Deep Source-Channel Coding for Sentence Semantic Transmission With HARQ

Peiwen Jiang, *Graduate Student Member, IEEE*, Chao-Kai Wen, *Senior Member, IEEE*, Shi Jin, *Senior Member, IEEE*, and Geoffrey Ye Li, *Fellow, IEEE*

***Abstract*— Recently, semantic communication has been brought to the forefront because deep learning (DL)-based methods, such as Transformer, have achieved great success in semantic extraction. Although semantic communication has been successfully applied in sentence transmission to reduce semantic errors, the existing architecture is usually fixed in terms of codeword length and inefficient and inflexible for varying sentence lengths. In this study, we exploit hybrid automatic repeat request (HARQ) to reduce the semantic transmission error further. We combine semantic coding (SC) with Reed-Solomon (RS) channel coding and HARQ (called SC-RS-HARQ). SC-RS-HARQ exploits the superiority of SC and the reliability of conventional methods successfully. Although SC-RS-HARQ can be easily applied in existing HARQ systems, we also develop an end-to-end architecture called SCHARQ to pursue enhanced performance. Numerical results demonstrate that SCHARQ significantly reduces the required number of bits for semantic sentence transmission and the sentence error rate. We also attempt to replace error detection from cyclic redundancy check to a similarity detection network called Sim32 to allow the receiver to reserve wrong sentences with similar semantic information and conserve transmission resources.**

***Index Terms*— HARQ, semantic communication, quantization, sentence similarity, joint source-channel coding, transformer.**

I. Introduction

AMONG deep learning (DL)-based methods utilized in the physical layer of communication [1]–[3], joint design [4]–[12] has become a potential direction to outperform the conventional communication structure. At outset, a fully-connected network can be used to replace channel estimation and signal detection at the receiver [4]. To optimize the transmitter and the receiver together, the pilot can also be jointly designed with channel estimation to reduce its overhead [5], and the receiver can be jointly optimized through beamforming [6] and precoding [7]. Moreover, the entire network [8] can be combined with different modules, including pilot design, channel estimation, and channel information state feedback. The whole conventional communication system can be jointly designed to improve their performance, such as DL-based joint encoder-decoder [9]–[11]. In addition, several DL-based methods [13], [14] can enhance the hybrid automatic repeat request (HARQ).

The great success of DL has made many semantic tasks possible [15], [16]. As a result, semantic communication [17]–[20] has been brought to the forefront. Traditional communication systems concentrate on bit- or symbol-level performance. Semantic communication focuses on transmitting the desired meaning, which is regarded as second-level communication [21]. Semantic communication usually transcends the traditional Shannon's paradigm because the separated source and channel coding approach is not always optimal in practice. The semantic counterparts of Shannon's source and channel coding theorems were investigated in [22], and the results revealed that semantic communication is content-related. Thus, shared and local knowledge can help in the joint design of source and channel coding and improve transmission efficiency.

Joint source and channel coding has been demonstrated as an effective framework for semantic communication, and has been applied in image [23], [24], video [25], speech [26], and text [27], [28] transmission. Several architectures for DL-based semantic communication can outperform those based on traditional communication in certain semantic metrics, especially when communication resources are limited and noise is high. At the outset, fully connected and convolutional neural networks have been initially exploited for specific transmission tasks, such as image transmission [23], [24]. DL-based joint source and channel coding improves the peak signal-to-noise ratio according to [23], and high image classification accuracy was reported by [24]. Shared knowledge of the transmission task is implicitly stored in the trained weights of the neural networks and makes DL-based semantic methods better than conventional encoding methods, which do not exploit semantic knowledge.

Semantic sentence transmission is a focal topic in semantic communication. Many state-of-the-art methods have been

Manuscript received 9 September 2021; revised 12 January 2022, 29 March 2022, and 20 May 2022; accepted 30 May 2022. Date of publication 8 June 2022; date of current version 16 August 2022. This work was supported by the National Nartrual Science Foundation of China (NSFC) under 61941104 Grants 61921004, and the Key Research and Development Program of Shandong Province under Grant 2020CXGC010108. The work of C.-K. Wen was supported in part by the Ministry of Science and Technology of Taiwan under grant MOST 108-2628-E-110 -001 -MY3 and by Qualcomm through the Taiwan University Research Collaboration Project. The associate editor coordinating the review of this article and approving it for publication was C.-H. Lee. *(Corresponding author: Shi Jin.)*

Peiwen Jiang and Shi Jin are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: peiwenjiang@seu.edu.cn; jinshi@seu.edu.cn).

Chao-Kai Wen is with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail: chaokai.wen@mail.nsysu.edu.tw).

Geoffrey Ye Li is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: geoffrey.li@imperial.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2022.3180997.

Digital Object Identifier 10.1109/TCOMM.2022.3180997

proposed, and semantic metrics have been defined. In [27], a long short-term memory (LSTM) architecture was exploited for sentence transmission under the erasure channel, and its superiority was demonstrated under a high bit drop rate. The semantic transceiver in [28], which is called DeepSC, is based on Transformer [29] and has been proved to be better than RNN in understanding the meaning of text sentences. DeepSC was extended to the Internet of Things in [30], where many practical issues, such as channel impact, quantization, and network compression, were considered. Given that a fixed length code is unsuitable for sentences that have different lengths, a variable-length code based on LSTM was proposed in [31] and determined to perform better than that in [27] under long sentences. The performance measures of semantic sentence transmission, such as required bits and transmission accuracy, can be further improved by introducing state-of-the-art semantic technologies, including the universal Transformer [32]. Although the universal Transformer can adapt to different channel conditions, it still cannot dynamically change it code length according to the channel and its flexibility is limited. How to combine semantic transmission and conventional transmission is an urgent issue and the focus of this work. For example, HARQ, which was developed for conventional communication, can guarantee that transmitted packets are received correctly according to acknowledgment (ACK) feedback, and it is an essential part of a reliable transmission system. In semantic transmission, semantic similarity is more important than the bit error in conventional communication. Thus, the impact of semantic coding on HARQ should be examined.

In this study, we focus on semantic coding based on Transformer for semantic transmission of text sentences and propose a variable-length code based on Transformer. HARQ is also introduced for further performance improvement. This work is different from previous studies. First, our semantic coding (SC) network for source coding is combined with the conventional Reed-Solomon (RS) channel coding and HARQ. Second, an end-to-end autoencoder called SCHARQ is introduced to improve the transmission efficiency and reduce the sentence-error rate (SER) under a high bit error rate (BER). Lastly, we replace the conventional error detection method with a Transformer-based network called Sim32 to detect the meaning error in estimated sentences. The major contributions of this work are summarized as follows:

1) To improve the reliability of semantic sentence transmission, we combine SC with conventional RS channel coding and HARQ, and call the resulting architecture SC-RS-HARQ. The code length of SC can be changed in accordance with the length of the sentences, thereby enabling SC to perform better than methods with a fixed code length when the required average number of bits is the same. The proposed SC helps reduce SER under high BER and always has few wrong words under low BER because DL has no mechanism to guarantee its performance when testing. The proposed parallel SC-RS-HARQ exploits the advantages of the semantic architecture and conventional RS code and outperforms competing semantics-based and conventional methods in terms of word error rate (WER), SER, and bit consumption.

2) Although the proposed SC is easily applied in conventional HARQ systems because only source coding is replaced, we attempt to reduce SER and the required bits further when transmitting a long sentence in an end-to-end manner. A Transformer-based joint source and channel coding called SCHARQ is proposed. SCHARQ is more flexible than existing RNN- and Transformer-based sentence transmission methods because it can transmit incremental bits in accordance with HARQ. The code length of SCHARQ is controlled by the requirement of the receiver; hence, it is more efficient than competing methods in transmitting sentences with different lengths. SCHARQ can also cope with high BER better than separate designs can.

