is usually proposed to facilitate universal communication [4], [29]. Some papers use the term background knowledge instead of shared knowledge [7]. Such a framework captures changes in the real world and formalizes them into common knowledge. Along with such a concept, several existing works also proposed connecting the real world with a central database of the word or sentence level, such as WordNet [30].

Qin et al. [7] highlighted the major challenges for the system design of SemCom, such as system components, semantic noise, and performance metrics. Lan et al. [3] discussed the approach for designing SemCom systems based on knowledge graphs. SemCom integrates users, application requirements, and the meaning of information into data processing and transmission. It is predicted to be the core paradigm in 6G networks [8].

B. CROSS-DOMAIN INTEROPERABILITY

Cross-domain interpretation is another open issue for SemCom. In the most challenging applications, the context's domain is unspecified. Existing solutions for domain classification are usually based on concept detection [31], [32], [33]. The difficulty is that the real world is vast and heterogeneous, and the sender and receiver must share the same knowledge domains. Classifying knowledge accurately into domains is extremely difficult [4]. Some attempts have been made in the past few years. For instance, Zheng [15] defined new methods of cross-domain data fusion, which focus on knowledge fusion rather than traditional scheme mapping and data merging. Lakka et al. [17] proposed using semantic interoperability mechanisms to enable semantic cross-domain interactions among different systems. However, existing works are still limited in addressing this particular issue.

C. NATURAL LANGUAGES VS. CODED LANGUAGES

Coded languages have clear and concise rules that determine how computers should interpret and execute the symbols [34]. However, the rules of natural languages, on the other hand, depend on the context and the speaker's intention, which change dynamically. Natural languages have an almost infinitive and dynamic set of words, rules, and conventions that can be used to create texts.

Some researchers suggested that semantic languages should be developed for better automatic processing by allowing semantic languages to mimic natural languages and be less focused on syntax and pragmatics [35]. However, in many situations, communications aim to exchange natural languages [7]. Natural language is still the mainstream working language for robots or autonomous machines. The reasons are as follows.

Currently, the world is managed by humans, who develop and maintain natural languages to describe the world and use them to communicate with each other. In this perspective, natural languages are the best mediums for world representation. Secondly, humans must regularly inspect communication contents among robots for security concerns. In this regard, the selected communication languages should be easily understandable by humans. In the long run, semantic languages should be developed. However, it is challenging to require every human to learn semantic languages. New technologies may offer a solution to bridge the gap by automatically translating the neural signals of humans into semantic languages in the future.

D. SEMANTIC METRICS AND SENTENCE SIMILARITY MEASUREMENT

When there is a need to ensure text alignment among communication parties, a high-level similarity measurement is usually performed between the receiver and the sender [36]. Semantic metrics are measures to compare the semantic similarity between text corpora, such as computing the similarity and dissimilarity among their ontological entities.

The metric aims to provide a broad representation of distance across specific linguistic aspects, such as syntactic or semantic [37]. Text semantic metrics depend on the form of the text representation, such as token-based or vector embedding, that can be distinguished into three aspects: lexicographical statistics, distributional metrics, and discriminability metrics [37].

Recently, Getu et al. [38] analyzed the performance of semantic metrics for text quality assessment. The quality can be evaluated at the word level using semantic distance or word error rate. At the sentence level, the quality is evaluated by variant semantic metrics, such as semantic similarity metrics (SSMs), machine-translation-based metrics (Bilingual Evaluation Understudy and Metric for Evaluation of Translation with Explicit Ordering), and consensus-based image description evaluation. Moreover, Kour et al. [37] proposed seven criteria to evaluate the robustness of metric measures.

Recent studies have shown that integrated metric approaches can improve the accuracy of sentence semantic similarity [25], [27], [28], [39], [40]. These methods consider both semantics and syntax (in terms of word order). Let two input texts be represented in two vectors [26], [28], and the symbol $sim_s\left(T^i, T^j\right) \in [0, 1]$ represents the similarity measurement between two sentences (T^i, T^j). Then, the overall sentence similarity between T^i and T^j is defined as a combination of semantic similarity and word order similarity with weight δ [27], [40]. Some researchers suggest that the weight δ should be chosen in the range of $(0.5, 1]$ [40].

$$sim_s\left(T^i, T^j\right) = \delta sim_{se} + (1 - \delta) sim_{sy}, \tag{6}$$

where sim_{se} and sim_{sy} are the similarity measurements for semantic and syntactic, respectively.

E. SENTENCE GENERATION AND PARAPHRASING

Sentence generation and paraphrasing are two essential requirements for M2M SemCom. Sentence generation [41], on the one hand, is a computational process of automatically generating sentences in certain human languages based on specific communicative intentions. In most current approaches for sentence generation, the input is either a logical form or a set of facts in some knowledge representation language [41]. On the other hand, paraphrasing is a restatement of a text, paragraph, or work, giving meaning in another form. There are different approaches for generating sentences or paraphrases, such as the VAE model [42], the encoder-decoder method [43], [44], and other methods [45], [46].

F. RESEARCH CHALLENGES

Solutions to SemCom have remained largely unexplored. Due to various challenges, existing works are limited only to addressing some fundamental problems in communications. In the literature, many researchers have highlighted several common challenges in SemComs. For example, Luo et al. [4] summarized that insufficient theoretical research, inconsistent KBs, multi-user interpretation, and implementation are open issues for future investigations. Moreover, Yang et al. [8] pointed out that further explorations are needed into system effectiveness, sustainability, and trustworthiness for SemCom in 6G.

In this paper, we focus on three fundamental problems in M2M communications: (P1) Why is interpretability so challenging in SemCom? (P2) What is the ideal framework for cross-domain interpretation? (P3) Is it possible to measure the DoU among the communication parties?

1) EVER-CHANGING WORLD (P1)

Interpretability in SemCom is so challenging because the world is changing. Languages are the mediums of communication, and understanding the meaning behind words is a basic requirement for humans. However, every language evolves continuously. The meaning of the same word may change over time. For instance, before the invention of AI, the phrase ‘‘neural network’’ was primarily used in the biology domain. The rate of change in each language may vary considerably due to internal and external factors [47]. In fact, the English language has changed greatly since Old English, and about 4,000 new words are added to the dictionary each year [48].

The rapid pace of technological advancements further compounds the challenge of interpretability in SemCom. The real world, including its objects and phenomena, is far from static. Cultures, for instance, evolve and change continuously in terms of their products, practices, and values [49]. Technological innovations, from smartphones to social media, have revolutionized our society and daily lives in the past two decades [50]. New words and terminologies are invented to accommodate these changes every year. Therefore, learning the changes in the real world's representation regarding the languages we use and the new things or objects we describe is a challenge.

2) CROSS-DOMAIN INTERPRETATION (P2)

One major challenge of SemCom is the structural heterogeneity of cross-domain applications. There are several definitions for a domain, but in this context, a domain is defined as the area of knowledge or activity the system covers in an application (e.g., healthcare management and stock forecasting are two different application domains) [51].

M2M communications consist of largely distributed, autonomous, and diverse machines in different applications or domains. Existing solutions can only infer the meaning of a specific message within a limited context domain [52]. Due to the lack of core knowledge shared by machines, they cannot correlate information across domains like humans [4].

3) MEASURE THE DOU (P3)

Understanding is different from learning facts. It is the ability to form an opinion or reach a conclusion through reasoning and information, and actions can be taken accordingly [53]. Different receivers may have different DoUs of the same message, which might cause unexpected actions to be executed. Measuring the DoU can ensure the correctness of task execution. To our knowledge, existing works do not offer such solutions.

IV. EVIDENCE-BASED CROSS-DOMAIN SEMCOM MODEL

A KB is a conceptual presentation of the real world. Intelligent machines rely on KBs to understand the world. It is necessary to constantly learn about the changes in the world, in which factual knowledge will form a common KB for reasoning. In this section, Sub-section A gives an overview of the communication model. Sub-sections B and C introduce the concepts of the knowledge domain. Sub-section D highlights the main idea of the word semantic database.

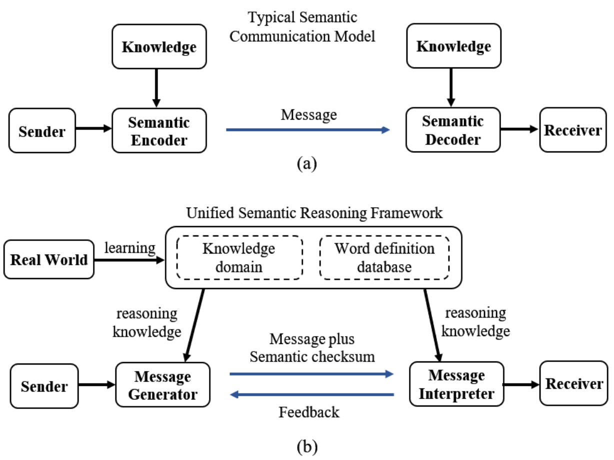

FIGURE 2. **(a) Typical SemCom Model; (b) Proposed SemCom Model.**

A. GENERAL MODEL

The real world is highly heterogeneous, and it may not always be possible to map all the aspects of context into a single formalism. One way to deal with this is to build up a common framework to restrict all semantic definitions to be expressed in a particular formalism [52].

In general, there are three critical steps in relation to understanding a message in natural languages. Both communication parties need to (1) specify the knowledge domain for the message in order to reduce semantic ambiguity [54]; (2) figure out each word's meaning, which is the basic unit of a sentence, and (3) interpret the meaning of each sentence as a whole.

Figure 2 illustrates a model of SemCom proposed in this paper in comparison with the typical model. In the model, knowledge reasoning relies on a unified semantic reasoning framework (USRF), which has two modules: knowledge domain (KD) and word semantic database (WSD). KD is used to specify a sentence's domain, while WSD is used to look up the semantic meanings of each word in the sentence. USRF is fully shared by all the communication parties. A typical SemCom session consists of six main steps:

(1) The sender first breaks down a message into sentences. The communication is on a sentence basis.
(2) The sender sends a sentence to the message generator (MG) for reasoning.
(3) MG specifies the knowledge domain using DDC, looks up the proper meaning for each word, and then produces a semantic checksum for the sentence.
(4) MG sends the encoded sentence to the receiver's message interpreter (MI).
(5) MI decodes the message using USRF and gives feedback to MG. MG and MI iteratively improve the DoU based on the feedback.
(6) Upon achieving a good DoU, MI passes the sentence together with its meaning to the receiver.

The sender and the receiver share the USRF in support of cross-domain interpretation. Based on the iterative feedback, the DoU between two parties can be measured and quantified. The measuring of understanding (MoU) is initiated and evaluated by the sender, who is responsible for ensuring the faithful transmission of meaning for the sentence.

Usually, the KD can be specified once in each communication session. To improve efficiency, the paraphrasing and MoU can be conducted only in the critical sentences of the communication session. According to Weaver [1], the efficiency of SemCom is the third level of communication, which is our future research direction.

Although the proposed communication model is designed for textual data, the semantic features of images [19] and speeches [18] can be converted into text-based descriptors in some applications. However, the conversion of forms might introduce semantic noises. Nevertheless, the proposed model can be scaled up to accommodate a broader scope of data and information by splitting the description of an object into layers and adding a feedback loop to reduce semantic errors.

B. KNOWLEDGE REPRESENTATION AND REASONING

Knowledge representation and reasoning are two essential elements in SemCom systems. To represent knowledge, formal languages, such as rule-based and decision trees, are used [11].

