DIGITAL SEMANTIC COMMUNICATION METHODS FOR TEXT

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Tuna Özateş

June 2025

Digital Semantic Communication Methods for Text
By Tuna Özateş
June 2025

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Aykut Koç(Advisor)

Erdal Arıkan

Javad Haghighat

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

DIGITAL SEMANTIC COMMUNICATION METHODS FOR TEXT

Tuna Özateş
M.S. in Electrical and Electronics Engineering
Advisor: Aykut Koç

June 2025

Semantic communication, a paradigm concentrating on correctly transmitting underlying semantic information instead of bit sequences, has proved effective. Deep learning (DL) enabled methods are mainly used with basic natural language processing (NLP) techniques for the semantic communication of text. However, most previous DL-enabled semantic communication systems use continuous-valued channel vectors incompatible with existing digital modulation-based systems and binary channels. First, to address this issue, we propose Sememe-based Semantic Communications (SememeSC), a semantic communication paradigm that utilizes a fundamental concept from linguistics, sememes. We design a semantic source encoding-decoding strategy based on linguistics and DL, combined with conventional communication algorithms. We provide experimental results verifying that the proposed SememeSC performs superior to baselines in additive white Gaussian noise (AWGN) and Rayleigh fading channels. Second, we extend DL-based joint source-channel coding architecture to binary channels with insertions, deletions, and substitutions (IDS). We propose a novel three-stage training algorithm combining gated recurrent unit (GRU) networks for marker detection, transformer-based semantic communication for continuous latent space, and lookup-free quantization for binarized latent space optimization. The proposed DeepJSOC is the first to integrate DL-based error correction networks into joint-source channel coding schemes for binary channels with synchronization errors. We demonstrate the effectiveness of DeepJSOC by numerical experiments, achieving significant improvements over existing methods in text transmission over IDS channels.

Keywords: semantic communication, joint source-channel coding, IDS channel, deep learning, sememe, natural language processing (NLP), text.

ÖZET

METİN İÇİN SAYISAL SEMANTİK İLETİŞİM YÖNTEMLERİ

Tuna Özateş
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Aykut Koç
Haziran 2025

Semantik iletişim, bit dizileri yerine altında yatan semantik bilgilerin doğru bir şekilde iletilmesine odaklanan bir paradigma olup, etkinliğini kanıtlamıştır. Derin öğrenme (DL) tabanlı yöntemler, metnin semantik iletişimi için genellikle temel doğal dil işleme (DDİ) teknikleriyle birlikte kullanılmaktadır. Ancak, önceki birçok derin öğrenme (DL) tabanlı semantik iletişim sistemi, mevcut dijital modülasyon tabanlı sistemlerle ve ikili kanallarla uyumsuz olan sürekli değerli kanal vektörlerini kullanmaktadır. İlk olarak, bu sorunu çözmek için, dilbilimdeki temel bir kavram olan sememeleri kullanan Sememe-tabanlı Semantik İletişim (SememeSC) adlı bir semantik iletişim paradigması öneriyoruz. Dilbilim ve derin öğrenme (DL) tabanlı, geleneksel iletişim algoritmalarıyla birleşen bir semantik kaynak kodlama-çözme stratejisi tasarlıyoruz. Önerilen SememeSC'nin, toplanır beyaz Gauss gürültüsü (AWGN) ve Rayleigh sönümlemesi kanallarında temel yöntemlere kıyasla üstün performans sergilediğini doğrulayan deneysel sonuçlar sunuyoruz. İkinci olarak, derin öğrenme (DL) tabanlı birleşik kaynak-kanal kodlama mimarisini, eklemeler, silmeler ve yer değiştirmeler (IDS) içeren ikili kanallara genişletiyoruz. İşaretçi tespiti için kapalı geçitli tekrarlayan ünite (GRU) ağlarını, sürekli gizli uzay için dönüştürücü tabanlı semantik iletişimi ve ikili hale getirilmiş gizli uzay optimizasyonu için aramasız nicemlemeyi birleştiren yenilikçi üç aşamalı bir eğitim algoritması öneriyoruz. Önerilen DeepJSOC, senkronizasyon hataları içeren ikili kanallar için birleşik kaynak-kanal kodlama şemalarına DL tabanlı hata düzeltme ağlarını entegre eden ilk yaklaşımdır. DeepJSOC'un etkinliğini, IDS kanalları üzerinden metin iletimi konusunda mevcut yöntemlere göre önemli iyileştirmeler elde eden sayısal deneylerle gösteriyoruz.

Anahtar sözcükler: semantik iletişim, birleşik kaynak-kanal kodlama, IDS kanalı, derin öğrenme, sememe, Doğal Dil İşleme (DDİ), metin.

Acknowledgement

This work is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), Directorate of Science Fellowships and Grant Programmes (BİDEB), through the 2210-A National MSc Scholarship Program.

This work is also supported by Türk Telekomünikasyon A.Ş. through the 5G and Beyond Joint Graduate Support Program coordinated by the Information and Communication Technologies Authority.

I would like to express my deepest gratitude to my supervisor, Dr. Aykut Koç, for his invaluable guidance, support, and encouragement throughout the course of my research. His insight and expertise have been instrumental in shaping this thesis, and I am truly grateful for the opportunity to learn under his mentorship. I would also like to thank all the members of our lab for creating a supportive and stimulating research environment throughout my studies.

I am deeply grateful to my family, including my mother, my father, and my brother, for their unwavering support and encouragement throughout my academic journey. A special thank you to my brother, Dr. Mert Özateş, whose technical guidance and constant companionship have been a source of great strength and motivation for me throughout both my BS and MS studies.

I would like to thank my friends at Bilkent, especially Enes Koç and İlter Onat Korkmaz, with whom I have shared countless memories and experiences since our undergraduate years, supporting each other throughout the challenging yet rewarding journey of both our BS and MS studies. I would also like to thank Uras Kargı for his collaboration and contributions as my co-author in our joint work.

Lastly, I would like to express my heartfelt gratitude to my girlfriend, Doğa Yılmaz, for her love, patience, and understanding throughout the course of my studies. Her constant support and encouragement have been invaluable, and I am deeply thankful for her presence by my side during this journey.

Contents

List of Figures

List of Tables

Disclaimer

This thesis is a cumulative work built upon a series of papers previously published by the author. Although each paper is self-contained, this thesis weaves them together into a cohesive narrative, integrating their findings and discussions to provide a comprehensive overview of the research. Any overlapping material has been properly cited. The original research and results presented here have been presented in the following publications:

- T. Ozates, U. Kargı and A. Koc, "Sememe-Based Semantic Communications," in IEEE Communications Letters, vol. 28, no. 10, pp. 2308-2312, Oct. 2024. [1]
- T. Ozates and A. Koc, "Semantic Communication Over Channels With Insertions, Deletions, and Substitutions," in IEEE Communications Letters, vol. 29, no. 3, pp. 586-590, March 2025. [2]

Chapter 1

Introduction

Modern communication systems are built on Shannon's work on information theory [3]. Shannon has set the goal of a communication system as the correct transmission of bits or symbols to receivers. This approach ignores the *intrinsic meaning* of messages. Weaver extended Shannon's theory and divided the communication problem into three levels: technical, semantic, and effectiveness [4]. However, work on communication systems mostly dealt with problems at the technical level for a long time, following Shannon's assumption. Although this paradigm led to constant improvement in the field, it suffers from a fundamental constraint called channel capacity, which limits the maximum rate of information that can be transmitted through a channel at a specific signal-to-noise ratio (SNR). Advanced channel coding techniques such as Turbo [5], low-density-parity-check (LDPC) [6], and polar codes [7] have managed to approach this capacity. However, with the growing amount of data and energy consumption over wireless networks, channel capacity became a serious limitation, and an alternative view in communication system design became necessary. To this end, researchers have started to work on problems at a semantic level, in addition to a technical level [8–11]. As a result, a new paradigm called semantic communications, which concentrates on the correct transmission of *meaning* of messages rather than minimization of bit errors, has emerged [12]. Semantic communication systems mainly aim to transmit the essential information related to the message rather than transmitting the message itself in its entirety.

A significant problem to solve at the semantic level is "How to define the meaning behind the bits?" according to Weaver [4]. There were already some attempts in the pre-deep learning (DL) era towards answering this question [8]. With the rise of DL, powerful feature extraction techniques have emerged, which lead to the rapid development of semantic communications [13], [14]. DL's significant success in computer vision and natural language processing (NLP) fields has also motivated transferring these methods to the semantic communications area. As a result, DL-based semantic communications systems are proposed for transmission of text [14–17], images [18–24], video [25, 26], and speech [27–31].

Conventional communication systems perform the source coding and channel coding separately. This approach is desirable in practical systems due to more straightforward implementation. However, to utilize the automated feature extraction ability of neural networks (NN), pure DL-based semantic communication systems perform source and channel coding jointly [14, 32]. Most such systems can be viewed as an autoencoder with a latent space affected by channel conditions. Although these systems achieve very robust results in low SNR regimes with less symbol usage, there are problems with their practical implementations. They require the transmission of continuous values over channels rather than bits, which makes them incompatible with established structures designed for digital transmission. Another disadvantage of pure DL-based semantic communication systems is that they require analog modulation or an infinite constellation space, which is expensive to implement. There are also attempts to design intelligent communication systems for text transmission that are also compatible with existing digital communication algorithms by utilizing knowledge graphs (KGs) [33], quantization [34], and context [35]. However, combining linguistic features and NLP tools in semantic communication systems with conventional communication techniques has yet to be deeply explored. Various concepts, techniques, and tools in traditional NLP exist, waiting to be leveraged in the area of semantic communications.

Another shortcoming of existing joint-source channel coding (JSCC) systems is the requirement of differentiable channel models such as additive white Gaussian noise and Rayleigh fading channels. Most of the work in this direction ignores the binary-input channels, which are not differentiable. A significant class of binary input channels are prone to synchronization errors, which can be modeled as insertion-deletion-substitution (IDS) channels. These types of channels occur in many scenarios like wireless communications [36, 37], file synchronization [38], bit-patterned media [39], racetrack memories [40], and DNA storage [41]. Synchronization errors are hard to deal with since an undetected deletion or insertion in the input sequence results in the wrong transmission of the rest of the input sequence. It should be noted that deletion is different from erasure in binary erasure channels (BEC) since in BEC, the receiver is notified about transmission, and no synchronization error occurs. Usually, reliable communication over IDS channels is ensured by inserting a pre-defined marker sequence as inner code [42], and this sequence is then detected using the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm to combat synchronization errors [43]. In addition to marker code, a common channel code such as low-density parity-check (LDPC) [6] or Polar [7] is used as an outer code for error correction [43]. Alternatively, [44] proposed a redundant residue number system-based error correction scheme. Recently, [45] and [46] proposed DL models for detecting marker codes. Furthermore, [47] applied a binarized version of joint-source-channel-coding architecture on deletion channels but failed to combat synchronization errors. Although a plethora of work is dedicated to reliable communication over IDS channels, to our knowledge, no work in the literature studies DL-enabled semantic communication over these channels.

Motivated by the shortcomings of existing semantic communication systems described above, this thesis proposes two novel textual semantic communication systems. The first study introduces a novel semantic source encoding and decoding strategy that is compatible with existing source-channel coding and digital modulation algorithms. The second study extends the transformer-based end-to-end textual semantic communication systems to channels with insertions, deletions and substitutions.