3) To fully exploit the potential of the proposed semantic coding methods, we introduce a network called Sim32 to detect the meaning error in the received sentences. This error detection enables the received sentences to tolerate several error words as long as their meaning is unchanged. Sim32 can conserve transmit resources because many lossy sentences can be received without requiring retransmission. However, the proposed error detection network still makes mistakes. For example, the replacement of nouns may be ignored by Sim32.

The remainder of this paper is organized as follows. Section II introduces the system model, including conventional RS encoder, HARQ, and classic DL-based autoencoder architectures. The proposed networks are shown and discussed in Section III. In Section IV, we demonstrate the superiority of the proposed networks in terms of SER and the required number of bits. Section V concludes this study.

II. System Model and Related Work

In this section, we introduce a HARQ method based on the RS code for sentence transmission. Then, we describe several existing DL-based end-to-end transmission methods and discuss the challenges in combining deep semantic networks with the HARQ system.

A. HARQ System for Sentence Transmission

To transmit sentence $\mathbf{s}$, the conventional transmitter converts it into bits via source and channel coding, and yields

$$\mathbf{b} = C_\beta(S_\alpha(\mathbf{s})) \tag{1}$$

where $S_\alpha(\cdot)$ and $C_\beta(\cdot)$ denote the source encoder through the α algorithm and channel encoder through the β algorithm, respectively. A sentence consists of $L_\mathbf{s}$ words, which are represented by serial numbers in the dictionary, namely, $\mathbf{s} = [w_1, w_2, \cdots, w_{L_\mathbf{s}}]$ and w_i is a positive integer. The recovered bits, $\hat{\mathbf{b}}$, at the receiver may be different from those at the transmitter due to channel distortion. The received sentence is decoded as

$$\hat{\mathbf{s}} = S_\alpha^{-1}(C_\beta^{-1}(\hat{\mathbf{b}})) \tag{2}$$

where $S_\alpha^{-1}(\cdot)$ and $C_\beta^{-1}(\cdot)$ represent the source and channel decoders with their corresponding algorithms α and β, respectively.

For wireless communication systems, incremental redundancy HARQ (IR-HARQ) adjusts its code rate when errors

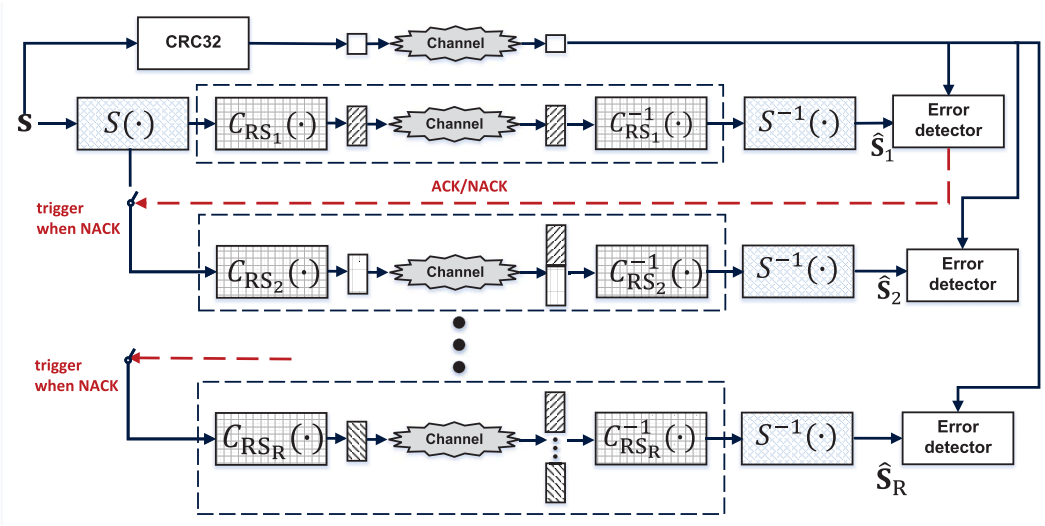


Fig. 1. The conventional RS-based IR-HARQ framework. If NACK is fed back, the next transmission is started.

happen and is a channel adaptive method. According to [33], IR-HARQ has an ergodic channel capacity for fading channels. Given the adaptive correction capability of IR-HARQ, it is commonly used in communication, especially in wireless communication. Here, we consider an RS code, which is a maximum distance separable (MDS) code. An n-symbol RS code with k symbols of information can correct $n-k$ erasure symbols or $(n-k)/2$ error symbols. The corresponding code rate can be calculated by $\frac{k}{n}$. RS codes can be easily used for IR-HARQ because a punctured MDS code is still an MDS code [34], [35]. For example, n total symbols are punctured to n' $(<n)$. The correction capability becomes $n'-k$ erasure symbols or $(n'-k)/2$ error symbols. Thus, if transmitting n' symbols is not enough to correct the error symbols, we can transmit $n-n'$ incremental symbols to increase the correction capability from $(n'-k)/2$ to $(n-k)/2$ error symbols.

The conventional HARQ transmission process for sentences is described in Fig. 1. The sentence $\mathbf{s}=[w_1,w_2,\cdots,w_{L_\mathbf{s}}]$ has $16L_\mathbf{s}$ bits with each integer w_i expressed by 2 bytes. Then, the 32-bit parity code of $\mathbf{s}$ is generated by cyclic redundancy check (CRC) error detection, called CRC32, and is sent at the very beginning. The full-length codeword $\mathbf{b}_\mathrm{R}$ is punctured to R different lengths $\mathbf{b}_1,\cdots,\mathbf{b}_\mathrm{R}$. In these codes, $\mathbf{b}_1$ has the shortest error correcting code while $\mathbf{b}_\mathrm{R}$ has the longest one. The $\mathbf{b}_1$ is sent firstly and the incremental symbols of error correcting code between $\mathbf{b}_\mathrm{i}$ and $\mathbf{b}_{\mathrm{i}-1}$ are sent to improve the correction capability successively. If the estimated sentence of the i-th transmission $\mathbf{s}_i$ at the receiver is determined to be correct by CRC32, the ACK signal will be sent to the transmitter, and the reserved incremental symbols will not be required. Otherwise, the next transmission will start according to NACK feedback. The sentence is unsuccessfully received when all incremental symbols are transmitted, but the full-length codeword still cannot be decoded correctly.

B. DL-Based Autoencoder

Joint source-channel coding has great potential to improve transmission efficiency. For the DL-based autoencoder, the conventional encoder and decoder, $C_\beta(S_\alpha(\cdot))$ and $S_\alpha^{-1}(C_\beta^{-1}(\cdot))$, are replaced by a DL-based encoder and decoder. To train the encoder-decoder architecture in an end-to-end manner, they are connected by a channel layer, which usually consists of a dropout layer and an additive white Gaussian noise (AWGN) layer if quantization is not considered. The channel layer can be a bit-erasure or bit-error layer for the quantized encoder and decoder. In addition, the channel layer can learn the fading channel through the generative adversarial network [10].