Knowledge reasoning is used to infer facts from existing data. Many methods have already been proposed for reasoning with knowledge. The most common approaches are probability-based, logic-based, and case-based [55].

C. SPECIFYING KNOWLEDGE DOMAIN

The knowledge domain refers to the descriptions of objects, actions, functions, and other instances. It defines possible classes and/or instances of things in the world. An object or instance may belong to multiple domains. The challenge of cross-domain interpretation is resolving the ambiguity of domain-dependent word semantics. For instance, the word ''bank'' has several meanings, such as a commercial bank or a river bank. Also, the semantics in each domain evolve with time. Specifying the domain for a sentence can greatly reduce the search space in meaning interpretation.

1) FRAMEWORK OF DEWEY DECIMAL CLASSIFICATION

Dewey Decimal Classification (DDC) originally is a book classification in library systems to designate a knowledge domain, which is built on sound principles that make it ideal as a general knowledge organization tool. It uses meaningful notation in universally recognized Arabic numerals, with well-defined categories, well-developed hierarchies, and a rich network of topic relationships [56]. At the broadest level, the DDC has ten main classes, as shown in Table 2, that cover the entire world of knowledge. Each main class is divided into ten divisions, and each division is further divided into ten sections. Since the parts of the DDC are arranged by disciplines, not subjects, a subject may appear in more than one class.

TABLE 2. Ten main classes of DDC.

No.	Section	No.	Section
000	Computer science, information & general works	500	Science
100	Philosophy & psychology	600	Technology
200	Religion	700	Arts & recreation
300	Social sciences	800	Literature
400	Language	900	History & geography

TABLE 3. Illustration of the concept of a lexical matrix.

Word Meanings	Word Forms				
	F_1	F_2	F_3	…	F_n
M_1	$E_{1,1}$	$E_{1,2}$			
M_2		$E_{2,2}$	$E_{3,3}$		
M_3					
…					…
M_n					$E_{m,n}$

Polysemous words

Synonyms words

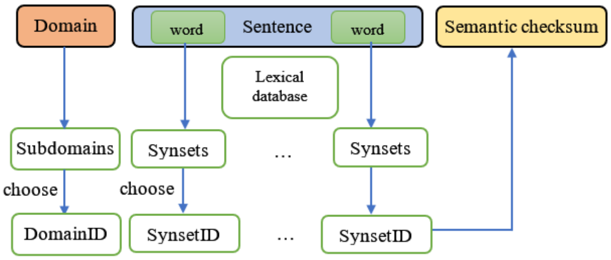

FIGURE 3. Calculation of semantic checksum.

Three digits of Arabic numerals are used to represent each class in the DDC hierarchically. The first digit represents the main class. The second digit indicates the division. A dot follows the third digit in a class number, after which division by ten continues to the specific degree of classification needed. For instance, the history of England is placed under 942, the history of the Stuart period at 942.06, and the history of the English Commonwealth at 942.063. The structural hierarchy represents all topics in a tree structure. With this characteristic, DDC exhausts the entire world of knowledge.

D. SELECTION OF WORD MEANINGS USING WORDNET

Words are the basic components of a sentence. Analyzing word meanings is the key to understanding a sentence. Different tools for word meaning reasoning exist, such as WordNet, Harvard General Inquirer, and Celex. We chose WordNet since it has been commonly adopted in many applications.

WordNet is a large online lexical reference system, an expert-built English-language network [57]. English nouns, verbs, adjectives, and adverbs are organized into synonym sets (called synsets), which are groups of synonyms with similar meanings.

The lexical matrix is the mapping between forms and meanings, as shown in Table 3. Word forms are listed as the headings for the columns, and word meanings are the headings for the rows. The letter E denotes an entry in the matrix. The word form is polysemous if two entries are in the same column (e.g., $E_{1,1}$ and $E_{1,2}$). The mappings are many-to-many relations. Some forms have several meanings, while some meanings can be expressed in several forms.

The new version, WordNet 3.1, contains more than 117,659 synsets organized into 206,941 lexicalized items [57]. Synsets and lexical items are connected to each other by various semantic and lexical relations in a hierarchy [30], [57]. Synsets form relations with other synsets to form a hierarchy of concepts, starting from the most general concepts, moving on to moderately abstract ones, and concluding with the most specific ones. WordNet provides a query interface to humans and machines for interactions with the central database. WordNet has been ported to many different computer systems [30].

V. SEMANTIC CONFIRMATION AT THE WORD LEVEL

This section describes the steps to ensure DoU at the word level (WDoU). First, a sentence is tokenized for semantic checksum calculation. Second, the sentence is paraphrased into several versions, and only the best paraphrasing sentence is selected. Lastly, we propose a scoring function to measure the WDoU.

A. SEMANTIC CHECKSUM FOR WORDS

Each sentence is decomposed into words. WordNet is used to look up each word's meaning [30]. For example, we look up the Synset for 'bank' in Python using "wn.synsets('bank')[0]", where the WordNet object is imported as 'wn'. Then Name method is used to get the unique ID for the synset directly: "bank.n.01". If multiple synset_IDs are returned, the sender or receiver must choose a proper ID for each word.

The DoU at the receiver is measured by checking whether both the sender and the receiver choose the same meaning for each word. Figure 3 shows the details of the implementation of how the sender encodes a sentence.

The sender performs stop word removal, tokenization, and stemming for the sentence, and removes the words with only one meaning because there is no need to make any choices on their meanings. Then, the sender specifies a KD using DDC, and the proper meaning of each word is looked up using WordNet. The domain ID and synset_IDs are encoded as a semantic checksum, which is appended at the end of the message. There are different ways for message checksum computation, such as the hash function Cyclic Redundancy Check (CRC).

The above steps are repeated at the receiver to compute the checksum, which is then compared with the sender's checksum. If the two checksums are the same, the semantics is verified at the word level. Otherwise, the weighted average of the semantic similarity is calculated.

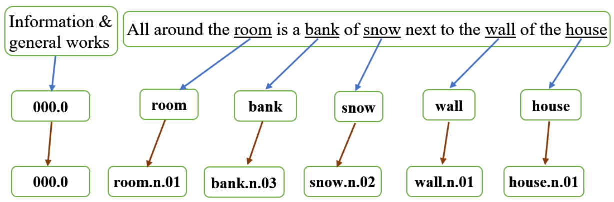

FIGURE 4. Example of sentence encoding.

The following is a real example to demonstrate the implementation. Suppose the sender sends the sentence, "All around the room is a bank of snow next to the wall of the house" to the receiver; then the sentence is encoded, as shown in Fig 4. If both the sender and receiver fully understand the meaning of the sentence, they should choose the same Synset IDs for the selected words.

B. SELECTION OF THE BEST PARAPHRASING SENTENCE

To minimize the understanding difference, the sender first paraphrases multiple versions of a sentence so that $T^i = \left\{T_1^i, T_2^i, \cdots T_j^i, \cdots T_d^i\right\}$, where d is the number of versions. Each version's difference from the original meaning is less than or equal to a negligible value Δ. The best paraphrasing sentence T_{best}^i is chosen based on both sides' understanding and the similarity between the paraphrasing sentence and the original sentence. The mathematical expression for selecting the best paraphrasing version of sentence i is:

$$T_{best}^i = argmin_{T_j^i \in T^i} \left\{T_j^i \,|\, f\left(T_j^i\right) \le f\left(T_k^i\right), \forall T_k^i \in T^i\right\}. \tag{7}$$

Let $\sum_{j\in n} sim_w\left(t_j^{is}, t_j^{ir}\right)$ denote the total understanding at the word level, and n is the sentence length of T_j^i. The similarity between T_j^i and T_0^i is denoted as $sim_s\left(T_0^i, T_j^i\right)$. Then the scoring function is expressed as:

$$f\left(T_j^i\right) = 1 - \left(\sum_{j\in n} sim_w\left(t_j^{is}, t_j^{ir}\right)\right) \cdot sim_s\left(T_0^i, T_j^i\right). \tag{8}$$

The different versions' WDoUs are stored in a set E^i as follows:

$$E^i = \left\{f\left(T_1^i\right), f\left(T_2^i\right), \cdots f\left(T_j^i\right), \cdots f\left(T_d^i\right)\right\}. \tag{9}$$

Then E^i is used in the verification of SDoU. If a version cannot fulfill the sentence understanding requirement, then that version will be removed from E^i.

C. MEASUREMENT OF THE DoU

We propose a factor-based method to evaluate whether the receiver can correctly select the same word meanings as the sender. The factors are the matching of meaning selection, the importance of words in the sentence, and the difficulty of meaning selection. They are denoted by the vectors V, U and D, separately.

Let $sim_w\left(t_j^{is}, t_j^{ir}\right)$ denote the scoring function of word t_j^i's understanding.

$$sim_w\left(t_j^{is}, t_j^{ir}\right) = \sum_{k=1}^{n}\left(v_{jk}^i \cdot u_{jk}^i \cdot d_{jk}^i\right), \tag{10}$$

where v_{jk}^i, u_{jk}^i and d_{jk}^i are the elements of vectors V, U, and D, respectively. The value of each element is defined as follows:

- $v_{jk}^i = 1$ if two parties select the same meaning; otherwise, the value is 0.
- $0 \le u_{jk}^i \le 1$. u_{jk}^i can be computed using numerous schemes, such as smooth inverse frequency (SIF), inverse document frequency (IDF), term frequency-inverse document frequency (TF-IDF), and POS tag [25].
- The difficulty level of selection is proportional to the number of word meanings (f_{jk}^i) available to be chosen, then $d_{jk}^i = \frac{f_{jk}^i}{\sum_{k=1}^{n} f_{jk}^i}$, where $\sum_{k=1}^{n} d_{jk}^i = 1$.

A higher score reflects a deeper DoU of word meanings.

VI. SEMANTIC ASSURANCE AT THE SENTENCE LEVEL

This section proposes a model to measure the SDoU. We call it Multiple Guesses Under Changing Constraints (MGUCC). Assume that a sentence has the true meaning M_0^i. The sender intuitively knows the sentence's original meaning, such that it can produce another version of the sentence so that its meaning M_j^{is} is very close to M_0^i. Based on this assumption, the next step is to test whether the receiver can obtain the value of M_0^i. The problem is that M_0^i is not directly measurable. However, we can indirectly get it by comparing the sender's and receiver's performance in the measurement.

To do so, we conduct multiple tests or guesses, as the name of our model MGUCC states. In each test, we apply different constraints and ask both the sender and the receiver to paraphrase a new sentence. If the receiver knows the true value, he or she should be able to produce a sentence that is very close to M_0^i (as well as to M_j^{is}). Then, we measure the DoU difference between the sender and the receiver. If the receiver consistently produces a very similar meaning value to that of the sender in the various tests, we can say that the receiver has successfully understood the meaning. The main advantage of our proposed method is that the receiver can produce evidence to show its understanding. This approach can also be generalized to other similar applications.