This thesis is based on the two published journal articles: "Sememe-Based

Semantic Communications" [1] and "Semantic Communication Over Channels With Insertions, Deletions, and Substitutions" [2], which present the outcomes of the research conducted during the M.Sc. studies.

In the first study included in this thesis, we introduced a fundamental concept from linguistics, *sememes*, which are the smallest indivisible semantic units of natural languages [48, 49]. While words are the smallest elements of language, they can still be semantically decomposed. Linguists advocate that it is possible to express all words in a language by combinations of *sememes* [50]. For example, the word `winter` can be considered a combination of `season` and `cold`, whereas the word `wind` can be divided into meanings of `air` and `movement`. In these examples, `season`, `cold`, `air`, and `movement` are considered as sememes. Words `winter` and `wind` are annotated by a combination of those sememes. In a sense, we can make an analogy with chemistry and say that sememes resemble the elements in the periodic table. Sememes are limited in number, and we can compress the language and reconstruct words with a well-constructed sememe knowledge base (SKB) [50]. SKBs are knowledge bases that keep a set of sememes for every sense of every word in a language, in a dictionary format [51]. Expressing words in terms of combinations of sememes is essential for interpreting the intrinsic semantics of a word. Extracting the underlying meanings of words and utilizing them in communications is an essential step towards answering Weaver's question about "defining the meaning behind bits" [4]. In this study, we introduce sememe-based semantic communications (SememeSC), a novel semantic source encoding and decoding strategy to enhance text transmission in traditional communication systems. The proposed SememeSC uses sememes as the central semantic unit for source encoding. The main idea behind expressing words as a combination of sememes is to divide the semantic content and to map semantically similar words to similar bits to tackle errors in communication channels.

In the second study included in this thesis, we introduce deep joint source-outer channel coding (DeepJSOC), an end-to-end semantic communication architecture for IDS channels. We propose a three-stage training algorithm for the architecture: First, we train a gated recurrent unit (GRU) based marker detection network as in [46]. Second, we train a transformer-based semantic communication system with continuous latent space. Lastly, we binarize the latent space of the transformer using lookup-free quantization (LFQ) [52], integrate a pre-trained error corrector into the system, and jointly optimize the end-to-end system using a transfer learning approach.

The rest of this thesis is organized as follows: Chapter 2 reviews the literature in detail. Chapter 3 and Chapter 4 present the architecture and experimental results of SememeSC and DeepJSOC, respectively. We conclude the thesis with the discussion in Chapter 5.

Chapter 2

Related Work

Weaver first discusses the semantic aspect of communication [4]. Challenges in defining the meaning behind bits and defining appropriate metrics for semantic distortion made researchers leave this level of communication problems aside for a long time. Instead, they built systems on reliable transmission of bits. After several decades, this concept is revisited in [8], where a model based on logical probabilities is proposed for semantic data compression and reliable semantic communication. [53] extended this probabilistic approach and proposed a Bayesian game-based text transmission model that utilizes the meanings of words in source encoding. Although these models initiate an interest in the problem, the real source behind the accelerated development of semantic communication systems became DL [13, 14, 54]. Since semantic communication systems aim to extract important features for the task to be performed in the receiver, feature extraction capabilities of DL methods align with their design [9]. The task in the receiver can either be a reconstruction of the message fully (or with some loss) at the receiver or a downstream task that does not require the recovery of the message [55]. The paradigm that aims to transmit only related information for downstream tasks is called *task-oriented semantic communications* [56]. These tasks include image restoration [57], image classification [58, 59], question answering [60], visual question answering [61, 62], image retrieval [61], machine translation [61], time-series classification [63], text classification [64–66], textual similarity [65], and linguistic acceptability [65]. Since we proposed methods for reconstructing textual information in the receiver, further details of the task-oriented paradigm are out of our scope, and we will concentrate on the systems that aim to reconstruct input messages. In the rest of this chapter, we will mainly divide these methods into end-to-end DL-based methods that transmit continuous channel vectors (analog) and methods compatible with digital modulation and binary-input channels, explaining them in two parts. In each part, we will concentrate on the methods designed for text and mention examples from image counterparts only briefly.

2.1 Analog Semantic Communication Methods

The concept of designing an end-to-end communication system as a neural network was first introduced in [67], where an autoencoder structure is used. Channel effects are applied to the bottleneck of the autoencoder. This structure is source-ignorant, and the semantic aspect of the messages is neglected. It is shown that an autoencoder can learn to perform communication at the level of some channel coding and digital modulation techniques in terms of bit accuracy. The first model that uses this concept for the text semantic communication task is JSCC [13], which performs the joint source-channel coding of text with long short-term memory (LSTM) networks. Different from [67], the channel requires the transmission of continuous values instead of bits in this framework.

When JSCC was proposed, using LSTM's was the standard practice for text modeling. Shortly after, transformer networks have become state-of-the-art for NLP tasks [68]. Based on these models, one of the most influential models for DL-based semantic communications for text, DeepSC, is proposed in [14]. It consists of a transformer-based transmitter and receiver. Such a structure achieved reliable text communication over AWGN and Rayleigh fading channels in a very low SNR regime with few symbols per word. In addition to providing a model, [14] also utilized pre-trained language models to evaluate communication performance for the first time. As DeepSC uses a fixed network, a need emerged to make it more adaptive for different communication scenarios. In [15], a universal transformer is adopted to solve this issue. To combat the semantic noise caused by ambiguities, [16] utilized adversarial training in a transformer-based network. Since transformer models have substantial sizes, [69] proposed a framework to achieve similar performance with much fewer parameters. While previous models optimized networks with cross-entropy loss, methods are evaluated based on BERT [70] similarity. To solve this inconsistency, [71] incorporated semantic similarity to the model training objective to ensure semantic fidelity. To enhance the reliability of external knowledge, [72] incorporated emotion information into the model, and [73] used prior knowledge from a KG to improve the decoding capability of the receiver. [74] extended the single-user design of DeepSC to a multi-user case. [75] utilized generative pre-trained transformer models to replace missed or corrupted tokens. [76] is the first work to extend these methods for better resource allocation in text transmission.

In addition to methods for text, there is a plethora of important work for the semantic communication of images that relies on an end-to-end design. The success of the convolutional neural networks (CNN) in vision tasks motivated the first joint source-channel coding model for images to be designed based on CNNs [32]. Generative adversarial networks' capabilities are also transferred to semantic communication systems in [77]. [78] further improved on this approach by adopting and combining an RL-based semantic bit allocation model with a GAN-based semantic decoder. Pre-trained GANs are utilized in [79] and [80] exploited the image generation capability of diffusion models to accomplish the same task. [81] implemented the real-time semantic image transmission for the first time with a vision transformer. These methods achieved robust performance under low SNR. Since they do not have a fixed signal constellation scheme, they can extract features more freely and transmit any symbol through a channel. However, these methods are more flexible than traditional methods and cannot be designed non-jointly.

2.2 Digital Semantic Communication Methods

Rather than designing an end-to-end communication system relying on DL, some text semantic communication systems improve traditional design with semantic source encoding and decoding strategies. One popular paradigm in this direction is utilizing KGs as shared knowledge bases [82]. Using KGs, sentences can be decomposed into three components: head, entity, and tail [82]. These triplets can be used as semantic symbols subject to traditional source encoding methods instead of characters. Such an approach significantly decreases the required number of bits. [83] improved on this source encoding strategy by incorporating the relative importance information of the triplets into the transmitter and proposing a graph attention network-based restoration model.

Inspired by earlier works that use logical probabilities [8], [84] introduced an additional probability dimension to KG to represent the importance of the information and combined two approaches. [85] combined KGs with an attention-based RL algorithm to further optimize performance.

Apart from using KGs, some systems combine DL-based methods with quantization to obtain binary data as channel input. [17] extracted the textual features with word embeddings and transformers for feature extraction, combined the quantized features with conventional channel coding, and combat the semantic errors by iteratively enhancing LDPC decoder output with a neural network. In [34], the architecture of DeepSC is extended for digital modulation by quantizing the channel input without significant loss from the transmission. [86] combines pre-trained language models with conventional channel coding and modulation techniques to make the system more generalizable across different datasets. The rest of the methods in this family aim to reduce the semantic error caused by channel errors. [35] uses part-of-speech tagging to create separate codebooks, and a dynamic programming strategy assigns semantically distant words to the same codewords. Furthermore, it uses a context-based decoding strategy that relies on continuous-bag-of-words and LSTM. [87] uses the semantic distance of words to connect the semantic space with the traditional modulation constellation scheme.

In [88], a holistic model is proposed instead of concentrating on the physical layer. Power allocation is optimized based on the semantic importance of each word, which is decided by pre-trained language models [88]. Image counterparts of the models that do not require continuous data transmission used vector quantization [89], scalar quantization [90] and conceptual spaces [91].

Chapter 3

Sememe-Based Semantic Communications[1]

In this chapter, we first explained the concept of sememe in detail. We then introduced the proposed method, SememeSC. Finally, we presented our experimental results and compared them with baseline methods. Our contributions through SememeSC can be summarized as follows:

- Introduces a fundamental concept in linguistics and NLP, sememes, to the communications domain.
- Provides extensive comparisons with traditional baselines and state-of-the-art semantic communication benchmarks and reported performance improvements in AWGN and Rayleigh fading channel environments.
- Proposes a semantic communication system compatible with existing channel coding and modulation techniques.

[1]The content of this chapter reflects the work described in the following publication: T. Ozates, U. Kargı and A. Koc, "Sememe-Based Semantic Communications," in IEEE Communications Letters, vol. 28, no. 10, pp. 2308-2312, Oct. 2024.

3.1 Primer on Sememes

Sememes, the basic units of meaning in language, are vital elements that help decode the complexities of semantic expressions [49]. They are in a limited number, and the set of sememes is much smaller than a set of words in a language. It is possible to annotate elements of the latter as combinations of the former [50]. These annotations are kept in knowledge bases called SKBs. The primary example of SKBs is the HowNet, which is manually constructed by linguists [92]. However, manually constructing such a large base can take decades and automatizing it is desirable. Therefore, several methods are proposed for generating sememe annotations, given a pre-defined set of sememes, which is a task called *lexical sememe prediction* [93–96]. Another main problem related to sememe computation is constructing the set of initial sememes, which has semi-automatic [97] and automatic [98] solutions in the literature.

The first attempt for lexical sememe prediction utilizes word embeddings and matrix factorization [93]. [93] proposed a method that factorizes the word-sememe matrix generated by HowNet [92] to obtain sememe embeddings and used these embeddings to predict sememes for unlabelled words. [94] extended [93] by incorporating the dictionary definitions of words to predict new annotations. These methods predict a set of sememes instead of a tree structure. Since HowNet annotates in the latter, [96] is proposed to perform sememe prediction in this structure using transformers. Although these models successfully predicted new sememes for unannotated words, they still rely on an initial set of sememes; some even need manually constructed HowNet as a basis. Therefore, methods that can automatically construct SKBs are proposed.

EDSKB [97] is one of the bases constructed automatically from a dictionary. It does not require HowNet or the initial sememe set. However, it still needs a set of words called *controlled defining vocabulary (CDV)*, which is a set of words that is highly frequent in language. Using this set as a basis, they first constructed the sememe set with stop-word removal and some filtering. For the sememe annotation, they take the dictionary definition of words and remove the words not in the sememe set, which is repeated for every word sense. This method can be considered as a semi-automatic way to generate an SKB.