DL-based autoencoders perform better than conventional methods, especially under wired environments with nonlinear interference and limited transmission resources. Joint source-channel coding [27] initiates the words with Glove pre-trained embeddings [36] and uses RNN to learn the semantic information. In [28], the attention mechanism was used for the semantic coder, which was called Transformer in [29]. The first step of the semantic encoder for a sentence is to obtain the word embedding. As the input of a DL-based encoder, all sentences are zero-padded to a length of L. The word embedding process needs lookup table $\mathbf{\Psi}$ and $L_\mathbf{\Psi}$ words in the dictionary. Denote M as the length of the word vectors after the word embedding process. Thus, $\mathbf{\Psi}$ is an $L_\mathbf{\Psi}\times M$ real matrix whose parameters are trainable. The word embedding process is denoted as $f_{\mathrm{embed}}(\cdot;\cdot)$, and a sentence after wording embedding can be written as

$$\mathbf{V}=f_{\mathrm{embed}}(\mathbf{s};\mathbf{\Psi})=\begin{bmatrix}\mathbf{\Psi}[w_1]\\ \vdots \\ \mathbf{\Psi}[w_L]\end{bmatrix}+\mathbf{PE} \tag{3}$$

where $\mathbf{V}\in\mathbb{R}^{L\times M}$, $\mathbf{s}$ is the input sentence, $\mathbf{\Psi}[w_1]$ is the vector of the w_1-th row in the trainable $\mathbf{\Psi}$, and the additive matrix $\mathbf{PE}$ is a constant matrix for position encoding defined in [29]. The word vectors in $\mathbf{\Psi}$ contain the meaning of the words because the distance of any two similar-word vectors is usually shorter than that of dissimilar-word vectors. Pre-trained lookup tables, such as Word2Vec [37] and Glove [36], are available for extracting semantic information. The detailed architecture of

Transformer is provided in [29]. For convenience, we denote the Transformer-based encoder and decoder as $T_{\rm en}(\cdot)$ and $T_{\rm de}(\cdot)$, respectively. Here, the trainable parameters in these processes are not shown explicitly, and $f_{\rm embed}(\mathbf{s};\mathbf{\Psi})$ is simplified to $f_{\rm embed}(\mathbf{s})$. A fully-connected (FC) layer converts the output of Transformer in the decoder from $\mathbb{R}^{L\times M}$ to $\mathbb{R}^{L\times L_\Psi}$ via SoftMax activation. Thus, decoded sentence $\hat{\mathbf{s}}$ is obtained in accordance with the index of the maximum value on each row. The process is denoted as $f_{\rm argmax}(\cdot)$. For machine translation, translated sentence $\hat{\mathbf{s}}$ can be written as

$$\hat{\mathbf{s}} = f_{\rm argmax}(T_{\rm de}(T_{\rm en}(f_{\rm embed}(\mathbf{s})))). \tag{4}$$

For semantic communication, several FC layers [28] are used to compress $T_{\rm en}(f_{\rm embed}(\mathbf{s}))$ into transmit symbols, and the received symbols are decompressed by FC layers as

$$\hat{\mathbf{s}} = f_{\rm argmax}(T_{\rm de}(f_{\rm de}(h(f_{\rm en}(T_{\rm en}(f_{\rm embed}(\mathbf{s}))))))), \tag{5}$$

where $h(\cdot)$ is the channel layer, and $f_{\rm en}(\cdot)$ and $f_{\rm de}(\cdot)$ are the processes of FC layers. Then, this architecture is trained to cope with the effect of channels. This method is used as the basic framework of our study. For convenience, $f_{\rm en}(T_{\rm en}(f_{\rm embed}(\cdot)))$ and $f_{\rm argmax}(T_{\rm de}(f_{\rm de}(\cdot)))$ are denoted as $SC_{\rm en}(\cdot)$ and $SC_{\rm de}(\cdot)$, respectively.

We use three different channel models for training and testing. First, the binary symmetric channel has a certain BER, p, and the received bit, $\hat{b}_i$, can be expressed as

$$\hat{b}_i = h(b_i) = (1-p_i)\cdot b_i + p_i\cdot(1-b_i), \tag{6}$$

where b_i is the transmit bit in bit sequence $\mathbf{b}$ in Eq. (1) and p_i has a p probability of being 1; otherwise, p_i is 0. The $\mathbf{b}$ can also be modulated into a symbol $\mathbf{x}$. Then, the AWGN, $\mathbf{z}$, can be added to the transmit symbols, yielding

$$\hat{\mathbf{x}}_i = h(\mathbf{x}) = \mathbf{x} + \mathbf{z}, \tag{7}$$

where signal-to-noise ratio (SNR) is denoted as $\frac{||\mathbf{x}||^2}{||\mathbf{z}||^2}$ and this $h(\cdot)$ is called AWGN channel. Furthermore, the Rayleigh block fading channel is used with channel gain g, which obeys Rayleigh distribution. The channel gain is constant during a transmission block and changes when the next transmission starts. The SNR of this transmission block is $\frac{||g\cdot\mathbf{x}||^2}{||\mathbf{z}||^2}$ and the effect of channels is

$$\hat{\mathbf{x}}_i = h(\mathbf{x}) = g\cdot\mathbf{x} + \mathbf{z}. \tag{8}$$

Semantic networks demonstrate better performance than conventional ones, especially when new semantic metrics, such as bilingual evaluation understudy (BLEU) [38], are applied. However, the impact of semantic methods on the entire sentence transmission should be investigated further.

C. Challenges in Semantic Coding

A variable code length is beneficial for coding efficiency, whereas a fixed bit/symbol transmission [27], [28] performs better for short sentences than for long ones. Although a variable length method was studied in [31], the variable code length for Transformer-based coding is still lacking.

In addition, semantic networks lack combination with HARQ, which is important for successful transmission. The superiority of semantic networks is exhibited not only in lossless compression but also in lossy transmission. The sentences with wrong words estimated by semantic networks also contain useful semantic information under extremely hostile environments. The HARQ method should also be refined to adapt to this content-related encoder-decoder architecture.

III. HARQ-Based on a Semantic Coder

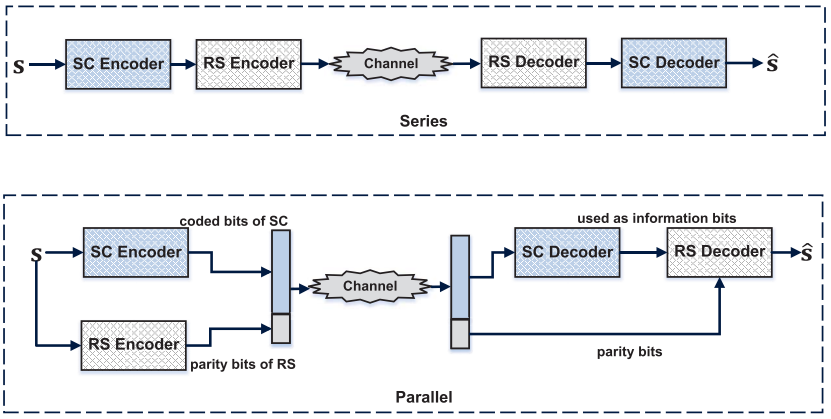
In this section, we propose different semantic architectures that are combined with IR-HARQ in different extents. First, the design of the semantic network is independent of the HARQ framework. Second, all the source-channel coding and the incremental encoded bits are generated by neural networks. Lastly, we replace CRC with a novel similarity detection method. The sentences with wrong words can be accepted by the receiver as long as their meanings are still similar to the original ones.