Let sentence $T^i = \left\{t_1^i, t_2^i, \cdots t_k^i, \cdots t_n^i\right\}$ be the version with the best WDoU, with n being the sentence length. We conduct l tests for T^i under different constraints. For each test, a subset T'^i of the word elements is randomly selected from T^i, where $T'^i \subseteq T^i$. Then both the sender and receiver are asked to produce a new sentence based on the constraint such that the meaning is as close as possible to the original meaning M_0^i. Suppose we have the following sentence outputs after l tests for the sender and receiver respectively: $T^{is} = \left\{T'^{is}_1, T'^{is}_2, \cdots T'^{is}_k, \cdots T'^{is}_l\right\}$, $T^{ir} = \left\{T'^{ir}_1, T'^{ir}_2, \cdots T'^{ir}_k, \cdots T'^{ir}_l\right\}$. Then we compute the average sentence DoU difference $\bar{M}^i$ for sentence T^i as follows:

$$\bar{M}^i = 1 - \frac{1}{l}\sum_{k=1}^{l} sim_s\left(T'^{is}_k, T'^{ir}_k\right). \tag{11}$$

The average DoU of the whole message is the average DoU of sentences.

VII. SOLUTION METHOD

In this section, we derive how we actually compute the solution. In the objective function, we minimize the DoU difference between the sender and receiver for the whole message. A sentence is a basic unit in a message. Then, the problem becomes minimizing the DoU difference for individual sentences.

To optimize the DoU on both sides, sentence T^i is paraphrased d versions, $T^i = \left\{T^i_1, T^i_2, \cdots T^i_j, \cdots T^i_d\right\}$, with $sim_s\left(\mathrm{T}^{is}_j, T^{ir}_j\right) \le \varepsilon$, where $j \in [1,d]$. Our objective is to select an T^i_j such that the DoU difference between the sender and the receiver is minimized. The revised objective function is:

$$F = min\left(1 - \sum_{t^i_j \in T^i_j} sim_w\left(t^{is}_j, t^{ir}_j\right)\right) + \left(1 - sim_s\left(T^{is}, T^{ir}\right)\right). \tag{12}$$

In (12), the first part minimizes the DoU difference at the word level, while the second minimizes the DoU difference at the sentence level. The equation can be rewritten as follows:

$$F = min\left(2 - \left(\sum_{t^i_j \in T^i_j} sim_w\left(t^{is}_j, t^{ir}_j\right) + sim_s\left(T^{is}, T^{ir}\right)\right)\right). \tag{13}$$

Minimizing F is equivalent to maximizing the second part in (13). So, we further revise the objective function as a maximization function:

$$F' = max\left(\sum_{t^i_j \in T^i_j} sim_w\left(t^{is}_j, t^{ir}_j\right) + sim_s\left(T^{is}, T^{ir}\right)\right), \tag{14}$$

where sim_w is dependent on variable T^i_j, and the input of the second part in the modified objective function is dependent on the output of the first stage. So, the function can be rewritten in the standard form of a two-stage optimization. Figure 5 illustrates this two-stage validation method [58], [59], [60], [61].

For the paraphrase T^i_j, we can express the similarity between the sender and receiver as $sim_w\left(t^{is}_j, t^{ir}_j\right) = \sum_{k=1}^{n}\left(v^i_{jk} \cdot u^i_{jk} \cdot d^i_{jk}\right)$, where $(u^i_{jk} \cdot d^i_{jk})$ can be considered the normalized integrated weight w^i_{jk} with $\sum_{k=1}^{n} w^i_{jk} = 1$. The objective function (14) can be rewritten as the form of the two-stage optimization problem by introducing variables x and y. Let

$$x_j = v^i_{jk}, \tag{15a}$$

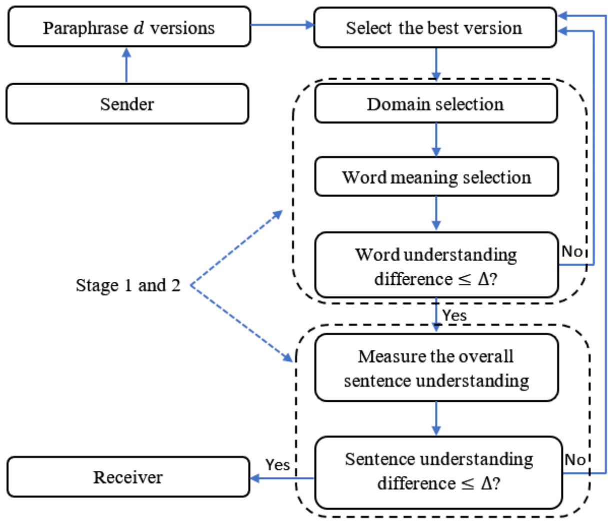

FIGURE 5. Framework of two-stage validation for sentence DoU.

$$y = sim_s\left(T^{is}, T^{ir}\right), \text{ and} \tag{15b}$$

$$w_j = w^i_{jk}. \tag{15c}$$

Then we rewrite the objective function (14) as follows:

$$max\ w_1x_1 + w_2x_2 + \cdots + w_nx_n + y, \tag{16}$$

subject to:

$$w_1x_1 + w_2x_2 + \cdots + w_nx_n \le 1, \tag{17a}$$
$$w_1x_1 + w_2x_2 + \cdots + w_nx_n \ge \beta_1, \tag{17b}$$
$$y \le 1, \tag{17c}$$
$$y \ge \beta_2, \tag{17d}$$
$$(w_1x_1 + w_2x_2 + \cdots + w_nx_n + y)/2 \ge \beta_3, \tag{17e}$$
$$\sum_{j=1}^{n} w_j = 1, \forall w_j \in [0, 1], \tag{17f}$$
$$x_j \in \{0, 1\}, \tag{17g}$$
$$\beta_1, \beta_2, \beta_3 \in [0, 1], \beta_1, \beta_2, \beta_3 \in \mathbb{R}. \tag{17h}$$

In the above formulation, the first and the second constraints ensure that the word similarity is within the range of $[\beta_1, 1]$. The third and the fourth constraints ensure the sentence similarity is within another range of $[\beta_2, 1]$. The fifth constraint ensures the combined similarity meets the minimum requirement. The sixth and seventh constraints require the summation of w_j as 1 and x_j as a binary variable. Lastly, the constants β_1, β_2, and β_3 are application-specific values that range from 0 to 1.

The objective function consists of two parts, reflecting the two decision stages in the whole process. In the first stage of the formulation, both the sender and receiver negotiate back and forth the proper version of M's expression, maximizing the WDoU in the first stage. The optimal second-stage decision is dependent on the output of the first stage. That means the output of the first stage is a decision variable in the objective function of the second stage.

To solve this problem, we further transform the above objective function into the standard form of the two-stage adaptive optimization problem as follows:

$$max\ c^T x + b^T y, \tag{18}$$

subject to

$$Fx \le f, \forall x \in \{0, 1\}, \tag{19a}$$
$$Hy \le h, \forall y \in [0, 1], \tag{19b}$$
$$Ax + By \le g, \forall x \in \{0, 1\}, \forall y \in [0, 1]. \tag{19c}$$

where c^T, b^T, F, f, H, h, A, B, and g in (18) and (19) are defined as follows:

$$\begin{aligned}
&c^T = [w_1, w_2, \cdots, w_n]^T, b^T = [1],\\
&F = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \\ -w_1 & -w_2 & \cdots & -w_n \end{bmatrix}, f = \begin{bmatrix} 0 \\ -\beta_1 \end{bmatrix},\\
&H = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, h = \begin{bmatrix} 1 \\ -\beta_2 \end{bmatrix},\\
&A = \left[-\frac{1}{2}w_1, -\frac{1}{2}w_2, \cdots, -\frac{1}{2}w_n\right], B = \left[\frac{1}{2}\right], g = [-\beta_3].
\end{aligned} \tag{20}$$

Then the formulation (18) can be rewritten as:

$$max\ c^T x + f(x), \tag{21}$$

subject to

$$Fx \le f, \tag{22a}$$
$$x \ge 0, \tag{22b}$$

where $f(x) = \max_{y \ge 0} \left\{ b^T y \mid Ax + By \le g \right\}$. The function $f(x)$ has a dual function $f'(x)$ as follows:

$$f'(x) = \min_{u \ge 0} \left\{ u^T (g - Ax) \mid B^T u \ge b^T \right\}. \tag{23}$$

So that the original problem (18) could be equivalently rewritten as follows:

$$max\ c^T x + \varphi, \tag{24}$$

subject to

$$Fx \le f, \tag{25a}$$
$$\varphi \ge f'(x), \tag{25b}$$
$$x \ge 0. \tag{25c}$$

In the above new formulation, we can see that the feasible region of $f'(x)$ does not depend on x; only the objective function depends on the input value of x. Then $f'(x)$ can be described as a set of extreme points and extreme rays. Let P be the set of all the extreme points of $f'(x)$ and R be the set of all the extreme rays of $f'(x)$. Then, an equivalent formulation of the original problem is expressed as the following full master problem (FMP). In the FMP, the variable y is eliminated from the original problem. A single scalar variable φ and a large number of constraints are added.

$$max\ c^T x + \varphi, \tag{26}$$

subject to

$$Fx \le f, \tag{27a}$$
$$\varphi \ge u^T (g - Bx), \forall u \in P, \tag{27b}$$
$$0 \ge u^T (g - Bx), \forall u \in R, \tag{27c}$$
$$B^T u \ge b^T, \tag{27d}$$
$$x \ge 0, \tag{27e}$$

where $B^T = \left[\frac{1}{2}\right]$ and $b^T = [1]$.

The original problem is now transformed into a linear programming problem. However, since the sets P and R are exponential in size, we need to use a delayed constraint generation algorithm to gradually reduce the search space. Using delayed constraint generation, we solve the FMP with only a small subset of the constraints while no other constraints are violated.

The feasible region of the *dual* is:

$$Dual := \left\{ u \mid B^T u \le b^T \right\}, \tag{28}$$

with the extreme points and extreme rays enumerated as follows: $P = \{p^1, p^2, \cdots, p^I\}$ are the extreme points, and $R = \{r^1, r^2, \cdots, r^I\}$ are the extreme rays. The solution to (26) is either optimal or unbounded.

- If it is unbounded, the algorithm will return one of the extreme rays $\bar{r} = r^j$ for some j, such that $(r^j)^T (g - Bx) > 0$ in which case $f(x) = +\infty$.
- If it is optimal, the algorithm will return one of the extreme points $\bar{p} = p^i$ for some i, such that $f(x) = (p^i)^T (g - Bx) = \max_{k=1,\ldots,I} (p^k)(g - Bx)$.

For T^i, e^i_j is a version's score of E^i. Let UB and LB be the upper bound and the lower bound of the solutions, respectively. Then the implementation of the delayed constraint generation algorithm is the following:

VIII. PERFORMANCE EVALUATION

This section conducts four experiments to evaluate the proposed model's performance. Subsections A and B experiments show that selecting synonyms and paraphrasing can improve the DoU between two independent machines at word and sentence levels. The third experiment in C demonstrates the effectiveness of the two-stage validation method using a study case. The fourth experiment in D evaluates the measurement effectiveness of meaning understanding.