MRD2SKB [98], a recent method, can construct the sememe set and annotations without any initial set and be regarded as a fully automatic method. MRD2SKB first constructs a word-document matrix $\mathbf{D}$ from dictionary definitions. $\mathbf{D}$ is an $m \times n$ matrix, where m is the number of definitions in dictionary and n is the number of terms. $\mathbf{D}_{i,*}$ is a row vector that shows sememe annotation of the i'th definition. $\mathbf{D}_{i,j}$ is a scalar that describes the occurrence of j'th term in the i'th definition. Since n is much larger than the size of a sememe set, $\mathbf{D}$ should be reduced. In order not to lose dimensions and avoid introducing a new space of words, columns with the least semantic information are iteratively eliminated. The column to be eliminated in the k'th step can be found by:

$$\begin{gathered}\mathbf{1} \cdot \mathbf{D}^{(k)} = \mathbf{C}^{(k)}, \\ \arg\min_{i}(\mathbf{C}^{(k)}_{(1,i)}) = c^{(k)}_{min},\end{gathered} \tag{3.1}$$

where $\mathbf{1}$ is a ones matrix of size $m \times m$, $\mathbf{C}^{(l)}$ is a matrix of column sums in k'th step and $c^{(k)}_{min}$ is the index of the column with least semantic information. However, directly deleting this column will result in a serious loss of information. Therefore, if the sememe candidate exists in the definition list, an additional step is included before dropping the column:

$$\begin{gathered}c^{(k)}_{min} = c_k, \\ \mathbf{D}'^{(k)} = \mathbf{D}^{(k)} + \alpha(\mathbf{D}^{(k)}_{c_k,*} \cdot \mathbf{D}^{(k)}_{*,c_k}),\end{gathered} \tag{3.2}$$

where α is a tunable semantic relation parameter. The purpose of this step is to preserve some of the column information to be dropped. Finally, the c_k'th column is dropped after the update in Eq. 3.2. The entries of the matrix in the next step are defined by:

$$\mathbf{D}'^{(k+1)}_{i,j} = \begin{cases} \mathbf{D}'^{(k)}_{i,j}, & \text{if } j < c_k, \\ \\ \mathbf{D}'^{(k)}_{i,j+1}, & \text{if } j \geq c_k. \end{cases} \tag{3.3}$$

A stop condition is required since this procedure will delete all the columns otherwise. The stopping condition can be the minimum number of columns to keep or the minimum semantic information needed to delete a column. The resulting matrix from this procedure is the SKB obtained.

In this work, we used HowNet and MRD2SKB to observe the performance of both manually and automatically constructed bases. The number of sememes in the MRD2SKB variant we used is 3000, an essential parameter for the knowledge base corresponding to the first stop condition described above. HowNet has a sememe set with a similar size. A too-small set of sememes will result in too-dense representations, while a large set will fail to represent semantic relations [98].

3.2 Proposed SememeSC

We consider the problem of transmitting a text message from a source to a destination through a noisy channel. We work on sentence level. We denote a sentence to be transmitted m, consisting of N words as $m = [w_1, w_2, ..., w_N]$. The received sentence is denoted by $\hat{m}$. The problem is to recover a sentence $\hat{m}$ that is semantically similar to m at the receiver while maintaining transmission efficiency. Our proposed SememeSC consists of three main blocks: The Symbol Extractor, The Communication Block, and The Symbol Decoder. Fig. 3.1 provides a general overview of the framework.

3.2.1 The Symbol Extractor

Before any pre-processing step, SememeSC tags named entities (NEs) via a pre-trained named entity recognizer (NER) [99]. NE refers to a specific object, person, location, organization, or any other entity with a proper name. While SKBs include some of the NEs and express them as combinations of sememes, their decomposition is less explainable than other words as they carry little semantic content but stand for proper names. Therefore, a better approach is filtering them and sending them as individual symbols. We define the recognizer function as $NER(.)$, which takes raw sentence m as input and returns a sentence where

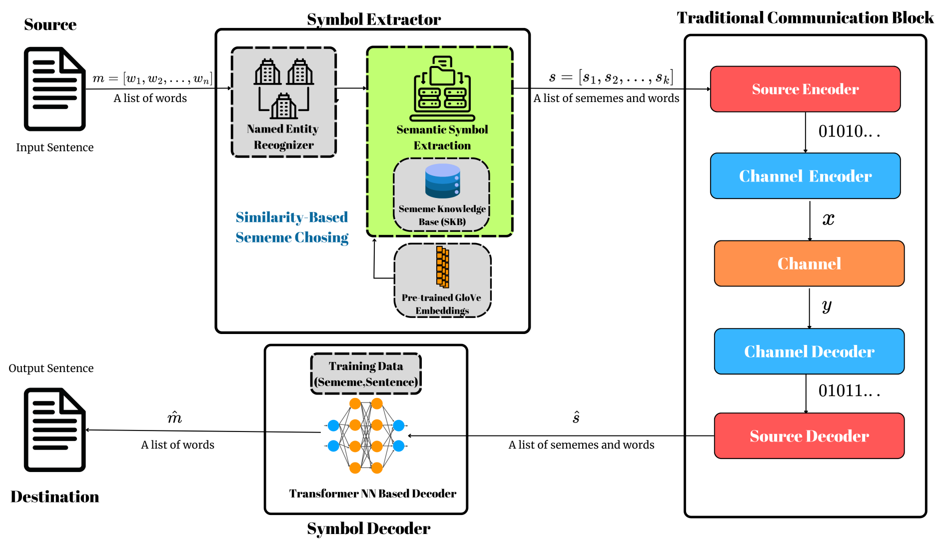

Figure 3.1: Overview of the proposed SememeSC.

NEs such as names, cities, and countries are tagged. Mapping is formulated as $\tilde{m} = NER(m)$, where $\tilde{m}$ denotes the NE-tagged sentence.

The next step is pre-processing input sentences to convert them into a proper form for sememe extraction. Given the NE-tagged input sentence $\tilde{m}$, SememeSC lower-cases the sentence first. Then, it removes punctuation and applies PoS tagging, which is the process that assigns grammatical groups to words such as nouns, verbs, and adjectives. Next, PoS-tagged sentences are tokenized and lemmatized, a procedure to get the root form of words. The aforementioned pre-processing procedures are performed using NLTK's pipelines [99]. After that, we use the English stop words list of NLTK [99], denoted by F, to detect stop words. Stop words are common words with low semantic values and are eliminated in most NLP procedures. Although we eliminate them from the sememe extraction process, we still send them as individual symbols as they are required for reconstructing the sentences. Next, we use an SKB to extract the K sememes for each token, except the stop words and NEs.

SKBs express each word with a different number of sememes. Including every element in the decomposition will significantly increase the number of bits required per word, which will conflict with the primary goal of semantic communications. Therefore, we choose at most K sememes for each word and perform a reduction on SKB. To this end, we used word similarity and chose the most similar sememes to the target word from the corresponding word's sememe set. Most similar sememes are those with the largest cosine similarity to the target word. Similarity scores are based on the cosine similarity of GloVe [100] word embeddings. It should be noted that alternative methods for measuring word similarity can also be readily used in this part of the algorithm. The reduction procedure is explained in Algorithm 1, where $G(.)$ is the function that maps words to GloVe embeddings, $v \in \mathbb{R}^{d_s \times 1}$ is the resulting embedding vector, where d_s is the embedding size. $K_b(.)$ denotes the function that returns a list of sememes from an SKB. We compute the cosine similarity of a word and its related sememes as follows:

$$c_s(w, s_k) = \frac{G(w).G(s_k)}{\|G(w)\|\|G(s_k)\|}, \tag{3.4}$$

where w is the target word, s_k is the k'th element in its sememe set V. For each word, we then take the K most similar sememes to the word according to scores computed by Eq. 3.4.

Algorithm 1 Similarity-Based Sememe Choosing

Input w, K, $G(.)$, $K_b(.)$
Output S
$V \leftarrow K_b(w)$
$v \leftarrow G(w)$
$C \leftarrow list$ ▷ Similarity Scores
for s_k in V **do**
    $g_k \leftarrow G(s_k)$
    $score \leftarrow c_s(v, g_k)$ ▷ Cosine Similarity
    $C.add(score)$
end for
Sort V according to C
$S \leftarrow V[1 : K]$

Semantic Symbol Extraction block is explained in Algorithm 2. We denote the symbol extraction method as a function $S_k(.)$ that maps an input sentence $\tilde{m}$ to decomposed sememes and semantic symbol list s, consisting of L words, which can be formulated as:

$$s = S_k(\tilde{m}), \tag{3.5}$$

where s is a list of semantic symbols such that $s = [s_1, s_2, ..., s_L]$, and $L \leq KN$. s consists of stop words, NEs, and sememes. K is the maximum number of semantic symbols per word. Some words are decomposed to less than K symbols when it is sufficient to express them to avoid redundancy. The reduced version of the SKB dictionary is denoted by $\hat{K}_b : W \rightarrow S_s$, where W is the set of words and S_s is the space of sememe sets. Fig. 3.2 shows an example of mapping from an input sentence to semantic symbols and reverting them.

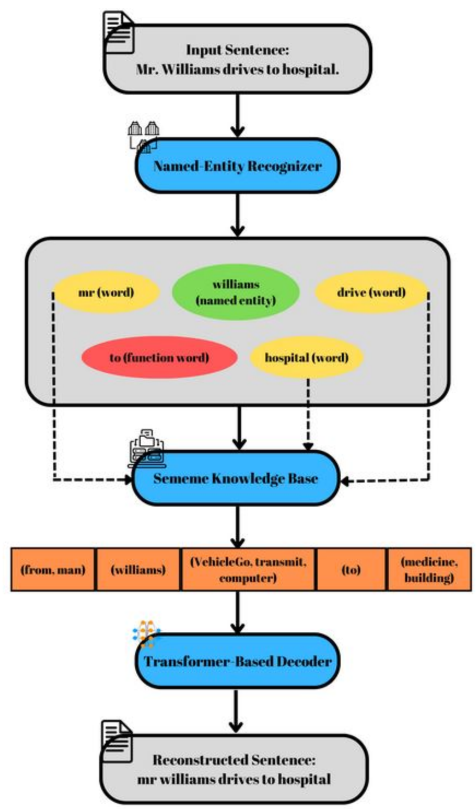

Figure 3.2: An example of symbol extraction and reconstruction of a sentence without a communication channel.

Algorithm 2 Semantic Symbol Extraction

Input m, $\tilde{m}$, F, $\hat{K}_b$
Output s

$s \leftarrow list$
for m_i in $\tilde{m}$ **do**
  if m_i is in F **then** ▷ Stop word check
    $s.add(m_i)$
  else if $w_i == m_i$ **then** ▷ Named-entity check
    $s.add(m_i)$
  else if m_i not in $\hat{K}_b$ **then**
    $s.add(m_i)$
  else
    $S_{list} \leftarrow \hat{K}_b[m_i]$
    $s.add(S_{list})$
  end if
end for

3.2.2 The Communication Block

SememeSC needs source encoding to generate bits from semantic symbols and channel coding to reliably transmit these bits through the channel. The communication block aims to perform these operations and get the semantic symbols in the receiver as correctly as possible. Thanks to its modular design, the Communication Block is compatible with most separate source and channel coding algorithms.