A. Semantic-Based Source Coding With RS-Based HARQ

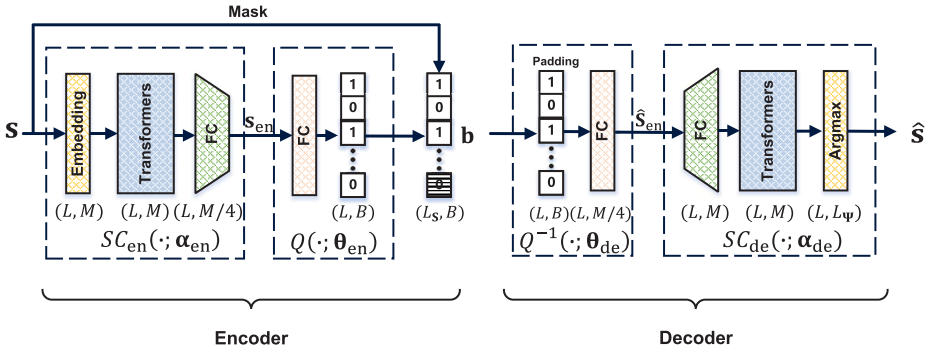
Joint semantic source and channel coding was studied in [27], [28]. Existing methods have shown their superiority under a low SNR and limited number of bits for each sentence. However, fixed architectures are not flexible and efficient because sentences usually have different lengths. Meanwhile, joint designs make the combination of channel coding and HARQ difficult because DL-based coding is inexplicable.

As shown in Fig. 2(a), we design SC separately without the combination of conventional RS coding and IR-HARQ, and IR-HARQ (not shown in the figure) is directly based on RS coding. The two proposed methods are called series and parallel SC-RS-HARQ. These methods exploit conventional RS coding in different stages. In series SC-RS-HARQ, RS coding is used to protect the coded bits of SC. That is, the received sentences are directly decoded by SC. Thus, this method may still commit mistakes under the absence of transmission errors because the performance of DL-based SC is not guaranteed when testing. In parallel SC-RS-HARQ, the sentence is encoded by SC and RS, and the parity bits of RS coding are transmitted directly. Therefore, RS coding is used to protect the original sentence and the correct sentence decoded by SC. Each word in the sentence is expressed by 2 bytes. The RS codeword of $\mathbf{s}$ is variable with zero padding. For example, the total bytes are padded to the minimum multiple of 5 when we apply RS(5,7). The received sentences are corrected by the RS decoder, and their performances are guaranteed under low BER. Overall, these methods can achieve good performance under high BER due to the introduction of SC. We choose CRC32 for error detection, which transmits a 32-bit CRC parity code of the transmitted sentence. The probability of CRC32 committing a mistake is very low and can be ignored.

In addition, we develop an SC to encode a sentence into different bit lengths in accordance with the length of the sentence (Fig. 2(b)), which is beneficial for code efficiency. The encoder and decoder are based on Transformer [29]. The SC encoder embeds $\mathbf{s}$ as in Eq. (4), and the embedded word vectors are processed by a Transformer-encoder with six blocks of self-attention layers. The second dimension size of



(a)



(b)

Fig. 2. (a) Two combination methods of SC- and RS- based IR-HARQ. (b) Architecture of Transformer-based SC. The encoded bits are masked the in accordance with input sentence length.

the Transformer-encoder is M. The output of Transformers is compressed by an FC layer, and the entire process is denoted as $SC_{\text{en}}(\cdot)$ with output

$$\mathbf{s}_{\text{en}} = SC_{\text{en}}(\mathbf{s}; \boldsymbol{\alpha}_{\text{en}}), \tag{9}$$

where $\mathbf{s}_{\text{en}} \in \mathbb{R}^{L\times M/4}$ and $\boldsymbol{\alpha}_{\text{en}}$ represent all the trainable parameters. We compress the second dimension into $M/4$ to reduce the complexity of the FC layers in quantization. However, a larger compression ratio, such as 8, may lead to information loss during compression and diminish the performance. The one-bit quantization module first converts $\mathbf{s}_{\text{en}}$ into $\mathbb{R}^{L\times B}$ with an FC layer, where B is the number of bits for each word. Different from existing methods, the quantization module also masks part of the bits in accordance with sentence length $L_{\mathbf{s}}$. We denote the quantization process as $Q(\cdot)$, and its output can be expressed as

$$\mathbf{b} = Q(\mathbf{s}_{\text{en}}; \boldsymbol{\theta}_{\text{en}}), \tag{10}$$

where $\mathbf{b}$ is an $L_{\mathbf{s}} \times B$ bit vector if the sentence length is $L_{\mathbf{s}}$. The other $(L - L_{\mathbf{s}}) \times B$ bits are not transmitted to save transmission resources because input sentence $\mathbf{s}$ only has L_s words and is zero-padded to L.

The dequantization module pads the input bits to an $L \times B$ binary matrix with zeros. Then, an FC layer is used for reshaping, and its output is $\hat{\mathbf{s}}_{\text{en}} \in \mathbb{R}^{L\times M/4}$, thereby yielding

$$\hat{\mathbf{s}}_{\text{en}} = Q^{-1}(\mathbf{b}; \boldsymbol{\theta}_{\text{de}}). \tag{11}$$

Afterward, $\hat{\mathbf{s}}_{\text{en}}$ is decompressed into $\mathbb{R}^{L\times M}$ with an FC layer and goes through a Transformer-decoder with six blocks of self and cross-attention layers. The second dimension of the Transformer-decoder is M. Then, the argmax layer uses an FC layer with SoftMax activation and outputs $L \times L_{\Psi}$ vectors of probabilities in the dictionary. The estimated sentence is composed of the maximum possible words. Similar to the encoder process, the estimated sentence can be written as

$$\hat{\mathbf{s}} = SC_{\text{de}}(\hat{\mathbf{s}}_{\text{en}}; \boldsymbol{\alpha}_{\text{de}}). \tag{12}$$

The number of bits for each word, B, is difficult to choose when the objective is to balance coding efficiency and bit consumption; moreover, an end-to-end training process consumes too much time. The entire training process is also divided into three steps.

1) The parameters in $SC_{\text{en}}(\cdot)$ and $SC_{\text{de}}(\cdot)$ are trained without quantization layers $Q(\cdot)$ and $Q^{-1}(\cdot)$, i.e., $\hat{\mathbf{s}}_{\text{en}} = \mathbf{s}_{\text{en}}$. This training process can be expressed as

$$(\hat{\boldsymbol{\alpha}}_{\text{en}}, \hat{\boldsymbol{\alpha}}_{\text{de}}) = \underset{\boldsymbol{\alpha}_{\text{en}}, \boldsymbol{\alpha}_{\text{de}}}{\arg\min}\, L_{\text{CE}}\left(\mathbf{s}, SC_{\text{de}}(SC_{\text{en}}(\mathbf{s}; \boldsymbol{\alpha}_{\text{en}}); \boldsymbol{\alpha}_{\text{de}})\right), \tag{13}$$

where $L_{\text{CE}}(\cdot)$ denotes the cross-entropy (CE) loss function.