A. ANALYSIS OF WORD UNDERSTANDING

In this experiment, the two machines have different reasoning abilities. The sender uses the T5_Paraphrase_Paws model based on the Google PAWs dataset, while the receiver uses the PEGASUS model based on the C4 & Hugenews dataset. The sender is able to understand the original meaning of the

Algorithm: Delayed Constraint Generation
Let $LB = -\infty$ and $UB = +\infty$;
while $UB\ LB > \varepsilon$ do
solve the master problem (FMP) and obtain $(\bar{x}, \bar{\varphi})$;
let LB be the lower bound obtained from the optimal objective value of the master problem;
for $w \in \Lambda$ do
solve the subproblem $Q_w(\bar{x})$;
if $Q_w(\bar{x})$ has an optimal solution, then
let $\bar{u}$ and $\bar{y}$ be the optimal and dual solution of $Q_w(\bar{x})$;
if $\bar{u}^T(g - Bx) > \bar{\varphi}$, then
add the constraint: $\bar{u}^T(g - Bx) > \varphi$ to the restricted master problem;
end
if $c^T\bar{x} + d^T\bar{y} \leq UB$, then
Best-Solution $\leftarrow (\bar{x}, \bar{y})$;
$UB \leftarrow \min\{UB, c^T\bar{x}, d^T\bar{y}\}$;
end
end
if $Q_w(\bar{x})$ is unbounded from the above, then
let $\bar{r}$ be the extreme ray that solve $Q_w(\bar{x})$;
add the constraint: $\bar{r}^T(g - Bx) \leq 0$ to the restricted master problem;
end
end
end

words in the sample sentence well. In contrast, the receiver's DoU ability at the word level is not as good as the sender's. This experiment aims to show that a better performance at the receiver can be achieved by selecting better synonyms.

In the experiment, the following sample sentence is given: "I will show you how to build a web application in Python using the SweetViz and its dependent library", and the following four tokens (keywords) are selected (i.e., "show", "build", "using" and "dependent"). In Section V, we proposed that there are three factors related to the understanding level of words: matching of meaning selection (V), the importance of words in the sentence (U), and the difficulty of meaning selection (D). In this experiment, let U and D contribute equally to the four keywords, while V has a different value for each keyword.

The following steps are performed to evaluate the DoU of the receiver. A set of synonyms are looked up for each keyword. For instance, the keyword "show" has three other synonyms: "see", "display", and "demonstrate". The WDoU can be obtained by calculating the similarity between the receiver's synonym candidates and the sender's synonym candidates for each keyword. Table 4 shows the different WDoUs in ascending order for the keyword "show" at the receiver.

The details of the experiment's configuration are given in Appendix A. The results are shown in Fig. 6.

TABLE 4. Different levels of understanding for the keyword "show".

Level	Synonym	Similarity between the synonym and the sender's understanding
Level-1	see	0.7374398438420945
Level-2	display	0.8522812866553033
Level-3	demonstrate	0.8703482097489922
Level-4	show	1.0 (fully understand)

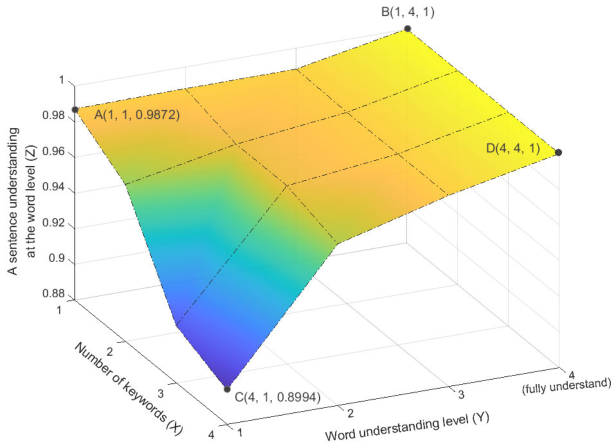

FIGURE 6. Example for the WDoU measurement.

Four boundary points are selected and respectively described to facilitate the understanding of Fig. 6. In particular, Point A (X: 1, Y: 1, Z: 0.9872) refers to the case that one keyword, "show", is chosen from the sample sentence with Level-1 understanding, the similarity between the sender's and receiver's WDoU is 0.9872; Similarly, Point B (X: 1, Y: 4, Z: 1.0) is the case that one keyword "show" is chosen from the sentence with Level-4 understanding, and the similarity between the sender's and receiver's sentence understanding is 1.0. In Point C (X: 4, Y: 1, Z: 0.8994), four keywords, "show", "build", "using", and "dependent", are selected from the sentence with Level-1 understanding, so the similarity between the sender's and receiver's WDoU is only 0.8994. While the four keywords of Point D (X: 4, Y: 4, Z: 1.0) are at Level-4 understanding, the understanding level between the sender and receiver reaches 1.0.

Based on the experiment results, the overall SDoU improves when the WDoU increases. When more keywords are selected, a more reliable understanding of the sentence can be achieved.

B. UNDERSTANDING MEASUREMENT AT SENTENCE LEVEL

In this experiment, we demonstrate the use of paraphrasing to improve the DoU between two independent machines in a natural language environment. Each machine individually paraphrases the most-equivalent sentence based on the input sentence. Then, we measure the similarity between these two paraphrased sentences, which reflects these two machines' DoU.

The objective of this experiment is to evaluate the effectiveness of sentence paraphrasing. In this experiment, two machines process the same original sentence. However, they have different paraphrasing models and reasoning abilities.

We set different knowledge reasoning abilities for the two machines by applying two different NLP paraphrasing models (i.e., PEGASUS and T5_Paraphrase_Paws models) that are trained based on different datasets (i.e., C4 & Hugenews corpora and Google's PAWs dataset). Details of the sample sentences are given in Appendix B.

For the case of ''without Sentence Paraphrasing (SP)'', the two machines each generate one paraphrasing sentence from the original sentence using their own paraphrasing models. The similarity between the two generated sentences is computed by the module en_core_web_sm from Spacy in Python.

For the case of ''with SP'', each machine generates 2 to 7 paraphrases from the original sentences. Then, the best meaning-matching paraphrase is selected. As expected, the more paraphrases the machine generates, the more likely a better version will be chosen.

Two factors influence the DoU between two machines. One is the number of generated paraphrases from each machine, and the other is the length of the original sentence. In addition, different lengths of original sentences are generated from the same meaning.

Figure 7 illustrates the DoU between two machines in M2M communication with the given number of generated paraphrases for each machine and different lengths of sentences. In all four sentences, the DoU between two machines generally increases with more paraphrases generated (horizontal axis). Meanwhile, with the same context, the longer original sentences generally result in a better DoU between the two machines. Sentence 4, with 40 words (top line), has the highest DoU, while Sentence 1, with 10 words (bottom line), has the lowest. This is because the longer original sentences have more unchanged words, increasing the similarity value in their meaning-match calculation. Interestingly, for Sentence 2 and Sentence 4, the two machines separately selected a better meaning-match paraphrase, but the DoU between the two machines performed lower than the case of ''without SP''. This is because the two machines paraphrase sentences according to their different databases and reference models, which cannot guarantee that a better-generated paraphrase for each entity can show better compatibility between the two machines' DoU.

Experiment B clearly shows that, under the same conditions, more paraphrases and longer sentences produce a higher DoU in a natural language environment. This outcome coincides with the numerical simulations in Experiment D: the performance of long-sentence is better than that of short-sentence.

C. PERFORMANCE OF TWO-STAGE VALIDATION METHOD

This experiment is an integrated evaluation for DoUs at the word and sentence levels, respectively. The experiment has two goals: (1) to reveal the relationship between WDoU and SDoU, and (2) to evaluate the wholistic performance of SDoU using Sentence Paraphrasing (SP) under different WDoUs.

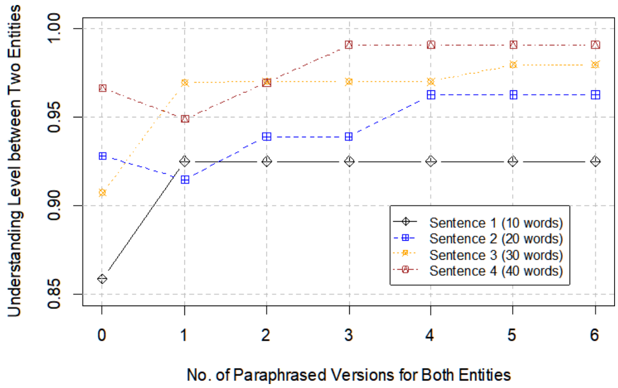

FIGURE 7. Performance versus a number of paraphrased versions.

The experiment was implemented on a workstation using Python 3.10 on Windows 10 with an Intel Xeon 3.6 GHz CPU and Nvidia Geforce RTX2080Ti GPU. The versions for Transformers and Sentence Piece are 4.30.0 and 0.2.0, respectively. The sender and receiver use two independent tools for sentence paraphrasing: T5_Paraphrase_Paws and PEGASUS. To further verify the robustness of the proposed model, we use a different tool, the transformer-based MiniLM, to measure sentence similarity. All the tested sentences are from a publicly available dataset, SemEval [62].

1) PART A

The experiment's Part A evaluates the relationship between WDoU and SDoU with two tests: Tests 1 and 2. We assume the sender possesses the exact meaning of each word in the sentence and can always generate the best paraphrasing version. On the receiver side, if it knows the exact meaning of the word, it can accurately look up the meaning's corresponding synonym word using WordNet. However, if the receiver does not know a word's meanings exactly, it picks one from the semantic meanings based on its understanding and then maps it to the corresponding synonym word. Then, the revised set of words is formed, and a paraphrasing sentence is generated using PEGASUS. Finally, this sentence is compared with the sender's version.

Some words are masked in the sentence during the test. In the NLP, a similar technique called the masked language model is used [63]. Words are randomly masked from the training data, forcing the model to guess the missing words. If a word is masked, the receiver may have different WDoUs on its meaning. We assume all words in the sentences are equally important and set the levels of WDoUs to 0%, 50%, and 100%, which are calculated based on (10). When WDoU = 50%, various combinations are available; the closest one is chosen.

In Test 1, the sentence length is set to 5, 12 same-length sentences are generated, and the results are taken as average. The overall results are shown in Fig. 8. The observations

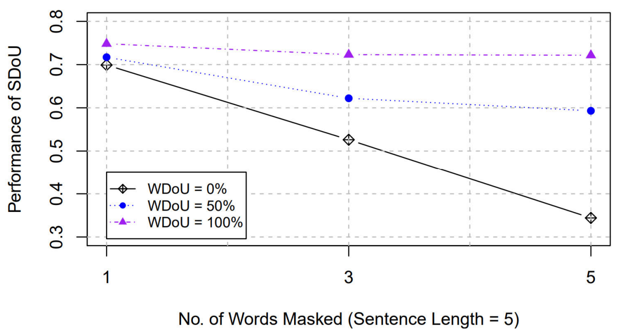

FIGURE 8. Relationship between WDoU and SDoU in short sentences.

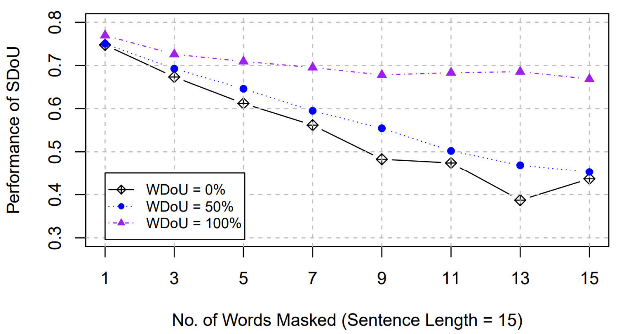

FIGURE 9. Relationship between WDoU and SDoU in long sentences.

are (1) a higher WDoU can result in a higher SDoU. For instance, when five words are masked, the SDoU is about 39% when WDoU = 0%, whereas the SDoU can reach 72% when WDoU increases to 100%. (2) When the number of words masked increases, the SDoU decreases.