Source encoder is the function $S_h(.)$ that maps the input symbol list s to a length R binary sequence $b \in \{0,1\}^R$. The dictionary that the source encoder uses consists of semantic symbols that are extracted by Algorithm 2. The channel encoder outputs another binary bitstream from the source-encoded data. Variants of phase-shift keying (PSK) and quadrature amplitude modulation (QAM) are valid for digital modulation in the system. Channel encoders combined with modulation can be denoted as a function $S_m(.)$, which maps the bit sequence to communication symbols transmittable through a channel. We provided details on the choices for source coding, channel coding, and modulation scheme in Chapter 3.3.1. Mapping from semantic symbols to communication symbols can be formulated as follows:

$$x = S_m(S_h(s)), \tag{3.6}$$

where $x \in \mathbb{C}^{D \times 1}$ is the input symbol sequence to the channel. R and D vary for each transmitted sentence.

We considered two channels: the AWGN and the single-path Rayleigh fading channels, where the transmission of x can be modeled:

$$y = hx + n, \tag{3.7}$$

where $y \in \mathbb{C}^{D \times 1}$ is the channel output, $n \in \mathbb{C}^{D \times 1}$ is Gaussian noise and $h \sim \mathcal{CN}(0,1)$ is fading coefficient, and n is sampled from $\mathcal{CN}(0, \sigma^2)$. Modulated symbols in x have unit average energy. Noise standard deviation σ depends on SNR, which is defined as $SNR = 10\log_{10}(1/\sigma^2)$. h is equal to 1 for the AWGN channel.

As the last step, semantic symbols should be recovered from corrupted signal y. Demodulation and channel decoder can be combined and expressed as the inverse function of $S_m(.)$. The inverse function of $S_h(.)$ denotes the source decoder. The recovery process is formulated as follows:

$$\hat{s} = S_h^{-1}(S_m^{-1}(y)), \tag{3.8}$$

where $\hat{s}$ denotes the recovered symbols.

3.2.3 The Symbol Decoder

The last main block of the SememeSC is the Symbol Decoder, a transformer-based sequence-to-sequence model inspired by machine translation models. The Decoder takes the source output in text format, denoted by $\hat{s}$, which is a corrupted version of semantic symbols, and outputs the raw version of the input sentences. As the network takes the whole sentence as input, it can correct errors with the help of the semantic context and correctly transmitted symbols. The decoder is modeled as the inverse of $S_k(.)$, which outputs the recovered sentences $\hat{m}$, and can be formulated as follows:

$$\hat{m} = S_k^{-1}(\hat{s}). \tag{3.9}$$

We trained the Decoder using the Europarl Dataset [101], which is widely used for machine translation. Although our model is compatible with longer sentences, we discarded the sentences shorter than 4 words and longer than 30 words to create consistency with other works in the literature [14]. A split of $120,000$, $20,000$, and $25,000$ sentences for the training, validation, and test sets, respectively, is used. Sub-word tokenization is used. $T(.)$ is the tokenizer function, which maps a text sequence to a maximum length W array. Dictionary size is defined as P and batch size is B. Training objective is to minimize the cross-entropy loss, L_{CE} between $T(m) \in \{0, 1, .., P\}^{B \times W}$ and $T(s) \in \{0, 1, .., P\}^{B \times W}$. The objective can be formulated as follows:

$$L_{CE}(T(m), T(s)) = -\frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{W} T(m)_{ij} \log(T(s)_{ij}). \tag{3.10}$$

For robustness under low SNR, our network is trained by randomly replacing some portion of the symbols.

3.3 Experiments and Results

3.3.1 Experimental Setup and Baselines

We used the test set reserved during the training of the Symbol Decoder, sampled from the EuroParl [101] dataset. We used two SKBs, HowNet [50], a manually constructed SKB, and MRD2SKB [98], an automatically constructed SKB. We treated them as two different models in comparisons, denoted by SememeSC-HowNet and SememeSC-MRD2SKB, respectively. We used Huffman encoding for the source encoding. Polar [7] codes are used for channel coding since it is one of the most widely used coding schemes for 5G communication systems. Successive cancellation list decoding is applied with a list size of 32 and the coding rate is 1/3. We used quadrature phase-shift keying (QPSK) modulation. Huffman dictionary size is $3,826$ for the test set. Transformer NN consists of 3 encoder & decoder layers, 8 attention heads, and 128 units. The feed-forward size is 512, the dropout rate is 0.1, the batch size is 64, and the prediction layer size is equal to the dictionary. We used Adam optimizer with a learning rate of 10^{-4}. We trained the network for 60 epochs with early stopping. We fine-tuned the parameters with grid search.

We compared our SememeSC with the semantic communication baselines: DeepSC [14], an end-to-end DL-based baseline, and ContextSC [35]. We retrieved some of the results for ContextSC from [87]. We also compared our work with traditional communication baselines using polar codes and two source encoding strategies: Huffman and 6-bit encoding, which use characters as semantic symbols.

If we denote the total number of words in the test set as W_{total} and the total number of modulated symbols as channel input used for the transmission of the whole test set as S_{total}, $S_{average} = S_{total}/W_{total}$. To ensure a fair comparison in terms of transmission length, we reproduced DeepSC to have the same communication symbol per word, $S_{average}$. Training and test batch sizes are equal for DeepSC. We matched the number of symbols per source-encoded bits with ContextSC. We matched the number of transmitted communication symbols per word with traditional baselines by tweaking their modulation order. We used two performance metrics. The first is the METEOR (Metric for Evaluation of Translation with Explicit ORdering) score, a standard machine translation metric [102]. It evaluates the match between two sentences by considering word order and stems. We also used BERT [70] similarity, which calculates the cosine similarity between two sentence vectors extracted by BERT and is more aligned with human judgment [14]. Since this metric has an empirical lower bound, we rescaled it with a range between 0 and 1. The maximum number of sememes per word K is 3.

3.3.2 Numerical Results

We provided simulation results in the AWGN and Rayleigh fading channel settings. Fig. 3.3a and Fig. 3.3b present the simulation results in the AWGN channel for -3 dB to 10 dB SNR range for METEOR and BERT score metrics. In the -3 dB to 2 dB SNR range, SememeSC outperformed all baselines according to both metrics. When SNR reaches 2 dB, traditional Huffman starts to have an almost perfect transmission. ContextSC starts to perform better than SememeSC methods when SNR is very close to 3 dB, and for higher SNR, this trend continues. DeepSC is behind the SememeSC and ContextSC in this range. When SNR is higher than 4 dB, all baselines almost converge to their upper bound. Fig. 3.3c and Fig. 3.3d show the simulation results in the Rayleigh fading channel for 0 dB to 12 dB SNR range. Regarding the METEOR score, DeepSC performed best in the $0-7$ dB range. At 7 dB, SememeSC, DeepSC and ContextSC performed similar. When the SNR is greater than 7 dB, ContextSC has the highest performance. However, in this setting, the two metrics are not entirely correlated. DeepSC is superior to other baselines in the $0-7$ dB range. When SNR is higher than 7 dB, SememeSC and DeepSC perform very close.

In the AWGN channel and 24 symbols per word usage, DeepSC shows a poor performance compared to our method, but one needs to note that this could change with less transmitted symbol requirements. Two metrics are correlated in this setting. Two versions of SememeSC models performed closely, but MRD2SKB was better by a small margin. DeepSC's training is specific to the channel. It optimizes according to the Rayleigh fading model and learns to capture the meaning in environments with a higher bit error rate (BER) in the Rayleigh fading channel. On the other hand, SememeSC is powered by traditional methods, and its performance is more dependent on BER, making it superior in the AWGN channel but worse than DeepSC in most of the range for the Rayleigh.

SememeSC excels in capturing meaning but may overlook linguistic details in highly noisy environments, reducing direct word matches. This inconsistency between METEOR and BERT metrics in Rayleigh fading channels is likely due

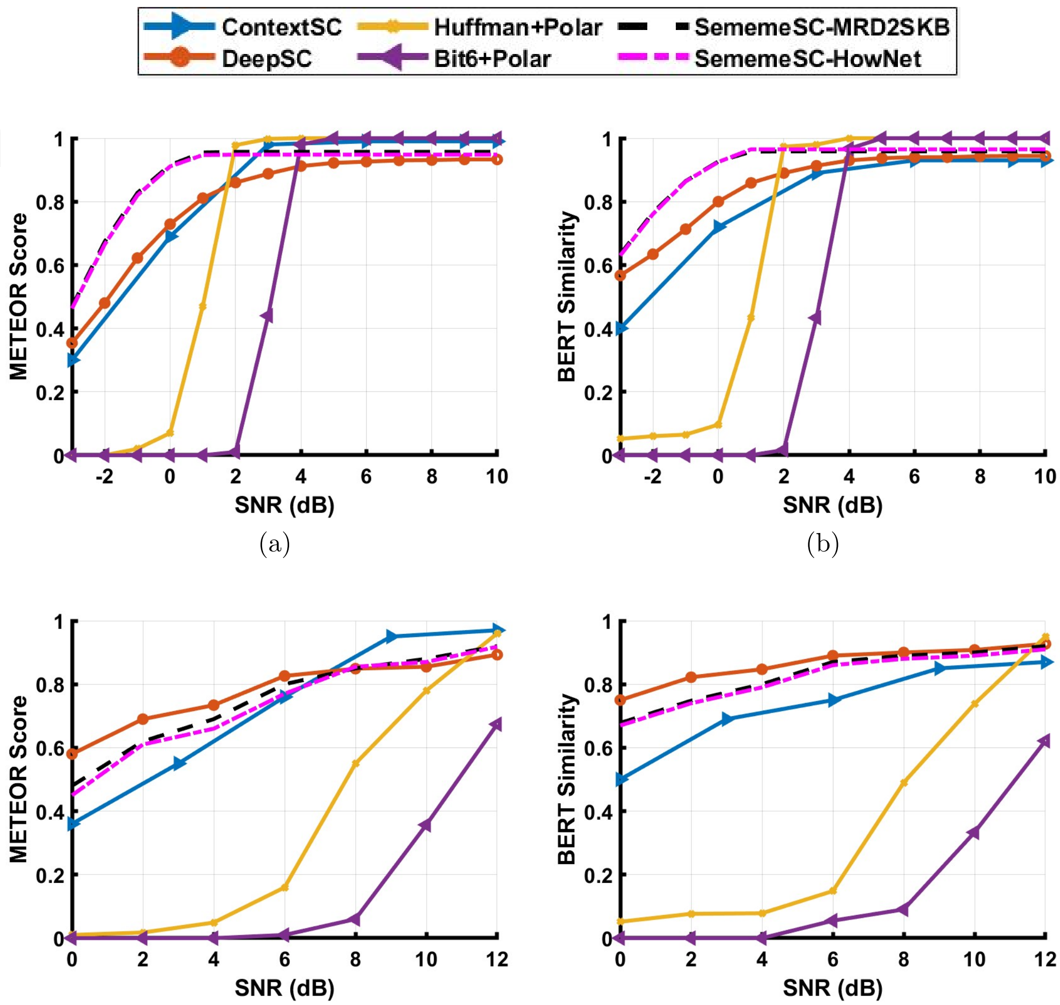

Figure 3.3: Comparison of methods. (a) METEOR on AWGN channel, (b) BERT similarity on AWGN channel, (c) METEOR on Rayleigh fading channel, (d) BERT similarity on Rayleigh fading channel.

to such limitations. However, these errors minimally impact human judgment, as indicated by BERT scores.