2) As a hyper-parameter, B affects the WER of the trained SC. The choice of B should protect the information in $\mathbf{s}_{\text{en}} = SC_{\text{en}}(\mathbf{s}; \hat{\boldsymbol{\alpha}}_{\text{en}})$ when going through the quantization

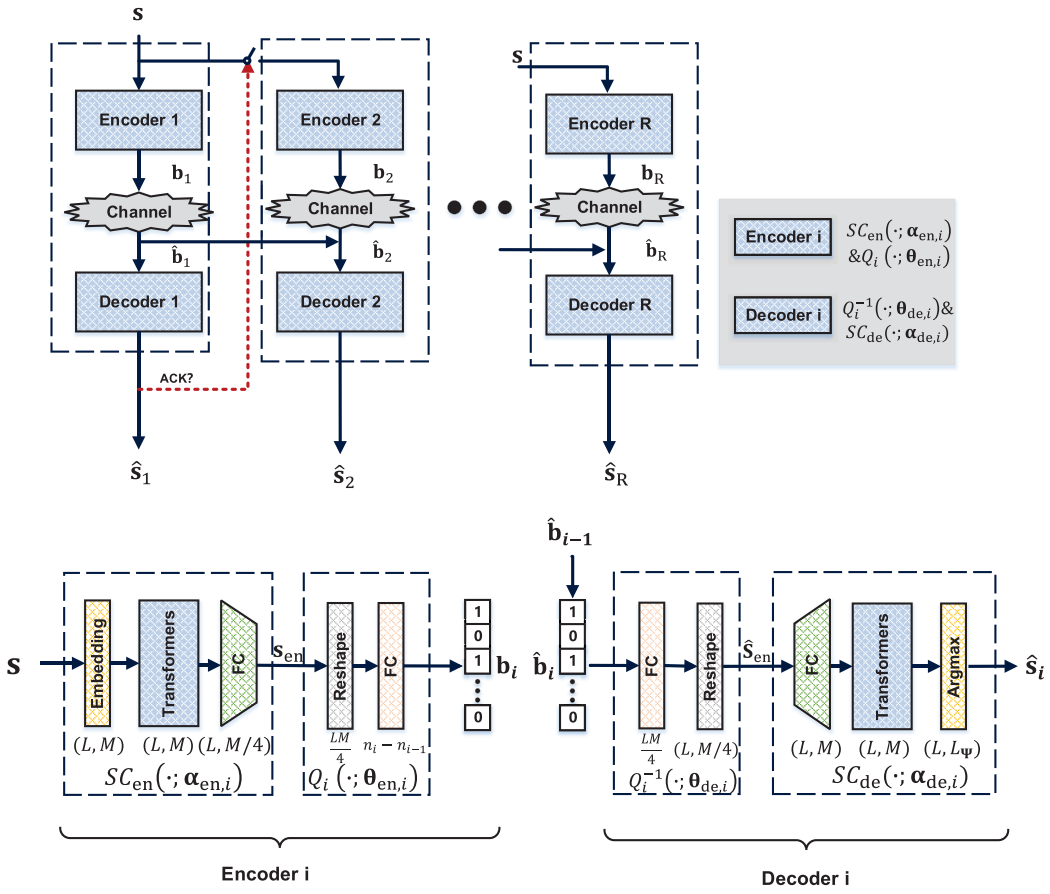


Fig. 3. Structure of SCHARQ, which can transmit R times at the maximum.

and dequantization layers. The derivative of the quantization layer is replaced with the derivative of the expectation in the backward pass [39]. The loss function is the mean-squared error (MSE), and the training process can be expressed as

$$(\hat{\boldsymbol{\theta}}_{\mathrm{en}}, \hat{\boldsymbol{\theta}}_{\mathrm{de}}) = \underset{\boldsymbol{\theta}_{\mathrm{en}}, \boldsymbol{\theta}_{\mathrm{de}}}{\arg\min}\, L_{\mathrm{MSE}}\left(\mathbf{s}_{\mathrm{en}}, Q^{-1}(Q(\mathbf{s}_{\mathrm{en}}; \boldsymbol{\theta}_{\mathrm{en}}); \boldsymbol{\theta}_{\mathrm{de}})\right). \tag{14}$$

3) All trainable parameters are fine-tuned in an end-to-end manner as follows:

$$(\hat{\boldsymbol{\alpha}}_{\mathrm{en}}, \hat{\boldsymbol{\alpha}}_{\mathrm{de}}, \hat{\boldsymbol{\theta}}_{\mathrm{en}}, \hat{\boldsymbol{\theta}}_{\mathrm{de}}) = \underset{\boldsymbol{\alpha}_{\mathrm{en}}, \boldsymbol{\alpha}_{\mathrm{de}}, \boldsymbol{\theta}_{\mathrm{en}}, \boldsymbol{\theta}_{\mathrm{de}}}{\arg\min}\, L_{\mathrm{CE}}\left(\mathbf{s}, \hat{\mathbf{s}}\right). \tag{15}$$

The two methods design SC and conventional IR-HARQ modules separately so that they can be applied easily. The advantages of SC are exploited under high BER. Meanwhile, the conventional method shows its superiority under lossless transmission, and this combination can address the drawback of the AI method. However, joint optimization is also a potential strategy for reducing transmission resources further. Thus, joint source-channel coding and the HARQ architecture are studied further.

B. Semantic-Based Joint Source-Channel Coding and HARQ

Long sentences can be dealt with by transmitting incremental bits because of the adjustable length of IR-HARQ. Thus, we propose an end-to-end semantic framework similar to the IR-HARQ framework, and it is called SCHARQ. SCHARQ transmits the incremental bits until the receiver estimates the sentence successfully or reaches the maximum number of retransmissions. These incremental bits can not only improve the correction capability but can also carry extra information for complex sentences. Overall, this framework is flexible under varying channels and different sentence lengths.

There are R SC-based encoders and decoders for R transmissions, as shown in Fig. 3. The encoder and decoder architectures differ from those in Fig. 2. The quantization process $Q_i(\cdot; \boldsymbol{\theta}_{\mathrm{en},i})$ in SCHARQ does not need to mask part of the bits, but it converts the output of $SC_{\mathrm{en}}(\cdot; \boldsymbol{\alpha}_{\mathrm{en},i})$ to $\mathbb{R}^{1\times(n_i - n_{i-1})}$ with a dense layer then uses one-bit quantization. An n_1-bit vector $\mathbf{b}_1$ is transmitted first, and the following transmissions use a similar architecture. The i-th transmission can be expressed as

$$\mathbf{b}_i = Q_i(SC_{\mathrm{en}}(\mathbf{s}; \boldsymbol{\alpha}_{\mathrm{en},i}); \boldsymbol{\theta}_{\mathrm{en},i}), \tag{16}$$

The previous transmitted bits are connected with the incremental bits $\hat{\mathbf{b}}_i$ and decoded together to yield

$$\hat{\mathbf{s}}_i = SC_{\mathrm{de}}(Q_i^{-1}([\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_i]; \boldsymbol{\theta}_{\mathrm{de},i}); \boldsymbol{\alpha}_{\mathrm{de},i}), \tag{17}$$

where $Q_i^{-1}(\cdot)$ reshapes these bits with a dense layer and $SC_{\mathrm{de}}(\cdot)$ has the same architecture as that in Fig. 2.

The lengths of $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R$ are $n_1, n_2 - n_1, \cdots, n_R - n_{R-1}$, which determines the output dimension of each encoder as hyper-parameters. These hyper-parameters are set in accordance with the goal of HARQ. For the first transmission, the transmitted bit length n_1 is used for good channel environments, and most sentences can be encoded into n_1 bits without any loss. Then, the bit lengths of the following transmissions are set to restore long sentences under poor channels. This choice is similar to selecting different code rates for each transmission in an IR-HARQ system.