In Test 2, the sentence length increases to 15, and the results are shown in Fig. 9. Similar to the previous test, we observe that the higher WDoU can achieve a better SDoU. Secondly, the SDoU increases under the same conditions when the sentence length increases.

2) PART B

The experiment's Part B shows how the SDoU can be improved using SP, includes another two tests: Tests 3 and 4. During the tests, all words in the sentence are masked. Three levels of WDoU are set to 0%, 50%, and 100%, respectively.

The SP is implemented as described in Section V-B. In the test, 35 different versions are generated for each sentence using Chatgpt_paraphraser_on_T5_base, and the top 20 sentences closest to the original sentence are filtered using MiniLM. The highest performance score is then selected as the final score.

In Test 3, the sentence length is set to 5, and the results are shown in Fig. 10. Without using SP, the SDoU is about 72.2% when the WDoU is 100%. With SP, the SDoU can reach 96.7% under the same conditions, improving performance by 33.9%.

The sentence length increases to 15 in Test 4. We observe that when sentences get longer, the difficulty level of

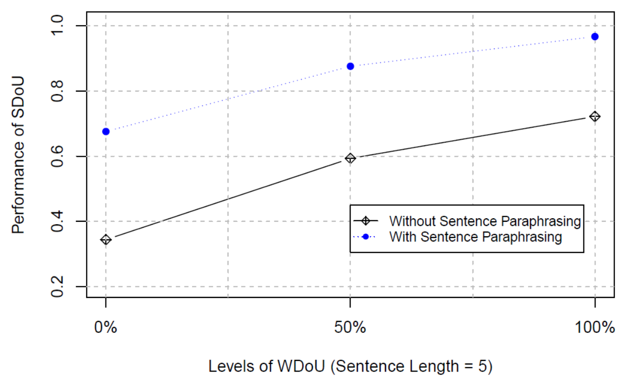

FIGURE 10. SDoU can be improved using SP in short sentences.

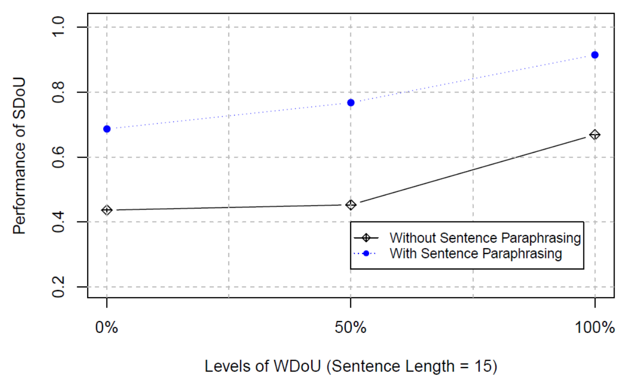

FIGURE 11. SDoU can be improved using SP in long sentences.

understanding increases; when the WDoU increases, the SDoU increases, as shown in Fig. 11. In the test, the performance of SDoU can be improved by at least 36.8% using SP.

D. QUANTITATIVE EVALUATION OF SENTENCE INTERPRETATION

Building a suitable dataset for performance benchmarking is very challenging due to subjectivity in language interpretation. However, language is technically just a set of formulas [64]. We designed the following special experiment to validate the results obtained in previous subsections. The main idea is to treat a message as a set of formulas. This special setting allows us to quantify the meaning of each sentence under different scenarios.

A sentence is generated based on the following structure: $X = x_1 \Box x_2 \cdots \Box x_i \cdots \Box x_n$, where x_i is an integer, $x_i \in [1, 100]$ and the arithmetic operator $\Box \in \{+, -, *, /\}$. If the receiver knows the meaning (i.e., denote it as M_0) of the terms, it should be able to make a good guess for any missing terms.

Given a set of sentences containing a mean number of terms from 6 to 26, in each test, the simulation randomly chooses a set of terms or joints, making them unknown to the receiver. Then, the receiver is asked to guess the possible missing terms or joints. Finally, the difference (in absolute value) between the best possible outcome (of the sender's version with Machine A) and the receiver's guess (with Machine B) is measured.

Example: To further explain the main idea behind the experiment, the following example is given to illustrate the details of the implementation. Suppose a sentence contains a formula like this:$2 + 4 \times 6 + 10 \div 2 - 3$ *, and the true value* M_0 *is 28. Machine A knows the true value, while Machine B does not. First, the last term, '3', is made unknown to both machines and asks them to guess. Machine A has no difficulty giving the best answer (say, 3), so the resulting meaning* $M^s = 28$*. However, Machine B does not know the value* M_0*, so it guesses a number randomly (say, 9), resulting in the meaning* $M^r = 22$*. To evaluate the performance, we compare* M^s *and* M^r *with* M_0*.*

The interpretation of a sentence is tested under the following four constraints, as shown in the equation at the bottom of the page.

1) EVALUATION OF SHORT SENTENCES

We first consider short sentences. The message length ranges from 3 to 9, with 6 as the average length. Each time, an integer is randomly chosen from [3, 9].

Both Machines A and B are put for testing under the same constraint. For a clearer view of the test results, only the two major performance categories (the best and lowest) are selected for comparison in the following figures. Without knowing the true value, the receiver is expected to have more difficulty making a good guess.

In each iteration, Machines A and B calculate the sentence's values (i.e., meanings). Then, these values are compared with the true value M_0. If the difference is less than 5%, this case is placed into the best performance category. If the difference is more than 100%, that case is placed into the lowest-performance category. We set the number of iterations to 10,000. Then, we count the frequencies of these two categories for machines A and B under the same constraints.

Since Machine A knows M_0, it always finds a value that is very close to M_0. In Fig. 12, the outcome of Machine A's best performance category (top line) is very close to 1 (89.2% to 97.3%). In contrast, the outcome of Machine B's best performance category (second line from the bottom) ranges from 3.4% to 12.7%. Moreover, the outcome of Machine B's lowest-performance category (second line from the top) ranges from 48.8% to 78.5%, while the result of Machine A's lowest-performance category (bottom line) is negligible (0.05% to 0.2%).

To further inspect the sensitivity of A and B under different constraints, we classify the performance into six categories based on the deviation from the true value M_0. The boundaries

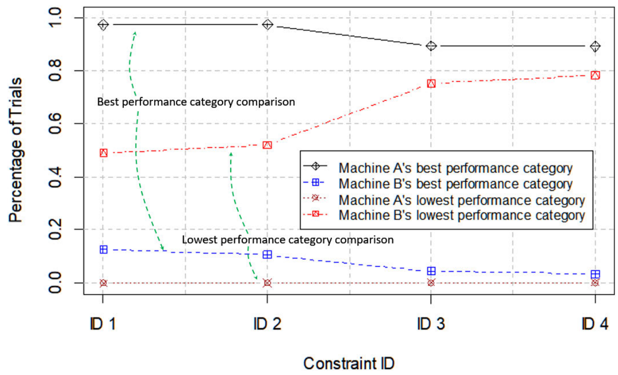

FIGURE 12. Comparison of the best and lowest-performance categories for short sentences. ID1 is Constraint 1, ID2 is Constraint 2, etc.

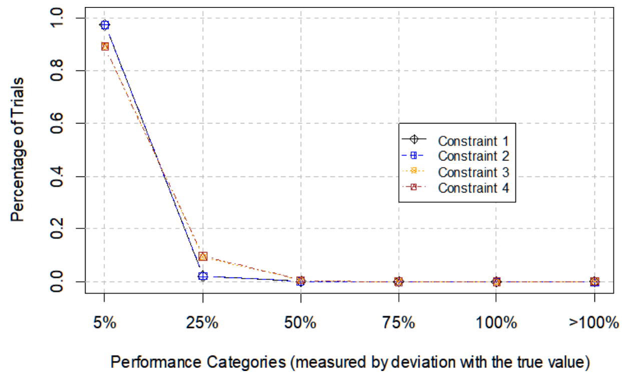

FIGURE 13. Machine A's performance for short sentences.

of the categories are 0%, 5%, 25%, 50%, 75%, 100%, and >100%. Fig. 13 shows that Machine A performs consistently well under different constraints. In most cases, its performance deviates by less than 5%. Understandably, the performance is a little lower when a harsh constraint (Constraint 3 or 4) is applied. Fig. 14 shows that Machine B performs poorly because the resulting deviations are larger than 100% in most cases.

2) EVALUATION OF LONG SENTENCES

In this experiment, the settings are similar to those used to simulate the short sentence, except that the sentence's length ranges from 13 to 19, with an average of 16.

Fig. 15 shows that Machine A performs even better in the long-sentence scenario under the same constraints. The outcome ranges from 93.9% to 98.9% in the best performance category (top line) because the impact of the constraints is diluted with the increase in sentence length. The same observation occurs in Machine B, and its lowest-performance

$$\left\{ \begin{array}{l|c} \text{Const1}: & \text{one keyword is removed from the full set, } X. \\ \text{Const2}: & \text{one keyword and one joint are removed from } X. \\ \text{Const3}: & \text{two keywords and one joint are removed from } X. \\ \text{Const4}: & \text{two keywords and two joints are removed from } X. \end{array} \right\}$$

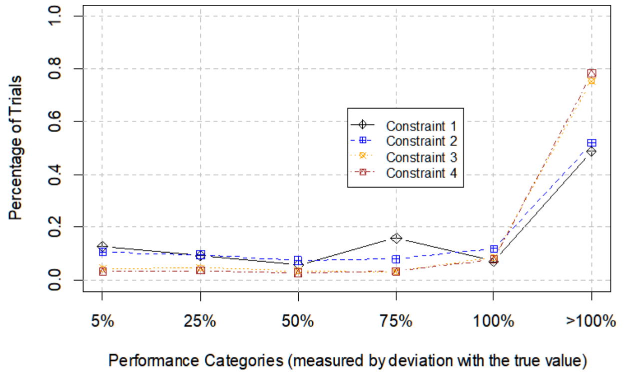

FIGURE 14. **Machine B's performance for short sentences.**

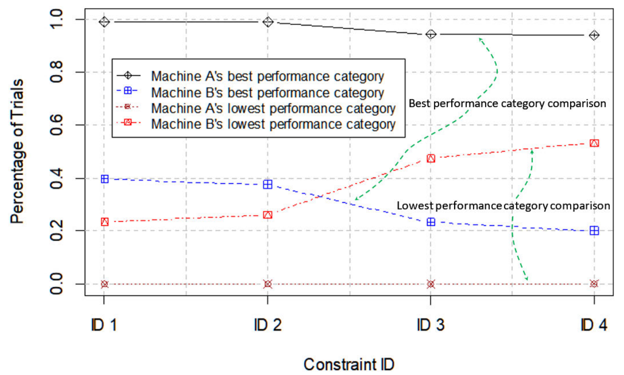

FIGURE 15. **Comparison of the best and lowest-performance categories for long sentences. ID1 is Constraint 1, ID2 is Constraint 2, etc.**