3.3.3 Qualitative Examples

Lastly, we present some sample transmitted and received sentences in Table 3.1. First, we gave some examples of lossless transmissions. Then, we presented the cases where the model fails to recover the input sentence exactly. We introduced the most common type of failure cases observed in the model. All example sentences are chosen from the simulation in the Rayleigh fading channel with an SNR of 10 dB, and MRD2SKB is used as the knowledge base.

The first three examples in Table 3.1 illustrate lossless transmissions. It should be noted that even in these cases we do not transmit or recover punctuation and uppercase letters as most of the text semantic communication models. The third example is quite a long sentence, a case where deep models generally struggle to recover without any loss. In the following two examples, the model confuses the tense of a verb and plural suffix, respectively. These two errors are not based on communication channel distortions. They stem from the lemmatization of the input sentences before extracting sememes. We expect a transformer-based decoder to recover such errors. While in many cases it does, sometimes it lacks external information to correct them as in these examples. The following example shows a case where a word is replaced with a similar word. This example shows the capability of our model rather than an error. A bit error probably occurred in this case, and the model managed to recover the semantic information. The following example stems from a channel distortion. Huffman decoder probably could not find the distorted sequence in the dictionary and discarded the information, resulting in a missing word. The last example is a case where a communication error occurred, and the model could not correct it, replacing the word with an unrelated word.

Table 3.1: Qualitative Results. Both positive and negative examples are provided.

Example Type	**Transmitted (TX) and Received (RX) Sentences**
Lossless I	**TX:** the document includes a timetable for the completion of the current negotiations **RX:** the document includes a timetable for the completion of the current negotiations
Lossless II	**TX:** the participation of regional and local authorities is very important **RX:** the participation of regional and local authorities is very important
Lossless III	**TX:** i think that this is the particular cause of the fear of instability instead of the stability which we want to ensure in the european union **RX:** i think that this is the particular cause of the fear of instability instead of the stability which we want to ensure in the european union
Tense Error	**TX:** when this attempt **failed** these people **were** indeed detained by the police **RX:** when this attempt **fails** these people **are** indeed detained by the police
Plural Error	**TX:** how do you assess the scenario if the legal **inventory** is completed in optimum time **RX:** how do you assess the scenario if the legal **inventories** were completed in optimum times
Rephrasing	**TX:** however enlargement of the european union must not dilute or **dissipate** the influence of smaller member states with key national interests which must be protected **RX:** however enlargement of the european union must not dilute or **abolish** the influence of small member states with key national interests which must be protected
Missing Word	**TX:** i would **also** like to say a few things about agriculture **RX:** i would like to say a few things about agriculture
Inexplicable Error	**TX:** however we should not **generalise** **RX:** however we should not **harmonise**

Chapter 4

Semantic Communication over Channels with Insertions, Deletions and Substitutions[2]

While the method proposed in Chapter 3 outperformed baselines with a 24 symbols per word budget, its performance can drop significantly with smaller bit budgets, since it still relies on the traditional architectures. Despite their drawbacks, JSCC-based methods can outperform such methods where the bit/symbol budget is small. Therefore, extending the capabilities of JSCC-based methods is still valuable, as discussed in Chapter 1. Our contributions through DeepJSOC can be summarized as follows:

- We propose the first joint semantic transmission scheme for binary channels with synchronization errors.
- We present a novel training strategy that utilizes a pre-trained error correction network in the joint-source channel coding schemes with a transfer learning approach.

[2]The content of this chapter reflects the work described in the following publication: T. Ozates and A. Koc, "Semantic Communication Over Channels With Insertions, Deletions, and Substitutions," in IEEE Communications Letters, vol. 29, no. 3, pp. 586-590, March 2025.

- We provide extensive comparisons with benchmark methods using a public dataset and achieved performance improvements in IDS channels.

In the rest of the chapter, we first described the proposed DeepJSOC. We then presented our experimental results and their comparison with baseline methods.

4.1 Proposed DeepJSOC[3]

We consider the problem of transmitting a text message from a source to a destination through a noisy channel. We work on sentence level and denote a sentence to be transmitted m, consisting of N words as $m = [w_1, w_2, ..., w_N]$. The received sentence is denoted by $\hat{m}$. The problem is to recover a sentence $\hat{m}$ that is semantically similar to m at the receiver while maintaining transmission efficiency.

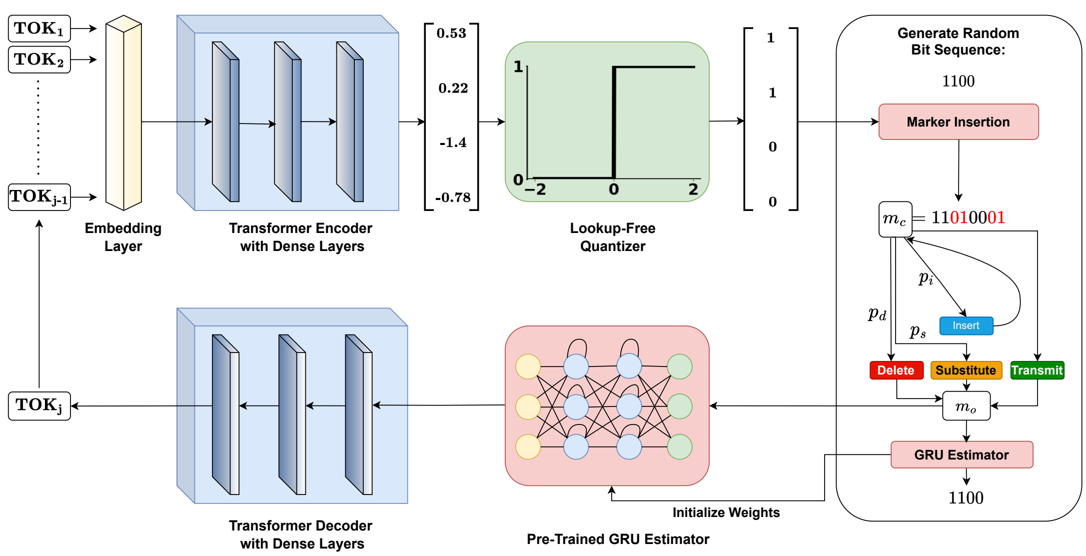

Figure 4.1: Overview of the proposed DeepJSOC.

The overall architecture of the proposed DeepJSOC is depicted in Figure 4.1. In a forward pass of the DeepJSOC, a tokenized sentence m first goes through

[3] Notation used in this chapter differs from Chapter 3 to reflect architectural differences.

the embedding layer and is encoded by a transformer encoder. Tokenization is word-based; each word is assigned to a unique token. In the testing phase, out-of-vocabulary words are assigned to the "UNK" token. Embedded representation then passes through dense layers. Let B be the batch size, C be the channel vector length, and $T_E(.)$ be the function that maps tokenized sentences to a continuous encoded space. Then, the process to obtain the encoded representations m_e can be expressed as:

$$m_e = T_E(m), \tag{4.1}$$

where $m_e \in \mathbb{R}^{B\times N\times C}$. Encoded representations are then quantized using LFQ to a binary space. LFQ is a binary quantization method enhanced by an entropy loss term. If we denote the quantizer function as $Q(.)$, with a codebook of size 2^K, quantized representations can be defined as:

$$m_q = Q(m_e), \tag{4.2}$$

where $m_q \in \{-1, 1\}^{B\times N\times(\frac{C}{K})\times K}$. The process described in Eq. 4.2 maps negative values to -1 and positive values to 1. Since quantization results in information loss, we wanted the encoder to produce diverse representations when quantized. LFQ includes an entropy loss for this purpose, which can be defined as:

$$\mathcal{L}_{\text{entropy}} = \mathbb{E}\left[\sum_{i=1}^{K} H((m_q)_{:,:,:,i})\right] - \sum_{i=1}^{K} H\left(\mathbb{E}[(m_q)_{:,:,:,i}]\right), \tag{4.3}$$

where $H(.)$ denotes the entropy function and $\mathbb{E}(.)$ denotes the expectation function. The first term of $\mathcal{L}_{entropy}$ discourages uncertain quantizations, while the second term encourages uniform usage of the codebook. We set $\hat{m}_q = (m_q + 1)/2$ for consistency with binary representation convention. We then apply marker code insertion on the quantized bits. The marker code, a pre-defined sequence of length M_n, is inserted every n bits along the vector $(\hat{m}_q)_{i,j,:}$ for each i-th batch and j-th token, where M_n and n are adjustable parameters, the marker sequence is independent of m_q. If we denote the marker encode function as $I_m(.)$ and channel input size as S, the marker encoding process becomes:

$$m_c = I_m(\hat{m}_q), \tag{4.4}$$

where $m_c \in \{0, 1\}^{B\times N\times S}$. Next, m_c goes through the IDS channel with deletion probability p_d, insertion probability p_i, and substitution probability p_s.

IDS channel operates along the channel input vectors $(m_c)_{i,j,:}$, and the output of the IDS channel for an entire batch is defined as m_o. If we assume that $k-1$ bits are received when the s-th element of the channel input vector enters to the channel $\mathcal{C}(.)$, we can define the k-th element of the output vector $(m_o)_{i,j,k} = \mathcal{C}(p_i, p_d, p_s, (m_c)_{i,j,s})$ as:

$$(m_o)_{i,j,k} = \begin{cases} \mathcal{C}(p_i, p_d, p_s, (m_c)_{i,j,s+1}) & \text{with } p_d \\ 1-(m_c)_{i,j,s} & \text{with } (1-p_d-p_i)p_s \\ \text{Bernoulli}(0.5) & \text{with } p_i \\ (m_c)_{i,j,s} & \text{with} \\ & (1-p_d-p_i)(1-p_s), \end{cases} \tag{4.5}$$

where Bernoulli(p) denotes a Bernoulli random variable with success probability p. Since the insertion and deletion operations can add or delete a varying number of bits to the input sequence, the length of the channel output is not constant. Furthermore, insertions on the sequence are not limited, making the output length unbounded. Therefore, we padded *safety bits* into the GRU estimator's input according to the channel function's expected output length. For a channel with insertion probability p_i, deletion probability p_d and input length S, we first derived the expected output length as follows:

$$\mathbb{E}[S] = n\frac{1-p_d}{1-p_i}. \tag{4.6}$$

The variance of the output length can be derived as:

$$\sigma_S^2 = \sigma_i^2 + \sigma_d^2 = np_i\frac{1-p_d}{(1-p_i)^2} + np_d(1-p_d), \tag{4.7}$$

where σ_i^2 and σ_d^2 express the variances introduced from insertion and deletion operations. Using Eq. 4.6, Eq. 4.7, and normal approximation, we find the channel output is shorter than padded sequence length $\hat{S}$ with probability p_m for the channel condition according to the following equation:

$$\hat{S} \leq S\frac{1-p_d}{1-p_i} + z_{p_m}p_i\frac{1-p_d}{(1-p_i)^2}S + p_d(1-p_d)S, \tag{4.8}$$

where z_{p_m} is the value corresponding to p_m from the normal table. Maxima of the right-hand-side (RHS) of Eq. 4.8 can be found with numerical methods for given

S and target p_i and p_d ranges to determine the *safety bits* to be added for GRU estimator. It should be noted that padding bits into the GRU estimator does not increase power consumption since these bits are added after the sequence leaves the channel. We discard excess bits if the channel output is longer than the safety bit-padded sequence.