The training process uses $\hat{\boldsymbol{\alpha}}_{\rm en}$ and $\hat{\boldsymbol{\alpha}}_{\rm de}$ trained in Section III A for the initiation of all $\boldsymbol{\alpha}_{{\rm en},i}$ and $\boldsymbol{\alpha}_{{\rm de},i}$. For the first transmission, the training process is similar to Steps 2 and 3 in Section III A, and no bits are in error because the first transmission is trained under good channel environments.

$$(\hat{\boldsymbol{\alpha}}_{{\rm en},1}, \hat{\boldsymbol{\alpha}}_{{\rm de},1}, \hat{\boldsymbol{\theta}}_{{\rm en},1}, \hat{\boldsymbol{\theta}}_{{\rm de},1}) = \underset{\boldsymbol{\alpha}_{{\rm en},1}, \boldsymbol{\alpha}_{{\rm de},1}, \boldsymbol{\theta}_{{\rm en},1}, \boldsymbol{\theta}_{{\rm de},1}}{\arg\min} L_{\rm CE}\left(\mathbf{s}, \hat{\mathbf{s}}_1\right), \tag{18}$$

where

$$\hat{\mathbf{s}}_1 = SC_{\rm de}(Q_1^{-1}(Q_1(SC_{\rm en}(\mathbf{s}; \boldsymbol{\alpha}_{{\rm en},1}); \boldsymbol{\theta}_{{\rm en},1}); \boldsymbol{\theta}_{{\rm de},1}); \boldsymbol{\alpha}_{{\rm de},1}). \tag{19}$$

For the i-th transmission ($i > 1$), the trainable parameters of the previous transmissions are fixed, yielding

$$\mathbf{b}_k = Q_k(SC_{\rm en}(\mathbf{s}; \hat{\boldsymbol{\alpha}}_{{\rm en},k}); \hat{\boldsymbol{\theta}}_{{\rm en},k}),\ k \le i-1, \tag{20}$$

and 5% of the bits are in error for all the received $\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_i$. Thus, the i-th encoder and decoder can learn to restore the sentences from noisy channels. The training process can be expressed as

$$(\hat{\boldsymbol{\alpha}}_{{\rm en},i}, \hat{\boldsymbol{\alpha}}_{{\rm de},i}, \hat{\boldsymbol{\theta}}_{{\rm en},i}, \hat{\boldsymbol{\theta}}_{{\rm de},i}) = \underset{\boldsymbol{\alpha}_{{\rm en},i}, \boldsymbol{\alpha}_{{\rm de},i}, \boldsymbol{\theta}_{{\rm en},i}, \boldsymbol{\theta}_{{\rm de},i}}{\arg\min} L_{\rm CE}\left(\mathbf{s}, \hat{\mathbf{s}}_i\right),\ i > 1, \tag{21}$$

where

$$\hat{\mathbf{s}}_i = SC_{\rm de}(Q_i^{-1}([\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_{i-1}, h(Q_i(SC_{\rm en}(\mathbf{s}; \boldsymbol{\alpha}_{{\rm en},i}); \boldsymbol{\theta}_{{\rm en},i}))]; \boldsymbol{\theta}_{{\rm de},i}); \boldsymbol{\alpha}_{{\rm de},i}), \tag{22}$$

and $h(\cdot)$ indicates that 5% of the bits are in error.

The application of this framework is similar to that of the conventional IR-HARQ shown in Alg. 1. After the i-th transmission, we check $\hat{\mathbf{s}}_i$ with CRC and transmit $\mathbf{b}_{i+1}$ if this transmission cannot obtain the correct sentence. If the i-th transmission passes the CRC error detection, then $\hat{\mathbf{s}} = \hat{\mathbf{s}}_i$, and the subsequent transmission is not required. Given that the whole HARQ framework is designed based on a semantic encoder-decoder, it cannot guarantee the absence of error sentences under the testing set. Thus, we attempt to protect the source directly with the conventional RS code, which can perfectly correct the sentences as long as the transmission errors do not surpass its correction capability. The RS parity code of $\mathbf{s}$, $\mathbf{s}_{\rm parity}$, is transmitted to the receiver, and the parity code is connected to $\hat{\mathbf{s}}_i$. The process is similar to the parallel framework in Fig. 2(a). Thus, errors in $\hat{\mathbf{s}}_i$ decoded by the i-th decoder in SCHARQ can be further corrected by the RS decoder with $\mathbf{s}_{\rm parity}$. This method is called SCHARQ-RS. The extra transmit resources are required for $\mathbf{s}_{\rm parity}$; thus, the tiny errors of SCHARQ are repaired. For example, SCHARQ may have 1% WER when BER $= 0$ because of the difference in the training and test datasets. SCHARQ-RS enhances SCHARQ and has 0 WER when BER $= 0$.

C. Semantic-Based Error Detection for HARQ

The abovementioned joint design with HARQ makes the network highly flexible under varying channels and sentences, but the system still aims to minimize word errors rather than the semantic errors in transmission. To fully exploit the potential of the semantic architecture, we attempt to change CRC to a similarity detection in this section.

Conventional transmission systems usually rely on CRC error detection to repeat requests automatically and receive the correct sentences and feedback ACK. However, traditional CRC error detection will regard a sentence as having an error if it contains error words but has the same meaning as the original one. Similar sentences are also useful for semantic transmission, especially in hostile environments.

Although several methods for the similarity measurement of sentences, such as Levenshtein distance and BLEU, have been proposed, they only calculate the change of words between two sentences and have no insight into the meaning of different words. Recently, BERT [40], a model that is pre-trained under billions of words and sentences, has achieved great success in extracting semantic information. This architecture has been applied to measure the similarity of sentences, such as in [28]. $BERT(\mathbf{s})$ converts the input sentences, $\mathbf{s}$, into real vectors, and cosine similarity is used to measure the similarity in their semantic information, which is defined as

$$Sim(\mathbf{s}, \hat{\mathbf{s}}) = \frac{BERT(\mathbf{s})BERT(\hat{\mathbf{s}})^T}{|BERT(\mathbf{s})||BERT(\hat{\mathbf{s}})|}. \tag{23}$$

However, the true sentence $\mathbf{s}$ in HARQ systems is unavailable at the receiver. Thus, a new method that is similar to CRC is proposed, and it is called Sim32 and shown in Fig. 4. Thirty-two extra bits are transmitted to the receiver for similarity detection. At the transmitter, the 32 bits are encoded as

$$\mathbf{b}_{\rm sim} = Q_{\rm sim}(SC_{\rm en}(\mathbf{s}; \boldsymbol{\alpha}_{{\rm sim},1}); \boldsymbol{\theta}_{{\rm sim},1}), \tag{24}$$

where $Q_{\rm sim}(\cdot; \boldsymbol{\theta}_{{\rm sim},1})$ converts the output of $SC_{\rm en}(\cdot; \boldsymbol{\alpha}_{{\rm sim},1})$ into 32 bits, and $\boldsymbol{\alpha}_{{\rm sim},1}$ and $\boldsymbol{\theta}_{{\rm sim},1}$ are the trainable parameters in these processes. At the receiver, similarity detection is based on two inputs, namely, $\hat{\mathbf{b}}_{\rm sim}$ and $\hat{\mathbf{s}}$, which can be expressed as

$$Sim32(\hat{\mathbf{s}}, \hat{\mathbf{b}}_{\rm sim}) = f_{\rm sim}(\hat{\mathbf{b}}_{\rm sim}, SC_{\rm en}(\hat{\mathbf{s}}; \boldsymbol{\alpha}_{{\rm sim},2}); \mathbf{W}_{\rm sim}), \tag{25}$$

where $f_{\rm sim}$ has two FC layers and the trainable parameters, $\boldsymbol{\alpha}_{{\rm sim},2}$ and $\mathbf{W}_{\rm sim}$, are introduced. Its hidden layer has four times the size of the input and ReLU activation function, and it only outputs one value by using the sigmoid activation function.