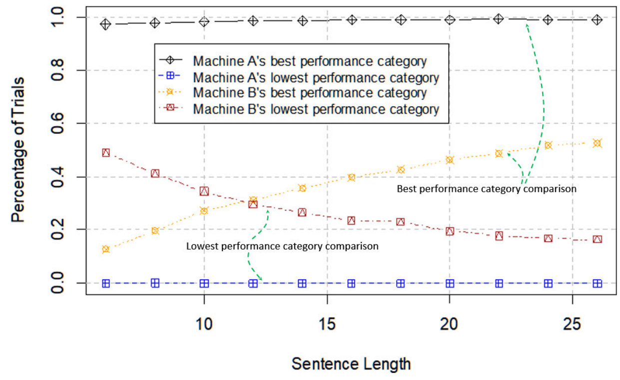

FIGURE 16. **Constraint 1: comparison between A and B.**

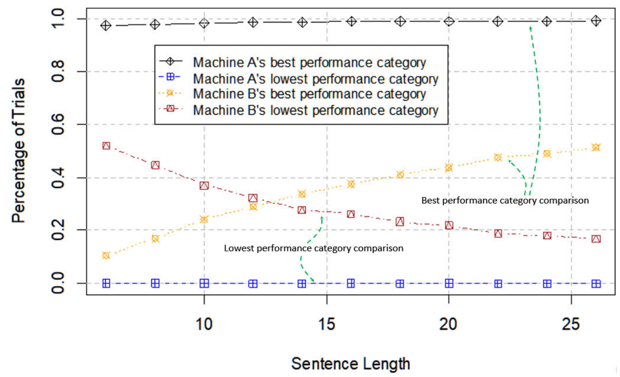

FIGURE 17. **Constraint 2: comparison between A and B.**

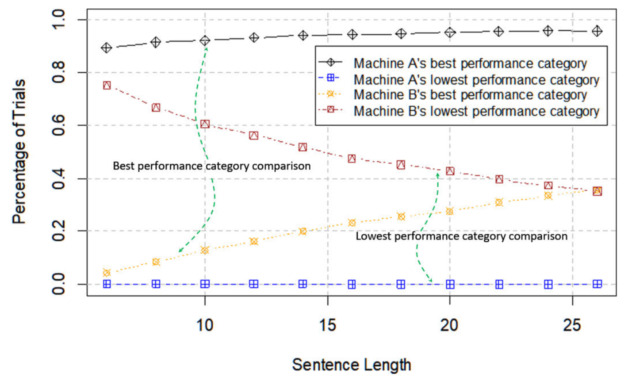

FIGURE 18. **Constraint 3: comparison between A and B.**

category ranges from 23.4% to 53.2%, which is considerably lower than the 48.8% to 78.5% for the short sentences in Fig. 12.

3) EVALUATION OF DIFFERENT CONSTRAINTS

The following experiments evaluate the effect of constraints at various sentence lengths. In Fig. 16, when Constraint 1 is applied, Machine A performs consistently well under different lengths. In contrast, Machine B's performance improves significantly as the length increases. The best category's performance increases as the sentence length increases, while the lowest category's performance decreases as the length increases.

When Constraint 2 is applied, Machine A still has outstanding performance, as shown in Fig. 17. In comparison, Machine B's performance is affected slightly by the new constraint.

When Constraint 3 is applied, as shown in Fig. 18, Machine A's performance is slightly affected. It cannot always get the best answer for each test. In comparison, Machine B is severely affected by the constraint. Its best performance category ranges from 4.5% to 35.5% because many unknown elements prevent Machine B from making a good guess.

In Fig. 19, when Constraint 4 is applied, Machine A's performance decreases a little bit, but it is still very close to 1. In comparison, Machine B's performance downgrades significantly, as its lowest-performance category increases from 40.1% to 78.5%.

4) EVALUATION OF DOMAIN SPECIFICATION

We set up another experiment to evaluate the performance with and without domain specification, as discussed in Section III-C. The harshest constraint (Constraint 4) is the default configuration. Given the following structure of a sentence: $x_1 \Box x_2 \cdots \Box x_i \cdots \Box x_n$, we group integer x_i into three combinations: all: $I_0 \in [1, 100]$, first-half: $I_1 \in [1, 50]$, second-half: $I_2 \in [51, 100]$. We also group operator $\Box$ into another three combinations: all: $O_0 \in \{+, -, *, /\}$, first-half:

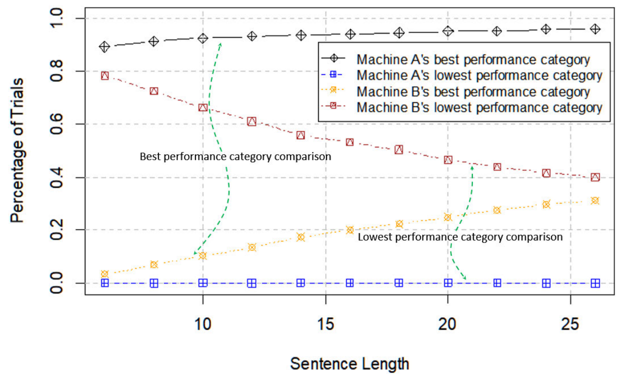

FIGURE 19. Constraint 4: comparison between A and B.

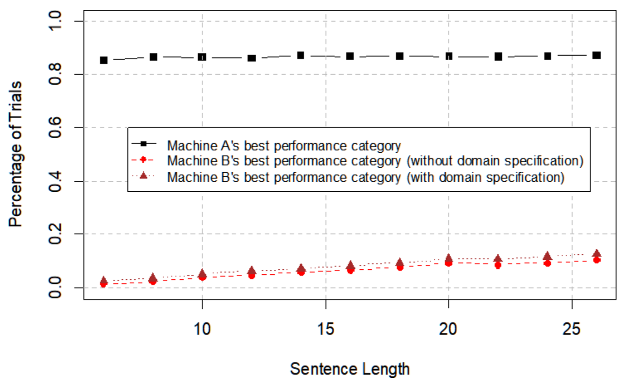

FIGURE 20. Performance of domain specification in the best performance category.

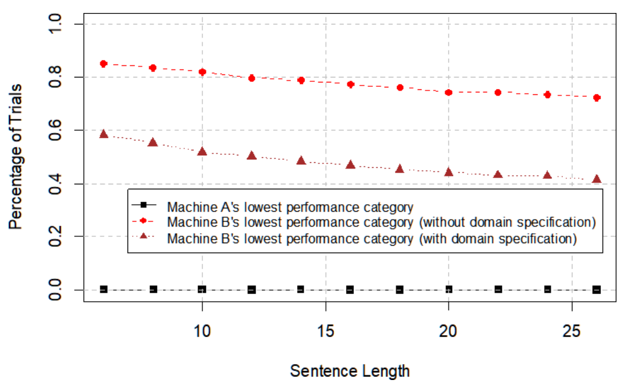

FIGURE 21. Performance of domain specification in the lowest performance category.

$O_1 \in \{+, -\}$, and second-half: $O_2 \in \{*, /\}$. A sentence can be generated using any combination of the above terms and joints. The objective is to check whether the performance can be improved when the combination (domain) is specified in advance.

In Fig. 20, we see that when the knowledge domain is specified, the performance improves significantly by 18.2% and to 82.7% in the best performance category. This performance difference may be difficult to see in Fig. 20 because we also plot the performance of Machine A's best category in the same graph. However, one can see in the graph that the "with domain specification" case is higher than the "without domain specification" case for Machine B.

A similar improvement is also observed in the lowest category, as the percentage of trials reduces from 31.4% to 42.7% in Fig. 21. We can clearly see that Machine B's lowest performance category for the "with domain specification" case (middle line) is much lower than the "without domain specification" case (top line). So, with domain specification, the best category is higher, while the worst category is much lower.

IX. CONCLUSION

This paper investigated the DoU problem in M2M SemComs. Three related sub-problems were studied: (a) inconsistent KB, (b) cross-domain interpretation, and (c) measurement of understanding. We proposed a novel model of cross-domain M2M SemCom to deal with (a) and (b). This model is the first to explore the application of an existing library knowledge management system in domain classification. For (c), we proposed a feedback-based method to quantify and optimize the DoU. We evaluated the performance of the proposed model using four experiments: analysis of word understanding, the understanding level and paraphrasing, the performance of the two-stage validation method, and the quantitative evaluation of sentence interpretation. The results look encouraging. The benefits of the proposed model are (1) improvement in the accuracy and efficiency of task execution in M2M applications, (2) minimization of cumulative semantic error in M2M communications, and (3) extendibility to other applications, such as M2H communications. In the future, we will continue investigating other relevant aspects of the studied problem, such as performance optimization, supporting a broader scope of data and information (images and speech), and semantic languages [35]. We will extend our research to the SemCom of multi-model [8] and structural heterogeneity applications.

APPENDIX A

Keyword: show

- A: Sender word understanding: show
- B.1 Level-1: Receiver Synonym 1: see
  Similarity between Receiver's Synonym 1 and sender's understanding: 0.7374398438420945
- B.2 Level-2: Receiver Synonym 2: display
  Similarity between Receiver's Synonym 2 and sender's understanding: 0.8522812866553033
- B.3 Level-3: Receiver Synonym 3: demonstrate
  Similarity between Receiver's Synonym 3 and sender's understanding: 0.8703482097489922
- B.4 Level-4: Receiver Synonym 4: show
  Similarity between Receiver's Synonym 4 and sender's understanding: 1.0 (fully understand)

Keyword: build

- Sender word understanding: build

- B.1 Level-1: Receiver Synonym 1: make
  Similarity between Receiver's Synonym 1 and sender's understanding: 0.8376394675571577
- B.2 Level-2: Receiver Synonym 2: create
  Similarity between Receiver's Synonym 2 and sender's understanding: 0.9299666424408446
- B.3 Level-3: Receiver Synonym 3: build
  Similarity between Receiver's Synonym 3 and sender's understanding: 1.0
- B.4 Level-4: Receiver Synonym 4: build
  Similarity between Receiver's Synonym 4 and sender's understanding: 1.0 (fully understand)

Keyword: using

- Sender word understanding: using
- B.1 Level-1: Receiver Synonym 1: with
  Similarity between Receiver's Synonym 1 and sender's understanding: 0.6039345547755669
- B.2 Level-2: Receiver Synonym 2: using
  Similarity between Receiver's Synonym 2 and sender's understanding: 1.0
- B.3 Level-3: Receiver Synonym 3: using
  Similarity between Receiver's Synonym 3 and sender's understanding: 1.0
- B.4 Level-4: Receiver Synonym 4: using
  Similarity between Receiver's Synonym 4 and sender's understanding: 1.0

Keyword: dependent

- Sender word understanding: dependent
- B.1 Level-1: Receiver Synonym 1: dependency
  Similarity between Receiver's Synonym 1 and sender's understanding: 0.7752782449611502
- B.2 Level-2: Receiver Synonym 2: subordinate
  Similarity between Receiver's Synonym 2 and sender's understanding: 0.8873387828048966
- B.3 Level-3: Receiver Synonym 3: dependent
  Similarity between Receiver's Synonym 3 and sender's understanding: 1.0
- B.4 Level-4: Receiver Synonym 4: dependent
  Similarity between Receiver's Synonym 4 and sender's understanding: 1.0

Analyze word understanding in a sentence

Context = "I will show you how to build a web application in Python using the SweetViz and its dependent library."

Sender's understanding = Context = "I will show you how to build a web application in Python using the SweetViz and its dependent library."

- **1 keyword (show)**

"I will see you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-1: (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9872218461917805

"I will display you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-2: (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9878798343505247

"I will demonstrate you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-3 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9881971324627044

"I will show you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-4 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):1.0

- **2 keywords (show, build)**

"I will see you how to make a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-1 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.968083827228167

"I will display you how to create a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-2 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9800206569662756

"I will demonstrate you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-3 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9871971324627044

"I will show you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-4 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):1.0

- **3 keywords (show, build, using)**

"I will see you how to make a web application in Python with the SweetViz and its dependent library."