Errors in channel output are then corrected by the pre-trained GRU estimator $E_c(.)$:

$$\hat{m}_e = E_c(m_o), \tag{4.9}$$

where $\hat{m}_e$ is analogous to encoded representation m_e. GRU estimator is a sequence-to-sequence model that takes zero-padded, length $\hat{S}$ channel output as input, fixes the synchronization errors in this sequence, and returns a length $\hat{S}$ sequence. Then, this sequence's first S bits are used as output. GRU estimator is trained in a supervised fashion using channel corrupted vectors as input and ground truth as output. It is possible to obtain a similar performance with LSTMs. However, due to their faster convergence and lower parameter count, we employed GRUs in our network, following the convention in [45] and [46]. GRU estimator does not use a textual dataset like the main network; it only needs randomly generated binary vectors. Then, the output sentence $\hat{m}$ is estimated by the transformer decoder $\hat{T}_E(.)$ as:

$$\hat{m} = \hat{T}_E(\hat{m}_e). \tag{4.10}$$

Since quantization operation and channel function are not differentiable, we used Straight-Through Estimator (STE) [103] to preserve the computational graph. During the backward pass of the network, we copy the gradients from the transformer decoder to the transformer encoder. The forward pass operates as usual. Overall loss function includes cross-entropy loss ($CE(.)$) between input and output tokens, mean-squared quantization error of the quantizer, and entropy loss of the codebook. Since the gradients are passed through the GRU estimator, we also added the mean-squared error ($MSE(.)$) between $\hat{m}_e$ and m_e as the training objective to keep GRU weights trained. Finally, the loss function is:

$$\mathcal{L}_{DeepJSOC}(m, \hat{m}) = CE(m, \hat{m}) + \mathcal{L}_{entropy} + \\ \mathbb{E}(||sg[m_e] - \hat{m}_e||_2^2 + \beta||m_e - sg[\hat{m}_e]||_2^2) + \\ \alpha MSE(\hat{m}_e, m_e), \tag{4.11}$$

where sg is stop-gradient operator used in STE, β is the commitment loss weight and α is the GRU estimator loss weight. The complete training framework is summarized in Algorithm 3, where w_n is the number of warm-start epochs.

Algorithm 3 Training of the DeepJSOC.

Input: m, w_n, p_i, p_d, S

Output: Trained model

1: **Find** maxima of the RHS of inequality in Eq. 4.8, given p_i, p_d, and S
2: **Generate** random binary vectors of size C
3: **Create** marker-inserted and corrupted batches using $I_m(\cdot)$ and $\mathcal{C}(\cdot)$ functions
4: **Construct** masking matrix $\mathbf{M}$ from the corrupted batch, where $M_{ij} = -2s_j + 1$ if $j < i$ and $M_{ij} = 0$ otherwise
5: **Train** $E_c(\cdot)$ with input $\mathbf{M}$ and output $\mathbf{C}$
6: **Train** the transformer encoder-decoder model on continuous space (Eq. 4.1 and Eq. 4.10) for w_n epochs with m using cross-entropy
7: **Initialize** GRU-estimator weights
8: **Train** the entire network end-to-end using Eq. 4.11

4.2 Experiments and Results

In our experiments, we used the Europarl dataset [101], the common benchmark in textual semantic communication. We considered two bit budgets: 24 bits per word and 36 bits per word. We used the following state-of-the-art baseline models to compare our architecture:

1. Traditional baselines: The combinations of different source and channel coding schemes followed by the BCJR algorithm. The code rate is adjusted to match the bit budget. Source codes include Huffman and fixed-length encoding. Channel codes include LDPC and Polar.

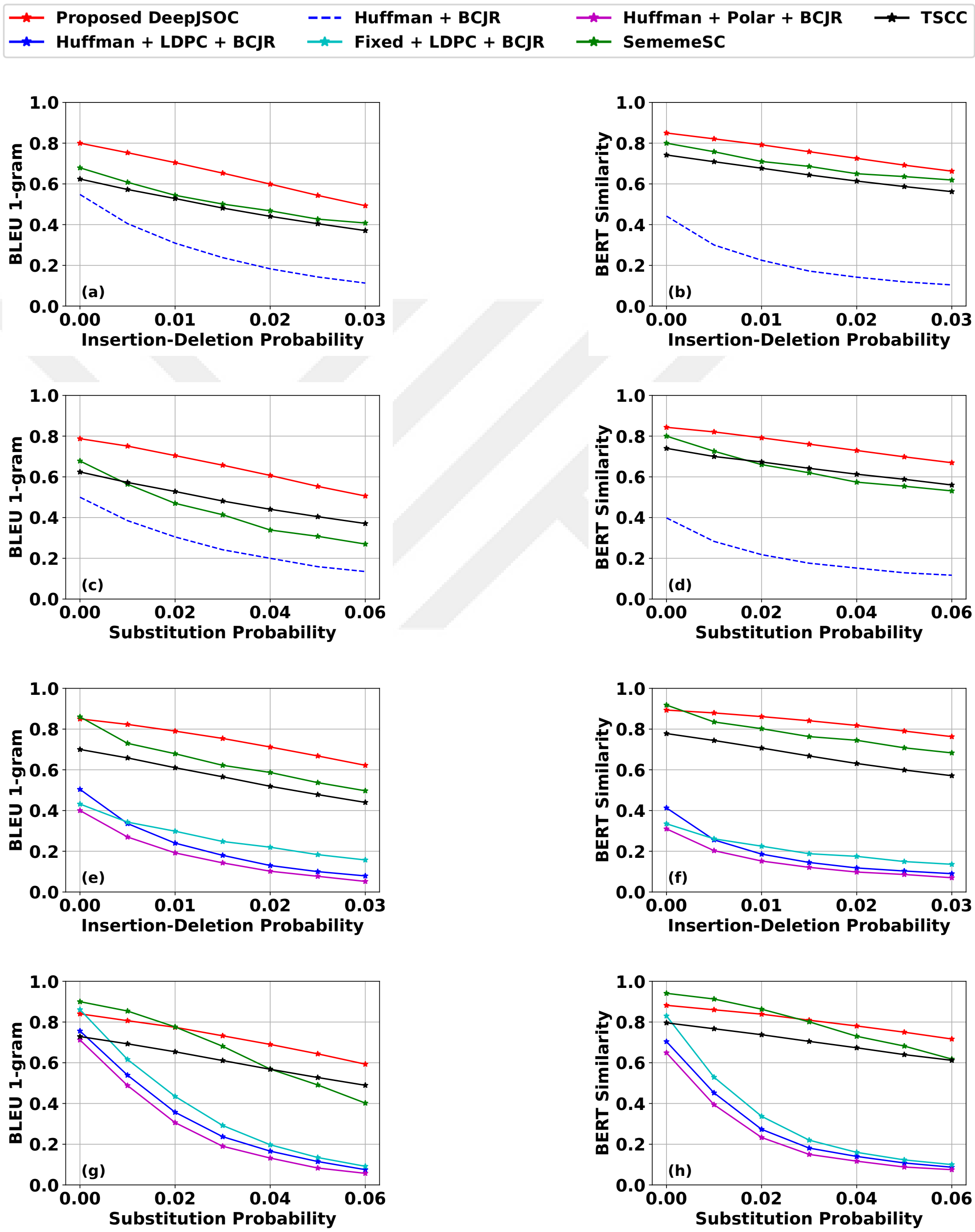

Figure 4.2: (a)-(e) BLEU-1 for $p_d \in [0, 0.03]$: $p_s = 0.02$ for the 24-bit, and $p_s = 0.03$ for the 36-bit settings. (b)-(f) BERT similarity for $p_d \in [0, 0.03]$: $p_s = 0.02$ for the 24-bit, and $p_s = 0.03$ for the 36-bit settings. (c)-(g) BLEU-1 for $p_d = 0.01$ and $p_s \in [0, 0.06]$ for both 24-bit and 36-bit settings. (d)-(h) BERT similarity for $p_d = 0.01$ and $p_s \in [0, 0.06]$ for both 24-bit and 36-bit settings.

2. SememeSC + BCJR: Semantic communication system in [1] with adding marker insertion and BCJR algorithm. The code rate is adjusted to match the bit budget.

3. TSCC: Binary joint source-channel coding system in [47].

In baselines that use markers, a marker sequence of $[0, 1]$ is inserted after every 9 bits for 36 bit case, making channel vector size 44. For the 24 bit case, the marker is inserted after every 6 bit, making the channel vector of length 32. A split of $66, 176$, $7, 424$, and $7, 296$ sentences for the training, validation, and test sets, respectively, is used. We used a learning rate of 0.001, a hidden dimension of 128, and 3 Bi-GRU layers for the GRU estimator. The estimator is trained for 200 epochs. For training, we set $p_i = p_d \sim \text{Uniform}(0, 0.02)$ and $p_s \sim \text{Uniform}(0, 0.05)$, where Uniform(.) describes continuous uniform distribution. Other detailed parameter settings for our main network are given in Table 4.1.

Table 4.1: System and Network Parameters

Parameter	Value
Transformer Encoder & Decoder Layers	4
Dense Encoder & Decoder Layers	2
Units	128
Dropout Rate	0.1
Attention Heads	8
Feedforward Size	512
Batch Size	128
Learning Rate	0.001
Warm-Start/Total Epochs	20/100
Commitment/GRU Estimator Loss Weight	0.1/0.1
Quantizer Codebook Size	64
Training p_i and p_d	Uniform (0, 0.02)
Training p_s	Uniform (0, 0.05)
Marker Sequence Length	2

We used two performance metrics. The first is the BLEU (bilingual evaluation understudy), a widely used metric in machine translation [104]. It evaluates sentences according to direct word-by-word matches, ignoring the underlying meaning. To capture the semantics, we utilized BERT [70] similarity, which is described in Chapter 3. We rescaled it to have a value between 0 and 1.

In simulations on IDS channels, we set $p_i = p_d$ in all experiments, following the convention in [47]. We expect the effect of p_i and p_d to remain unchanged for $p_i \neq p_d$ since the bit error rates of the GRU estimator and BCJR are consistent in both cases. To observe the effect of synchronization errors and substitution separately, we first set p_s as constant and observed the effects of p_i and p_d. In the second setup, we set p_i and p_d as constant and observed the effect p_s.

Figures 4.2a and 4.2b depict the performance depending on insertion-deletion probability with a bit usage of 24. We set the $p_s = 0.02$ and considered the range $p_d \in [0, 0.03]$. In this setting, DeepJSOC outperforms all baselines in both metrics. The closest baseline is SememeSC regarding BERT similarity, with a slightly inferior BLEU score. While the performance gap is very large for $p_d \in [0.005, 0.02]$, three methods converge to closer points for large p_d. Figures 4.2c and 4.2d show the performance w.r.t substitution probability for the 24-bit case. We set the $p_d = 0.01$ and considered the range $p_s \in [0, 0.06]$. For $p_s = 0$, SememeSC and DeepJSOC perform close in BERT similarity, while DeepJSOC is superior in BLEU. By introducing substitution into the system, DeepJSOC becomes superior to all models and outperforms all baselines significantly for $p_s > 0.01$. It should be noted that SememeSC is a channel-agnostic model, which is a reason for DeepJSOC's superiority against SememeSC over IDS channels. Figures 4.2e and 4.2f show the results when $p_s = 0.03$ for the 36-bit case. We consider the interval where $p_d = [0, 0.03]$. With $p_d = 0$, SememeSC has the best score for both metrics. Starting from $p_d = 0.05$, DeepJSOC has the highest BLEU score. Regarding BERT similarity, SememeSC and DeepJSOC are very close, with slightly higher scores for DeepJSOC. It is observed that traditional methods produce very low scores in both metrics for this setting.