The training process of this architecture collects the estimated sentences from the aforementioned frameworks under different channel conditions and retransmission stages. The label is based on Eq. (19), and it satisfies

$$label(\mathbf{s}, \hat{\mathbf{s}}) = \begin{cases} 1, & Sim(\mathbf{s}, \hat{\mathbf{s}}) > 0.98, \\ 0, & Sim(\mathbf{s}, \hat{\mathbf{s}}) \le 0.98, \end{cases} \tag{26}$$

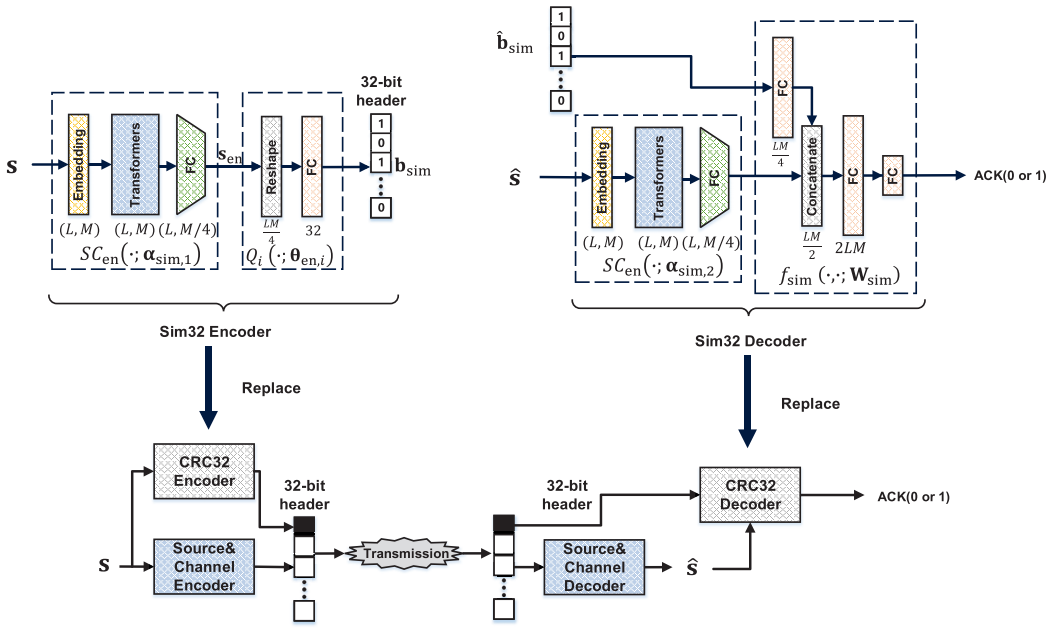


Fig. 4. Structure of the similarity detection method. The estimated sentence $\hat{\mathbf{s}}$ is received by the proposed methods, namely, SC-RS-HARQ and SCHARQ.

where $Sim(\mathbf{s}, \hat{\mathbf{s}}) > 0.98$ indicates that the estimated sentences are similar enough to express the semantic information and their labels are 1. The training process can be written as

$$(\hat{\boldsymbol{\alpha}}_{\text{sim},1}, \hat{\boldsymbol{\alpha}}_{\text{sim},2}, \hat{\boldsymbol{\theta}}_{\text{sim},1}, \hat{\mathbf{W}}_{\text{sim}}) = \underset{\boldsymbol{\alpha}_{\text{sim},1}, \boldsymbol{\alpha}_{\text{sim},2}, \boldsymbol{\theta}_{\text{sim},1}, \mathbf{W}_{\text{sim}}}{\arg\min} L_{\text{CE}} \left(label(\mathbf{s}, \hat{\mathbf{s}}), Sim32(\hat{\mathbf{s}}, \hat{\mathbf{b}}_{sim}) \right). \tag{27}$$

After training, 32-bit CRC can be replaced with Sim32. Similar to CRC, Sim32 needs to judge the similarity from the estimated sentence with only 32 encoded bits from the true sentence. The transmission is considered successful when $Sim32(\hat{\mathbf{s}}, \hat{\mathbf{b}}_{\text{sim}}) > 0.5$.

IV. Numerical Results

In this section, we present the numerical results of different frameworks and discuss the pros and cons of the semantic-based HARQ. We also compare their bit consumption with those of competing frameworks.

A. Configurations of the Simulation System

The English version of the proceedings of the European Parliament [41] is selected as the dataset, which has 2.2 million sentences and 53 million words. We set up the dictionary by using the 30,000 most common words. These words can be denoted by 2-byte integers. The length of the sentences is restricted between 4 and 30. After this pre-processing, the sentences are split into a training set of 500,000 sentences and a test set of 50,000 sentences. The input sentences $\mathbf{s}$ are padded to their maximum length $L = 30$ with zeros, and the number of units in the hidden layers M is 128. The detailed settings of the proposed networks are shown in TABLE I.

TABLE I
SETTINGS OF THE PROPOSED NETWORKS

Networks	Modules	Layer	Output dimensions	Activation function
SC	INPUT	$\mathbf{s}$	30	/
	SC_{en}	Embedding	(30,128)	None
		6 × Transformer	(30,128)	None
		FC	(30,32)	ReLU
	Q	FC	(30,30)	ReLU
		1-bit quantization	(30,30)	/
		Mask	(L_s,30)	/
	Q^{-1}	Padding	(30,30)	
		FC	(30,32)	ReLU
	SC_{de}	FC	(30,128)	ReLU
		6 × Transformer	(30,128)	None
		FC	(30,30000)	SoftMax
	OUTPUT	$\hat{\mathbf{s}}$	30	Argmax
SCHARQ	INPUT	$\mathbf{s}$	30	/
	Encoder i	SC_{en}	(30,32)	
		Reshape	960	/
		FC	b_i	ReLU
		1-bit quantization	b_i	/
	bit error channel			
	Decoder i	FC	960	ReLU
		Reshape	(30,32)	ReLU
		SC_{de}	(30,30000)	SoftMax
	OUTPUT	$\hat{\mathbf{s}}_i$	30	Argmax

* For convenience, SC_{en} is directly used in the layer column to represent the same layers shown in the SC_{en} module of the SC. Sim32 only consists of the proposed modules and some FC layers; thus, its detailed architecture is omitted here.

Huffman coding is used as the conventional source coding and the training and testing sets are constructed on a word-level together. The average length of the coded bits is compared with the SC in Table II. SC is still not efficient compared with the Huffman code due to the different lengths of sentences. Meanwhile, SC has no mechanism to guarantee the absence of errors, and it costs numerous bits to reduce WER further when WER is already very small. For example,

TABLE II
SC SOURCE CODING WITH DIFFERENT B

	B	Bits/Sentence	WER
SC	40	654	0.01%
	30	490	1.03%
Fixed SC	/	500	9.81%
Huffman	/	397	/

SC with $B = 40$ needs 1/3 more bits than that with $B = 30$, but WER only decreases by nearly 1%. In the following simulations, $B = 30$ is selected to balance the bit consumption and WER. The semantic encoder with a fixed bit length, which is called fixed SC, encodes the sentence into 500 bits. The WER of the fixed SC is much higher than that of the SC with an average of 490 bits per sentence because it is weak in dealing with long sentences. As is the fixed length method based on Transformer, [30] has the similar performance as the fixed SC and is not efficiency under the varying sentence length.

The transmission using Huffman source coding, RS channel coding, and HARQ is called Huffman-RS-HARQ, where the number of SC coders is $R = 4$. The alphabet size of the RS code is a byte, which is widely used in storage and transmission. The hard-decision RS decoder in [42] is used here. The Huffman coded bits are converted into bytes. Then, the byte vector is padded to a multiple of 5 and encoded into the lowest code rate $\frac{5}{19}$, and the total length is n_R. The parity bytes are punctured. In the first transmission, $\frac{5}{19}$ information bytes and $\frac{2}{19}$ parity bytes are transmitted. Thus, the code rate of the first transmission is $\frac{k}{n_1} = \frac{5/19n_R}{5/19n_R+2/19n_R} = \frac{5}{7}$. When errors are detected at the receiver, the incremental $\frac{4}{19}$ parity bytes are transmitted until no parity bytes are reserved. Thus, the code rates of these transmissions are $\frac{k}{n_2} = \frac{5}{5+2+4} = \frac{5}{11}$, $\frac{k}{n_3} = \frac{5}{5+2+4+4} = \frac{5}{15}$, and $\frac{k}{n_4} = \frac{5}{5+2+4+4+4} = \frac{5}{19}$. The bit length of n_R is $397 \times \frac{19}{5} = 1508$.