Overall understanding of Level-1 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9122574543443046

"I will display you how to create a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-2 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9800206569662756

"I will demonstrate you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-3 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9871971324627044

"I will show you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-4 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):1.0

- **4 keywords (show, build, using, dependent)**

"I will see you how to make a web application in Python with the SweetViz and its dependency library."

Overall understanding of Level-1 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.8993971396851741

"I will display you how to create a web application in Python using the SweetViz and its subordinate library."

Overall understanding of Level-2 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9706632439501583

"I will demonstrate you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-3 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):0.9871971324627044

"I will show you how to build a web application in Python using the SweetViz and its dependent library."

Overall understanding of Level-4 (The sentence similarity between the sender's understanding and receiver's paraphrasing on the word level):1.0

APPENDIX B

In the experiment, there are three numbers for each outcome. The first number refers to the similarity between the original sentence and its generated paraphrase (P1) by using the PEGASUS model based on the dataset of C4 & Hugenews. Similarly, the second number is the similarity between the original sentence and its generated paraphrase (P2) by using T5_Paraphrase_Paws model based on the dataset of Google's PAWs. The third number is calculated by comparing P1 and P2. In addition, P1/P2 is the biggest value selected from all generated phrases for each original sentence.

Sentence 1

context = "I will be showing you how to build web applications in Python."

Long sentences (10 words)- orig

P1: 0.8588469290308214 I will show you how to use Python.

P2: 1.0 I will be showing you how to build web applications in Python.

P1/P2: —- 0.8588469290308214

Long sentences (10 words)-orig+1

P1: 0.9252235167403953 I will show you how to build applications in Python.

P2: 1.0 I will be showing you how to build web applications in Python.

P1/P2: —- 0.9252235167403953

Long sentences (10 words)-orig+2

P1: 0.9252235167403953 I will show you how to build applications in Python.

P2: 1.0 I will be showing you how to build web applications in Python.

P1/P2: —- 0.9252235167403953

Long sentences (10 words)-orig+3

P1: 0.9252235167403953 I will show you how to build applications in Python.

P2: 1.0 I will be showing you how to build web applications in Python.

P1/P2: —- 0.9252235167403953

Long sentences (10 words)-orig+4

P1: 0.9252235167403953 I will show you how to build applications in Python.

P2: 1.0 I will be showing you how to build web applications in Python.

P1/P2: —- 0.9252235167403953

Long sentences (10 words)-orig+5

P1: 0.9252235167403953 I will show you how to build applications in Python.

P2: 1.0 I will be showing you how to build web applications in Python.

P1/P2: —- 0.9252235167403953

Long sentences (10 words)-orig+6

P1: 0.9252235167403953 I will show you how to build applications in Python.

P2: 1.0 I will be showing you how to build web applications in Python.

P1/P2: —- 0.9252235167403953

Sentence 2

context = "I will be showing you how to build a web application in Python using the SweetViz and its dependent library."

Long sentences (20 words)-orig

P1: 0.9200722397699096 I will show you how to use the SweetViz and its dependent library to build a web application.

P2: 0.9580654747817575 I will show you how to build a web application in Python using SweetViz and its dependent library.

P1/P2: —- 0.9280881781674782

Long sentences (20 words)-orig+1

P1: 0.9200722397699096 I will show you how to use the SweetViz and its dependent library to build a web application.

P2: 0.980780594295261 I will be showing you how to build a web application in Python using SweetViz and its dependent library.

P1/P2: —- 0.9145645868070892

Long sentences (20 words)-orig+2

P1: 0.9626324488165116 I will show you how to build a web application using the SweetViz and its dependent library.

P2: 0.980780594295261 I will be showing you how to build a web application in Python using SweetViz and its dependent library.

P1/P2: —- 0.9390893530779642

Long sentences (20 words)-orig+3

P1: 0.9626324488165116 I will show you how to build a web application using the SweetViz and its dependent library.

P2: 0.980780594295261 I will be showing you how to build a web application in Python using SweetViz and its dependent library.

P1/P2: —- 0.9390893530779642

Long sentences (20 words)-orig+4

P1: 0.9626324488165116 I will show you how to build a web application using the SweetViz and its dependent library.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library.

P1/P2: —- 0.9626324488165116

Long sentences (20 words)-orig+5

P1: 0.9626324488165116 I will show you how to build a web application using the SweetViz and its dependent library.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library.

P1/P2: —- 0.9626324488165116

Long sentences (20 words)-orig+6

P1: 0.9626324488165116 I will show you how to build a web application using the SweetViz and its dependent library.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library.

P1/P2: —- 0.9626324488165116

Sentence 3

context = "I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an IoT application."

Long sentences (30 words)-orig

P1: 0.9761649434335236 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an internet of things application.

P2: 0.944880806637361 I will show you how to build a web application in Python using SweetViz and its dependent library and you will show me an IoT application.

P1/P2: —- 0.9073555985295011

Long sentences (30 words)-orig+1

P1: 0.9761649434335236 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an internet of things application.

P2: 0.9904880122368274 I will be showing you how to build a web application in Python using SweetViz and its dependent library and you will be showing me an IoT application.

P1/P2: —- 0.969455844483223

Long sentences (30 words)-orig+2

P1: 0.9797133399303349 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an Internet of Things application.

P2: 0.9904880122368274 I will be showing you how to build a web application in Python using SweetViz and its dependent library and you will be showing me an IoT application.

P1/P2: —- 0.9699896129014628

Long sentences (30 words)-orig+3

P1: 0.9797133399303349 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an Internet of Things application.

P2: 0.9904880122368274 I will be showing you how to build a web application in Python using SweetViz and its dependent library and you will be showing me an IoT application.

P1/P2: —- 0.9699896129014628

Long sentences (30 words)-orig+4

P1: 0.9797133399303349 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an Internet of Things application.

P2: 0.9904880122368274 I will be showing you how to build a web application in Python using SweetViz and its dependent library and you will be showing me an IoT application.

P1/P2: —- 0.9699896129014628

Long sentences (30 words)-orig+5

P1: 0.9797151055012149 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an internet of Things application.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an IoT application.

P1/P2: —- 0.9797151055012149

Long sentences (30 words)-orig+6

P1: 0.9797151055012149 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an internet of Things application.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me an IoT application.

P1/P2: —- 0.9797151055012149

Sentence 4

context = "I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an IoT application in Python using the SweetViz and its dependent library."

Long sentences (40 words)-orig

P1: 0.9724433246320807 I will show you how to build a web application in Python using the SweetViz and its dependent library and you will show me how to build an internet of things application in Python using the SweetViz and its dependent library.

P2: 0.9562216055149231 I will show you how to build a web application in Python using SweetViz and its dependent library and you will show me how to build an IoT application in Python using SweetViz and its dependent library.

P1/P2: —- 0.9663017868971506

Long sentences (40 words)-orig+1

P1: 0.9724445311092788 I will show you how to build a web application in Python using the SweetViz and its dependent library and you will show me how to build an Internet of Things application in Python using the SweetViz and its dependent library.

P2: 0.9799679805090751 I will be showing you how to build a web application in Python using SweetViz and its dependent library and you will be showing me how to build an IoT application in Python using SweetViz and its dependent library.

P1/P2: —- 0.9489028152872432

Long sentences (40 words)-orig+2

P1: 0.9905452465636423 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an Internet of Things application in Python using the SweetViz and its dependent library.

P2: 0.9799679805090751 I will be showing you how to build a web application in Python using SweetViz and its dependent library and you will be showing me how to build an IoT application in Python using SweetViz and its dependent library.

P1/P2: —- 0.9693359206599423

Long sentences (40 words)-orig+3

P1: 0.9905452465636423 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an Internet of Things application in Python using the SweetViz and its dependent library.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an IoT application in Python using the SweetViz and its dependent library.

P1/P2: —- 0.9905452465636423

Long sentences (40 words)-orig+4

P1: 0.9905452465636423 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an Internet of Things application in Python using the SweetViz and its dependent library.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an IoT application in Python using the SweetViz and its dependent library.

P1/P2: —- 0.9905452465636423

Long sentences (40 words)-orig+5

P1: 0.9905452465636423 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an Internet of Things application in Python using the SweetViz and its dependent library.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an IoT application in Python using the SweetViz and its dependent library.

P1/P2: —- 0.9905452465636423

Long sentences (40 words)-orig+6

P1: 0.9905452465636423 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an Internet of Things application in Python using the SweetViz and its dependent library.

P2: 1.0 I will be showing you how to build a web application in Python using the SweetViz and its dependent library and you will be showing me how to build an IoT application in Python using the SweetViz and its dependent library.

P1/P2: —- 0.9905452465636423

REFERENCES

[1] W. Weaver and C. Shannon, "Recent contributions to the mathematical theory of communication," in *A Mathematical Theory of Communication*. Urbana, IL, USA: Univ. of Illinois Press, 1949.

[2] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 5–41, Jan. 2023.

[3] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, "What is semantic communication? A view on conveying meaning in the era of machine intelligence," *J. Commun. Inf. Netw.*, vol. 6, no. 4, pp. 336–371, Dec. 2021.

[4] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 210–219, Feb. 2022.

[5] E. C. Strinati and S. Barbarossa, "6G network: Beyond Shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107930, doi: 10.1061/j.comnet.2021.107930.

[6] G. Shi, Y. Xiao, Y. Li, and X. Xie, "From semantic communication to semantic-aware networking: Model, architecture, and open problems," *IEEE Commun. Mag.*, vol. 59, no. 8, pp. 44–50, Aug. 2021.

[7] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Ye Li, "Semantic communications: Principles and challenges," 2021, *arXiv:2201.01389*.

[8] W. Yang, H. Du, Z. Q. Liew, W. Y. B. Lim, Z. Xiong, D. Niyato, X. Chi, X. Shen, and C. Miao, "Semantic communications for future internet: Fundamentals, applications, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 213–250, 1st Quart., 2023.

[9] D. Wheeler and B. Natarajan, "Engineering semantic communication: A survey," *IEEE Access*, vol. 11, pp. 13965–13995, 2023.

[10] A. Li, S. Wu, S. Meng, R. Lu, S. Sun, and Q. Zhang, "Towards goal-oriented semantic communications: New metrics, open challenges, and future research directions," 2023, *arXiv:2304.00848*.

[11] M. Kalfa, M. Gok, A. Atalik, B. Tegin, T. M. Duman, and O. Arikan, "Towards goal-oriented semantic signal processing: Applications and future challenges," *Digit. Signal Process.*, vol. 119, Dec. 2021, Art. no. 103134, doi: 10.1016/j.dsp.2021.103134.

[12] Z. Liu, Y. Lin, and M. Sun, "Sentence representation," in *Representation Learning for Natural Language Processing*. Berlin, Germany: Springer, 2020, ch. 4, pp. 59–89.

[13] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan, "Incorporating prior domain knowledge into deep neural networks," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2018, pp. 36–45.

[14] L. B. Bhajantri and R. Pundalik, "Data processing in semantic sensor web: A survey," in *Proc. 3rd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Tumkur, India, Dec. 2017, pp. 166–170.

[15] Y. Zheng, "Methodologies for cross-domain data fusion: An overview," *IEEE Trans. Big Data*, vol. 1, no. 1, pp. 16–34, Mar. 2015.