Figure 4.2g and 4.2h present the results when $p_d = 0.01$ with a variable substitution probability in the range $[0, 0.06]$ for the 36-bit case. When $p_s = 0$, SememeSC, Fixed+LDPC+BCJR, and DeepJSOC have BLEU scores of about $0.85 - 0.9$ and a BERT similarity higher than 0.8. For $p_s < 0.02$, SememeSC continues to produce the highest scores in both metrics. As p_s increases over 0.02, we observe that DeepJSOC outperforms all baselines significantly.

An important observation is the relation between the two metrics. For the same BLEU score, DeepJSOC gives much higher BERT similarity results than traditional metrics. This result implies that the proposed system is not dependent on bit-error rate; it can capture the meaning and produce semantically similar sentences. A similar phenomenon is also observed in SememeSC and TSCC baselines.

We conducted ablation studies to quantify the contributions of the integral parts of the DeepJSOC by testing four variants:

1. Without warm-start: End-to-end network is trained from scratch instead of first training over a continuous space.
2. Without GRU: Marker-detecting network is removed.
3. Without LFQ: We removed LFQ and quantized each bit to 0 and 1 without a loss that encourages diversity.
4. Without marker: We removed marker codes and GRU completely and expected the transformer to handle errors. This is a complete joint source-channel coding system.

Ablation experiments are conducted for $p_s = 0.03$ and $p_d \in [0, 0.03]$. The channel vector is 44-bit for each transmission (36 + 8 for marker-coded schemes). The resulting BLEU scores are given in Figure 4.3. Results show that when training starts in the binary domain, about 5% of the performance is lost since it is hard to find the convergence point for the low p_d. When we remove the entropy penalty in quantization, the low p_d range is not affected, but about 8% of the score

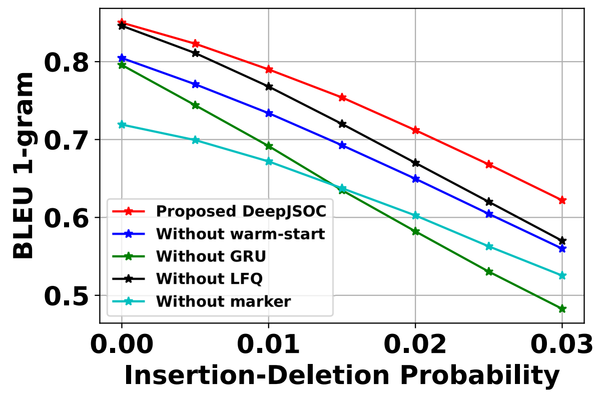

Figure 4.3: Ablation study results.

is lost as we increase p_d. We noted that with lower bit-usage, entropy penalty is essential for the low p_d range. When the GRU estimator is removed, a quite inferior performance is observed in the high p_d range. The network struggles to learn the representations if the marker code is removed, even without the channel effects.

Chapter 5

Discussion and Conclusion

In this thesis, we introduced two novel textual semantic communication methods to overcome the shortcomings of existing approaches. To be more specific, we wanted to make these systems compatible with existing communication algorithms. Furthermore, we aim to extend them to binary channels with synchronization errors. While the proposed schemes yielded promising results, the feasibility of the employment of these systems may be limited to specific cases as other textual semantic communication systems.

In the first study, we proposed a semantic source encoding and decoding strategy that relies on sememes and DL. The proposed SememeSC can be integrated into conventional communication systems more easily than end-to-end systems. We have numerically demonstrated that our method is superior to existing traditional communication methods and semantic communication baselines in AWGN and Rayleigh fading channels in the symbol-per-word rate we work on. We also showed that training the decoder module to mimic the errors in the communication channel makes the system more robust, and using a wider SKB yields slightly better results. One limitation of our proposed method is that it may perform worse with a very low symbol-per-word rate than end-to-end deep learning-based models. Another limitation is that our time complexity is higher than that of a character-based traditional method due to increased Huffman codebook sizes.

In the second study, we proposed a textual semantic communication system that is robust to insertions, deletions, and substitutions. The proposed method is the first step towards adapting deep learning-enabled semantic communication systems to binary channels with synchronization errors. It outperformed all baselines in most of the considered range. With a tiny bit budget, DeepJSOC's superiority is more significant. When the bit budget is increased, traditional methods tend to close the gap; however, they cannot compress the data as much as the proposed system. One drawback of the proposed system is the slower inference time compared to models like DeepSC since additional modules are integrated into transformer-based backbone.

One possible future work is to investigate the incorporation of sememe knowledge into end-to-end DL-based systems. For the channels with synchronization errors, it is also possible to extend the proposed idea to image semantic communication systems. Another research direction for the IDS channels may be excluding marker codes from the system to fully utilize DL in the end-to-end process. While our initial efforts showed the failure of such a setup, using different optimization techniques such as policy gradient methods instead of STE may be helpful in optimizing the parameters to learn the error patterns.

We believe that the proposed SememeSC and DeepJSOC will contribute to the practical applicability of semantic communication systems in real-world networks and pave the way for new research in this direction.

Bibliography

[1] T. Ozates, U. Kargı, and A. Koc, "Sememe-based semantic communications," *IEEE Communications Letters*, vol. 28, no. 10, pp. 2308–2312, 2024.

[2] T. Ozates and A. Koc, "Semantic communication over channels with insertions, deletions, and substitutions," *IEEE Communications Letters*, vol. 29, no. 3, pp. 586–590, 2025.

[3] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[4] W. Weaver, "Recent contributions to the mathematical theory of communication," *ETC: A Review of General Semantics*, vol. 10, no. 4, pp. 261–281, 1953.

[5] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[6] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[7] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[8] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *2011 IEEE Network Science Workshop*, pp. 110–117, 2011.

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

[10] 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 Communications Surveys Tutorials*, vol. 25, no. 1, pp. 213–250, 2023.

[11] K. Lu, Q. Zhou, R. Li, Z. Zhao, X. Chen, J. Wu, and H. Zhang, "Rethinking modern communication from semantic coding to semantic communication," *IEEE Wireless Communications*, vol. 30, no. 1, pp. 158–164, 2023.

[12] K. Niu, J. Dai, S. Yao, S. Wang, Z. Si, X. Qin, and P. Zhang, "A paradigm shift toward semantic communications," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 113–119, 2022.

[13] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2326–2330, 2018.

[14] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[15] Q. Zhou, R. Li, Z. Zhao, C. Peng, and H. Zhang, "Semantic communication with adaptive universal transformer," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 453–457, 2022.

[16] X. Peng, Z. Qin, D. Huang, X. Tao, J. Lu, G. Liu, and C. Pan, "A robust deep learning enabled semantic communication system for text," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 2704–2709, 2022.

[17] S. Yao, K. Niu, S. Wang, and J. Dai, "Semantic coding for text transmission: An iterative design," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 4, pp. 1594–1603, 2022.

[18] J. Dai, P. Zhang, K. Niu, S. Wang, Z. Si, and X. Qin, "Communication beyond transmitting bits: Semantics-guided source and channel coding," *IEEE Wireless Communications*, vol. 30, no. 4, pp. 170–177, 2023.

[19] H. Zhang, S. Shao, M. Tao, X. Bi, and K. B. Letaief, "Deep learning-enabled semantic communication systems with task-unaware transmitter and dynamic data," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 170–185, 2023.

[20] J. Xu, T.-Y. Tung, B. Ai, W. Chen, Y. Sun, and D. Gündüz, "Deep joint source-channel coding for semantic communications," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 42–48, 2023.

[21] J. Dai, S. Wang, K. Tan, Z. Si, X. Qin, K. Niu, and P. Zhang, "Nonlinear transform source-channel coding for semantic communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 8, pp. 2300–2316, 2022.

[22] Y. Shao and D. Gunduz, "Semantic communications with discrete-time analog transmission: A papr perspective," *IEEE Wireless Communications Letters*, vol. 12, no. 3, pp. 510–514, 2023.

[23] C. Dong, H. Liang, X. Xu, S. Han, B. Wang, and P. Zhang, "Semantic communication system based on semantic slice models propagation," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 202–213, 2023.

[24] W. Lin, Y. Yan, L. Li, Z. Han, and T. Matsumoto, "Semantic-forward relaying: A novel framework toward 6g cooperative communications," *IEEE Communications Letters*, vol. 28, no. 3, pp. 518–522, 2024.

[25] S. Wang, J. Dai, Z. Liang, K. Niu, Z. Si, C. Dong, X. Qin, and P. Zhang, "Wireless deep video semantic transmission," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 214–229, 2023.

[26] Y. Duan, Q. Du, X. Fang, Z. Xie, Z. Qin, X. Tao, C. Pan, and G. Liu, "Multimedia semantic communications: Representation, encoding and transmission," *IEEE Network*, vol. 37, no. 1, pp. 44–50, 2023.

[27] Z. Weng, Z. Qin, and G. Y. Li, "Semantic communications for speech signals," in *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, 2021.

[28] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2434–2444, 2021.

[29] Z. Weng, Z. Qin, X. Tao, C. Pan, G. Liu, and G. Y. Li, "Deep learning enabled semantic communications with speech recognition and synthesis," *IEEE Transactions on Wireless Communications*, vol. 22, no. 9, pp. 6227–6240, 2023.

[30] Z. Zhou, S. Zheng, J. Chen, Z. Zhao, and X. Yang, "Speech semantic communication based on swin transformer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 3, pp. 756–768, 2024.

[31] Z. Weng, Z. Qin, and G. Y. Li, "Semantic communications for speech recognition," in *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2021.

[32] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4774–4778, 2019.

[33] F. Zhou, Y. Li, X. Zhang, Q. Wu, X. Lei, and R. Q. Hu, "Cognitive semantic communication systems driven by knowledge graph," in *ICC 2022 - IEEE International Conference on Communications*, pp. 4860–4865, 2022.

[34] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142–153, 2021.

[35] Y. Zhang, H. Zhao, J. Wei, J. Zhang, M. F. Flanagan, and J. Xiong, "Context-based semantic communication via dynamic programming," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 3, pp. 1453–1467, 2022.

[36] H. G. Olanrewaju and W. O. Popoola, “Effect of synchronization error on optical spatial modulation,” *IEEE Transactions on Communications*, vol. 65, no. 12, pp. 5362–5374, 2017.

[37] I. Tinnirello, D. Croce, N. Galioto, D. Garlisi, and F. Giuliano, “Cross-technology wifi/zigbee communications: Dealing with channel insertions and deletions,” *IEEE Communications Letters*, vol. 20, no. 11, pp. 2300–2303, 2016.

[38] F. Sala, C. Schoeny, N. Bitouzé, and L. Dolecek, “Synchronizing files from a large number of insertions and deletions,” *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2258–2273, 2016.

[39] R. White, R. Newt, and R. Pease, “Patterned media: A viable route to 50 gbit/in/sup 2/ and up for magnetic recording?,” *IEEE Transactions on Magnetics*, vol. 33, no. 1, pp. 990–995, 1997.

[40] Y. M. Chee, H. M. Kiah, A. Vardy, V. K. Vu, and E. Yaakobi, “Coding for racetrack memories,” *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7094–7112, 2018.

[41] I. Shomorony and R. Heckel, “Information-theoretic foundations of DNA data storage,” *Foundations and Trends in Communications and Information Theory*, vol. 19, no. 1, pp. 1–106, 2022.