[16] H. Xie and Z. Qin, "A lite distributed semantic communication system for Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Jan. 2021.

[17] E. Lakka, N. E. Petroulakis, G. Hatzivasilis, O. Soultatos, M. Michalodimitrakis, U. Rak, K. Waledzik, D. Anicic, and V. Kulkarni, "End-to-end semantic interoperability mechanisms for IoT," in *Proc. IEEE 24th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Limassol, Cyprus, Sep. 2019, pp. 1–6.

[18] Y. Lu, Y. Shi, G. Jia, and J. Yang, "A new method for semantic consistency verification of aviation radiotelephony communication based on LSTM-RNN," in *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, Beijing, China, Oct. 2016, pp. 422–426.

[19] D. Huang, X. Tao, F. Gao, and J. Lu, "Deep learning-based image semantic coding for semantic communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[20] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "SEED: Semantics enhanced encoder-decoder framework for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13525–13534.

[21] E. Uysal, O. Kaya, S. Baghaee, and H. Burhan Beytur, "Age of information in practice," 2021, *arXiv:2106.02491*.

[22] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 96–102, Jun. 2021.

[23] A. Shet, C. Henson, and S. S. Sahoo, "Semantic sensor web," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 78–83, Jul./Aug. 2008.

[24] J. Ahmed and D. Muqeem Ahmed, "Big data and semantic web, challenges and opportunities a survey," *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 631–633, Sep. 2018.

[25] Z. Quan, Z.-J. Wang, Y. Le, B. Yao, K. Li, and J. Yin, "An efficient framework for sentence similarity modeling," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 4, pp. 853–865, Apr. 2019, doi: 10.1109/TASLP.2019.2899494.

[26] Z. Liang and S. Zhang, "Generating and measuring similar sentences using long short-term memory and generative adversarial networks," *IEEE Access*, vol. 9, pp. 112637–112654, 2021.

[27] M. H. Nguyen and D. Q. Tran, "A semantic similarity measure between sentences," *Southeast Asian J. Sci.*, vol. 3, no. 1, pp. 63–75, Jun. 2014, [Online]. Available: https://www.researchgate.net/publication/272088094_A_SEMANTIC_SIMILARITY_MEASURE_BETWEEN_SENTENCES

[28] A. Pawar and V. Mago, "Calculating the similarity between words and sentences using a lexical database and corpus statistics," 2018, *arXiv:1802.05667*.

[29] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *Proc. IEEE Netw. Sci. Workshop*, New York, NY, USA, Jun. 2011, pp. 110–117.

[30] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Introduction to WordNet: An on-line lexical database," *Int. J. Lexicogr.*, vol. 3, no. 4, 1990, pp. 235–244, doi: 10.1093/ijl/3.4.235.

[31] F. Lv, T. Liang, X. Chen, and G. Lin, "Cross-domain semantic segmentation via domain-invariant interactive relation transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4333–4342.

[32] Y. Su and X. Yan, "Cross-domain semantic parsing via paraphrasing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, 2017, pp. 1235–1246.

[33] K. Sartipi and A. Dehmoobad, "Cross-domain information and service interoperability," in *Proc. 10th Int. Conf. Inf. Integr. Web-Based Appl. Services*, Nov. 2008, pp. 25–32.

[34] C. Casalnuovo, K. Sagae, and P. Devanbu, "Studying the difference between natural and programming language corpora," *Empirical Softw. Eng.*, vol. 24, no. 4, pp. 1823–1868, Jan. 2019.

[35] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. Vincent Poor, "Less data, more knowledge: Building next generation semantic communication networks," 2022, *arXiv:2211.14343*.

[36] R. N. Rathi and A. Mustafi, "The importance of term weighting in semantic understanding of text: A review of techniques," *Multimedia Tools Appl.*, vol. 82, no. 7, pp. 9761–9783, Apr. 2022.

[37] G. Kour, S. Ackerman, E. D. Farchi, O. Raz, B. Carmeli, and A. A. Tavor, "Measuring the measuring tools: An automatic evaluation of semantic metrics for text corpora," in *Proc. 2nd Workshop Natural Lang. Gener., Eval., Metrics (GEM)*, 2022, pp. 405–416.

[38] T. M. Getu, G. Kaddoum, and M. Bennis, "Making sense of meaning: A survey on metrics for semantic and goal-oriented communication," *IEEE Access*, vol. 11, pp. 45456–45492, 2023.

[39] M. Farouk, "Measuring sentences similarity: A survey," *Indian J. Sci. Technol.*, vol. 12, no. 25, pp. 1–11, Jul. 2019.

[40] Y. Li, Z. Bandar, D. McLean, and J. D. O'Shea, "A method for measuring sentence similarity and its application to conversational agents," in *Proc. 17th Int. Florida Artif. Intell. Res. Soc. Conf.*, 2004, pp. 820–825.

[41] J. A. Bateman, "Sentence generation and systemic grammar: An introduction," in *Iwanami Lecture Series: Language Sciences*, vol. 8. Tokyo, Japan: Iwanami Shoten, 1997.

[42] T. Song, J. Sun, B. Chen, W. Peng, and J. Song, "Latent space expanded variational autoencoder for sentence generation," *IEEE Access*, vol. 7, pp. 144618–144627, 2019.

[43] B. Trisedya, J. Qi, and R. Zhang, "Sentence generation for entity description with content-plan attention," in *Proc. AAAI Conf. Artif. Intell.*, New York, NY, USA, Apr. 2020, pp. 9057–9064.

[44] B. D. Trisedya, J. Qi, R. Zhang, and W. Wang, "GTR-LSTM: A triple encoder for sentence generation from RDF data," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1627–1637.

[45] D. Liu, J. Fu, Q. Qu, and J. Lv, "BFGAN: Backward and forward generative adversarial networks for lexically constrained sentence generation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 12, pp. 2350–2361, Dec. 2019.

[46] S. García-Méndez, M. Fernández-Gavilanes, E. Costa-Montenegro, J. Juncal-Martínez, F. J. González-Castaño, and E. Reiter, "A system for automatic English text expansion," *IEEE Access*, vol. 7, pp. 123320–123333, 2019.

[47] S. S. Mufwene, "The evolution of language as technology: The cultural dimension," in *Proc. 23rd Annu. Symp. About Lang. Soc.*, 2015, pp. 365–394.

[48] Lingoda Team. (May 4, 2020). *30 Fun Facts About the English Language*. [Online]. Available: https://blog.lingoda.com/en/fun-facts-english-language/

[49] M. E. W. Varnum and I. Grossmann, "Cultural change: The how and the why," *Perspect. Psychol. Sci.*, vol. 12, no. 6, pp. 956–972, Sep. 2017.

[50] M. Hillyer. (Nov. 18, 2020). *How Has Technology Changed—And Changed US—In the Past 20 Years?* World Economic Forum. [Online]. Available: https://www.weforum.org/agenda/2020/11/heres-how-technology-has-changed-and-changed-us-over-the-past-20-years/

[51] T. Al-Moslmi, N. Omar, S. Abdullah, and M. Albared, "Approaches to cross-domain sentiment analysis: A systematic literature review," *IEEE Access*, vol. 5, pp. 16173–16192, 2017.

[52] S. Willmott, J. Dale, and P. Charlton, "Agent communication semantics for open environments: Issues and challenges," Infoscience, École Polytechn. Fédérale de Lausanne (EPFL), Lausanne, Switzerland, Tech. Rep. EPFL-REPORT-52461, Jul. 2002.

[53] M. G. Behlol and H. Dad, "Concept of learning," *Int. J. Psychol. Stud.*, vol. 2, no. 2, pp. 231–239, Dec. 2010.

[54] Y. Xie, Y. Sun, and E. Bertino, "Learning domain semantics and cross-domain correlations for paper recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021, pp. 706–715.

[55] A. Hommersom and P. J. F. Lucas, "An introduction to knowledge representation and reasoning in healthcare," in *Foundations of Biomedical Knowledge Representation*. Berlin, Germany: Springer, 2015, pp. 9–32.

[56] C. D. Batty, *Introduction to the Dewey Decimal Classification*, Bingley, U.K., 1971.

[57] T. Miller and I. Gurevych, "WordNet-Wikipedia-Wiktionary: Construction of a three-way alignment," in *Proc. LREC*, Reykjavik, Iceland, 2014, pp. 2094–2100.

[58] P. Du, B. Li, Q. Zeng, D. Zhai, D. Zhou, and L. Ran, "Distributionally robust two-stage energy management for hybrid energy powered cellular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12162–12174, Oct. 2020.

[59] A. Liu, V. K. N. Lau, and M.-J. Zhao, "Online successive convex approximation for two-stage stochastic nonconvex optimization," *IEEE Trans. Signal Process.*, vol. 66, no. 22, pp. 5941–5955, Nov. 2018.

[60] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, "Adaptive robust optimization for the security constrained unit commitment problem," *IEEE Trans. Power Syst.*, vol. 28, no. 1, pp. 52–63, Feb. 2013.

[61] K. Shang, F. T. S. Chan, S. Karungaru, K. Terada, Z. Feng, and L. Ke, "Two-stage robust optimization for the orienteering problem with stochastic weights," *Complexity*, vol. 2020, pp. 1–15, Nov. 2020, doi: 10.1155/2020/5649821.

[62] (2018). *Semantic Evaluation Datasets*. Accessed: Apr. 19, 2024. [Online]. Available: https://www.kaggle.com/datasets/azzouza2018/semevaldatadets

[63] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.

[64] R. Fagin, J. Y. Halpern, Y. Moses, and M. Vardi, *Reasoning About Knowledge*. Cambridge, MA, USA: MIT Press, 2004.

RICKY YUEN-TAN HOU (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in computer science from Hong Kong Baptist University (HKBU), in 1999 and 2003, respectively. He joined the Department of Computer Science, HKBU, as an Assistant Professor after his graduation. In 2005, he joined the BNU-HKBU United International College (UIC), where he is currently the Associate Dean and an Associate Professor with the Division of Science and Technology. His research interests include autonomous driving, P2P networking, and unmanned autonomous systems.

HUI ZHANG (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees (Hons.) in computer science and engineering from Zhejiang University, in 1999 and 2002, respectively, and the Ph.D. degree in computer vision from The University of Hong Kong, in 2007. She is currently with the Program of Computer Science and Technology, BNU-HKBU United International College, as a Professor. She has authored/coauthored more than 40 papers in top journals/conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, CVPR, and ACM MM. Her research interests include camera calibration, motion recovery, and 3D reconstruction.

GANG LIU received the bachelor's degree in computer science and technology from the Central South University of Forestry and Technology, Changsha, China, in 2018. He is currently pursuing the M.Phil. degree in computer science and technology with BNU-HKBU United International College, Zhuhai, China. His current research interests include wireless communications and networking, vehicle privacy protection, optimization algorithm, and their applications.

JEFFERSON FONG received the B.A. degree from the University of California at Berkeley, and the Ph.D. degree from Brown University, Providence, RI, USA. He is currently teaching with the Division of Science and Technology, BNU-HKBU United International College (UIC).

SEON-PHIL JEONG received the master's and Ph.D. degrees in management information systems from Chungbuk National University. He joined BNU-HKBU United International College (UIC), in 2008, and is currently an Associate Professor with the Department of Computer Science. His current interests include recommender systems, business intelligence systems, and distributed computing systems.