[42] M. Davey and D. Mackay, “Reliable communication over channels with insertions, deletions, and substitutions,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 687–698, 2001.

[43] F. Wang, D. Fertonani, and T. M. Duman, “Symbol-level synchronization and LDPC code design for insertion/deletion channels,” *IEEE Transactions on Communications*, vol. 59, no. 5, pp. 1287–1297, 2011.

[44] J. Luo, L. Mu, Y. Huang, Y. Yan, G. Han, and Y. Zhong, “Two rrns-based error correction schemes for dna storage channels,” *IEEE Communications Letters*, vol. 28, no. 12, pp. 2729–2733, 2024.

[45] G. Ma, X. Jiao, J. Mu, H. Han, and Y. Yang, "Deep learning-based detection for marker codes over insertion and deletion channels," *IEEE Transactions on Communications*, vol. 72, no. 10, pp. 5945–5959, 2024.

[46] E. U. Kargı and T. M. Duman, "A deep learning based decoder for concatenated coding over deletion channels," in *ICC 2024 - IEEE International Conference on Communications*, pp. 2797–2802, 2024.

[47] S. Liu, Z. Gao, G. Chen, Y. Su, and L. Peng, "Transformer-based joint source channel coding for textual semantic communication," in *2023 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1–6, 2023.

[48] S. M. Lamb, "The sememic approach to structural semantics 1," *American anthropologist*, vol. 66, no. 3, pp. 57–78, 1964.

[49] L. Bloomfield, "A set of postulates for the science of language," *Language*, vol. 2, no. 3, pp. 153–164, 1926.

[50] F. Qi, C. Yang, Z. Liu, Q. Dong, M. Sun, and Z. Dong, "Openhownet: An open sememe-based lexical knowledge base," 2019.

[51] F. Qi, R. Xie, Y. Zang, Z. Liu, and M. Sun, "Sememe knowledge computation: a review of recent advances in application and expansion of sememe knowledge bases," *Frontiers of Computer Science*, vol. 15, p. 155327, Apr 2021.

[52] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, A. Gupta, X. Gu, and A. G. H. et al., "Language model beats diffusion - tokenizer is key to visual generation," in *The Twelfth Int. Conf. on Learning Representations*, 2024.

[53] B. Güler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, 2018.

[54] Y. Wang, Z. Gao, D. Zheng, S. Chen, D. Gündüz, and H. V. Poor, "Transformer-empowered 6g intelligent networks: From massive mimo

processing to semantic communication," *IEEE Wireless Communications*, vol. 30, no. 6, pp. 127–135, 2023.

[55] 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 Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, 2023.

[56] H. Zhang, H. Wang, Y. Li, K. Long, and A. Nallanathan, "Drl-driven dynamic resource allocation for task-oriented semantic communication," *IEEE Transactions on Communications*, vol. 71, no. 7, pp. 3992–4004, 2023.

[57] T. Van Chien, L. H. Phong, D. X. Phuc, and N. T. Hoa, "Image restoration under semantic communications," in *2022 International Conference on Advanced Technologies for Communications (ATC)*, pp. 332–337, 2022.

[58] Y. Yang, C. Guo, F. Liu, L. Sun, C. Liu, and Q. Sun, "Semantic communications with artificial intelligence tasks: Reducing bandwidth requirements and improving artificial intelligence task performance," *IEEE Industrial Electronics Magazine*, vol. 17, no. 3, pp. 4–13, 2023.

[59] S. Barbarossa, D. Comminiello, E. Grassucci, F. Pezone, S. Sardellitti, and P. Di Lorenzo, "Semantic communications based on adaptive generative models and information bottleneck," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 36–41, 2023.

[60] H. Xie, Z. Qin, and G. Y. Li, "Semantic communication with memory," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 8, pp. 2658–2669, 2023.

[61] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2584–2597, 2022.

[62] H. Xie, Z. Qin, and G. Y. Li, "Task-oriented multi-user semantic communications for vqa," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 553–557, 2022.

[63] B. Zhao, H. Xing, X. Wang, Z. Xiao, and L. Xu, "Classification-oriented distributed semantic communication for multivariate time series," *IEEE Signal Processing Letters*, vol. 30, pp. 369–373, 2023.

[64] E. Kutay and A. Yener, "Semantic text compression for classification," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1368–1373, 2023.

[65] Y. Sheng, F. Li, L. Liang, and S. Jin, "A multi-task semantic communication system for natural language processing," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pp. 1–5, 2022.

[66] C. Liu, C. Guo, Y. Yang, W. Ni, Y. Zhou, L. Li, and T. Q. S. Quek, "Explainable semantic communication for text tasks," *IEEE Internet of Things Journal*, vol. 11, no. 24, pp. 39820–39833, 2024.

[67] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[68] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[69] Y. Jia, Z. Huang, K. Luo, and W. Wen, "Lightweight joint source-channel coding for semantic communications," *IEEE Communications Letters*, vol. 27, no. 12, pp. 3161–3165, 2023.

[70] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[71] B. Tang, Q. Li, L. Huang, and Y. Yin, "Text semantic communication systems with sentence-level semantic fidelity," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2023.

[72] P. Luo, H. Zhao, K. Cao, Y. Liu, Y. Zhang, and J. Wei, "Emotion-aided semantic communication system for reliable semantic recovery under low snr," *IEEE Communications Letters*, vol. 28, no. 3, pp. 503–507, 2024.

[73] B. Wang, R. Li, J. Zhu, Z. Zhao, and H. Zhang, "Knowledge enhanced semantic communication receiver," *IEEE Communications Letters*, vol. 27, no. 7, pp. 1794–1798, 2023.

[74] H. Hu, X. Zhu, F. Zhou, W. Wu, R. Q. Hu, and H. Zhu, "One-to-many semantic communication systems: Design, implementation, performance evaluation," *IEEE Communications Letters*, vol. 26, no. 12, pp. 2959–2963, 2022.

[75] M.-K. Chang, C.-T. Hsu, and G.-C. Yang, "Gensc: Generative semantic communication systems using bart-like model," *IEEE Communications Letters*, vol. 28, no. 10, pp. 2298–2302, 2024.

[76] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, "Resource allocation for text semantic communications," *IEEE Wireless Communications Letters*, vol. 11, no. 7, pp. 1394–1398, 2022.

[77] D. Huang, X. Tao, F. Gao, and J. Lu, "Deep learning-based image semantic coding for semantic communications," in *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2021.

[78] D. Huang, F. Gao, X. Tao, Q. Du, and J. Lu, "Toward semantic communications: Deep learning-based image semantic coding," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 55–71, 2023.

[79] E. Erdemir, T.-Y. Tung, P. L. Dragotti, and D. Gündüz, "Generative joint source-channel coding for semantic image transmission," *IEEE Journal on Selected Areas in Communications*, vol. 41, pp. 2645–2657, Aug 2023.

[80] C. Liang, D. Li, Z. Lin, and H. Cao, "Selection-based image generation for semantic communication systems," *IEEE Communications Letters*, vol. 28, no. 1, pp. 34–38, 2024.

[81] H. Yoo, T. Jung, L. Dai, S. Kim, and C.-B. Chae, "Demo: Real-time semantic communications with a vision transformer," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–2, 2022.

[82] F. Zhou, Y. Li, M. Xu, L. Yuan, Q. Wu, R. Q. Hu, and N. Al-Dhahir, "Cognitive semantic communication systems driven by knowledge graph: Principle, implementation, and performance evaluation," *IEEE Transactions on Communications*, vol. 72, no. 1, pp. 193–208, 2024.

[83] S. Jiang, Y. Liu, Y. Zhang, P. Luo, K. Cao, J. Xiong, H. Zhao, and J. Wei, "Reliable semantic communication system enabled by knowledge graph," *Entropy*, vol. 24, no. 6, p. 846, 2022.

[84] Z. Zhao, Z. Yang, Y. Hu, L. Lin, and Z. Zhang, "Semantic information extraction for text data with probability graph," in *2023 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, pp. 1–6, 2023.

[85] Y. Wang, M. Chen, T. Luo, W. Saad, D. Niyato, H. V. Poor, and S. Cui, "Performance optimization for semantic communications: An attention-based reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2598–2613, 2022.

[86] J.-H. Lee, D.-H. Lee, E. Sheen, T. Choi, and J. Pujara, "Seq2seq-sc: End-to-end semantic communication systems with pre-trained language model," in *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 260–264, 2023.

[87] Y. Zhang, H. Zhao, K. Cao, L. Zhou, Z. Wang, Y. Liu, and J. Wei, "A highly reliable encoding and decoding communication framework based on semantic information," *Digital Communications and Networks*, 2023.

[88] S. Guo, Y. Wang, S. Li, and N. Saeed, "Semantic importance-aware communications using pre-trained language models," *IEEE Communications Letters*, vol. 27, no. 9, pp. 2328–2332, 2023.

[89] Q. Fu, H. Xie, Z. Qin, G. Slabaugh, and X. Tao, "Vector quantized semantic communication system," *IEEE Wireless Communications Letters*, vol. 12, no. 6, pp. 982–986, 2023.

[90] J. Park, Y. Oh, S. Kim, and Y.-S. Jeon, "Joint source-channel coding for channel-adaptive digital semantic communications," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 1, pp. 75–89, 2025.

[91] D. Wheeler, E. E. Tripp, and B. Natarajan, "Semantic communication with conceptual spaces," *IEEE Communications Letters*, vol. 27, no. 2, pp. 532–535, 2023.

[92] Z. Dong and Q. Dong, "Hownet - a hybrid language and knowledge resource," in *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, pp. 820–824, 2003.

[93] R. Xie, X. Yuan, Z. Liu, and M. Sun, "Lexical sememe prediction via word embeddings and matrix factorization," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, p. 4200–4206, AAAI Press, 2017.

[94] W. Li, X. Ren, D. Dai, Y. Wu, H. Wang, and X. Sun, "Sememe prediction: Learning semantic knowledge from unstructured textual wiki descriptions," 2018.

[95] M. Bai, P. Lv, and X. Long, "Lexical sememe prediction with rnn and modern chinese dictionary," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 825–830, 2018.

[96] Y. Ye, F. Qi, Z. Liu, and M. Sun, "Going "deeper": Structured sememe prediction via transformer with tree attention," in *Findings of the Association for Computational Linguistics: ACL 2022* (S. Muresan, P. Nakov,

and A. Villavicencio, eds.), (Dublin, Ireland), pp. 128–138, Association for Computational Linguistics, May 2022.

[97] F. Qi, Y. Chen, F. Wang, Z. Liu, X. Chen, and M. Sun, "Automatic construction of sememe knowledge bases via dictionaries," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (C. Zong, F. Xia, W. Li, and R. Navigli, eds.), (Online), pp. 4673–4686, Association for Computational Linguistics, Aug. 2021.

[98] O. M. Battal and A. Koc, "Automatic construction of sememe knowledge bases from machine readable dictionaries," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1023–1035, 2024.

[99] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* O'Reilly Media, Inc., 2009.

[100] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[101] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of Machine Translation Summit X: Papers*, (Phuket, Thailand), pp. 79–86, Sept. 13-15 2005.

[102] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, June 2005.

[103] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.

[104] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th*

Annual Meeting on Association for Computational Linguistics, ACL '02, (USA), p. 311–318, Association for Computational Linguistics, 2002.