Series and parallel SC-RS-HARQ methods only transmit three times at the most to require slightly fewer bits than the conventional one. For series SC-RS-HARQ, the process is the same as Huffman-RS-HARQ; thus, the bit length of n_R is $490 \times \frac{15}{5} = 1470$. For parallel SC-RS-HARQ, the source is protected directly, which means an integer vector is converted as a byte vector. Its RS code length is always equal to that of the series SC-RS-HARQ at each transmission. For SCHARQ, $R = 3$, and the code lengths are set as $n_1 = 300$, $n_2 = 500$, and $n_3 = 1000$. The settings are selected based on our tests. Nearly 90% of the sentences can be restored with 300-bit transmission when BER $= 0$. Nearly 99% of the sentences can be restored perfectly with 500-bit transmission when BER $= 0$. The last transmission is set to restore all sentences under noisy channels. In fact, nearly 99% of the sentences can be restored perfectly with 1000-bit transmission when BER $= 0.05$. When we train the networks, the channels are simulated in the same condition for the incremental transmissions and the number of bits is pre-determined. For example, if the channel condition of the first incremental transmission (n_2) is bad, the result of the incremental transmission $\hat{\mathbf{s}}_2$ may be worse than the previous one $\hat{\mathbf{s}}_1$. In this situation, the incremental transmission may be ineffective and a thorough retransmission from n_1 to n_3 is needed.

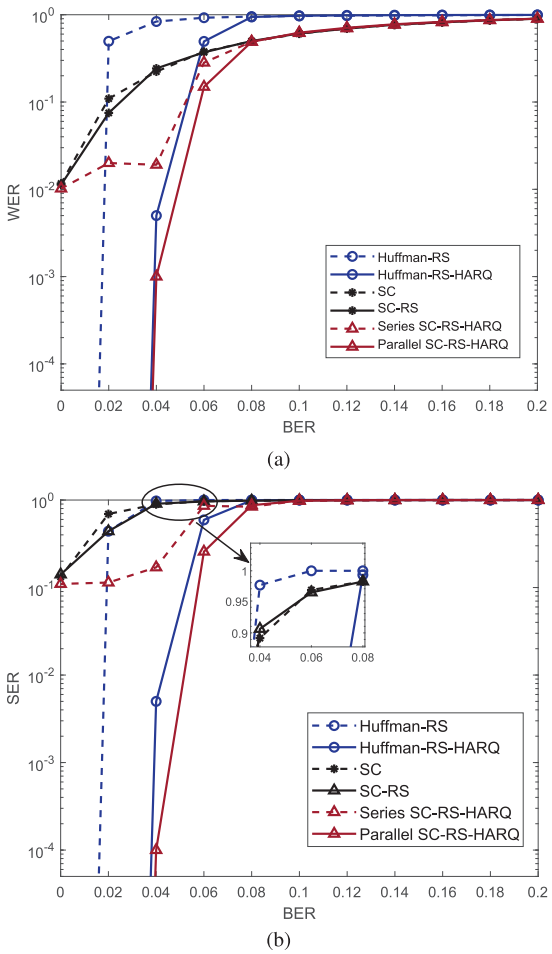


Fig. 5. (a) WER performance of the competing semantic and conventional methods. (b) SER performance of the competing semantic and conventional methods.

B. Performance of SC Combined With Conventional Methods

In Fig. 5(a), Huffman-RS represents the first transmission of Huffman-RS-HARQ with a code rate of $\frac{5}{7}$ and code length of approximately 555 bits. Among the compared methods, the Huffman-RS has the worst capability to cope with the change in BER. SC requires fewer bits (approximately 490 bits) but has better performance than the Huffman-RS, especially when BER is high. This phenomenon demonstrates that SC can handle bit errors even if it does not learn to do it. The series and parallel SC-RS-HARQ both perform better than Huffman-RS-HARQ when BER > 0.04. However, Huffman-RS-HARQ and parallel SC-RS-HARQ can guarantee an almost zero WER when BER < 0.04 because the estimated sentences are directly decoded by the conventional method, whereas series SC-RS-HARQ always has tiny errors when testing. Owing to the SC source coding, the series and parallel SC-RS-HARQ are better than Huffman-RS-HARQ when BER ≥ 0.06, and their performance is the same as that of SC when BER ≥ 0.08.

Therefore, semantic-based SC is still effective when the conventional method does not work.

As shown in Fig. 5(b), SC has a higher SER than Huffman-RS when BER $<$ 0.03 because nearly 1% of the error words are randomly spread out in the estimated sentences and cause about 10% of the estimated sentences to have one or two error words even when the conventional Huffman-RS-HARQ has no errors. Series SC-RS-HARQ also exhibits the same phenomenon; its SER can only reach 0.1, and its WER is around 0.01. By contrast, the conventional methods can estimate sentences perfectly if the number of wrong bits is below their error correction capability. Parallel SC-RS-HARQ can correct the error words after the SC decoder; thus, it can achieve perfect transmission when BER$<$ 0.04. Meanwhile, the SER of parallel SC-RS-HARQ always decreases earlier with BER than that of the competing methods, which shows its superiority under high-BER conditions.

The SER performance of the SC-based methods in Fig. 5(b) is worse than the WER performance in Fig. 5(a), except for parallel SC-RS-HARQ at BER $=$ 0.04. The semantic coding cannot guarantee 100% word accuracy in a sentence because it repairs the sentence in accordance with the semantic correlation. Meanwhile, 100% word accuracy is also unreachable even when BER $=$ 0 because the networks trained with the training dataset always exhibit slightly diminished performance under the test dataset. For the SC-based methods, the incorrect sentences with one or two wrong words may be considered to have no impact on the meaning of the sentences by SC. In the test, we observe 100 20-word sentences, and 1% of the words (20 words) are wrong when BER $=$ 0. Nearly 10% of the sentences are incorrect because each wrong sentence has about two wrong words when BER $=$ 0. This phenomenon is different from conventional RS coding. By contrast, the RS code can repair the sentences perfectly when the number of errors does not surpass its correction capability. Parallel SC-RS-HARQ uses the RS decoder to correct the few wrong words after the SC decoder and performs better than the other SC-based methods.

We also compare the BLEU performance of these methods in Fig. 6. Similar to their SER performance, parallel SC-RS-HARQ always has the best BLEU performance, whereas series SC-RS-HARQ is worse than conventional Huffman-RS-HARQ when BER $\leq$ 0.06. However, the performance gap between series SC-RS-HARQ and Huffman-RS-HARQ is smaller than that in Fig. 5(b) because the wrong sentences still reserve some correct semantic information. The three SC-based methods exhibit the same performance when BER $>$ 0.1 because the SC network plays a major role in high BER. Overall, series SC-RS-HARQ performs acceptably under the BLEU measurement because the reserved tiny error words when BER $<$ 0.03 have little impact on its semantic information.

The simulation above shows that semantic methods demonstrate better WER performance than the conventional methods when they are combined with conventional RS coding and HARQ properly. The SER performance of parallel SC-RS-HARQ is guaranteed to surpass that of conventional Huffman-RS-HARQ, whereas the SER performance of series SC-RS-HARQ is poor. Overall, the proposed methods can be easily applied in conventional HARQ systems, and the performance is improved under high-BER conditions.

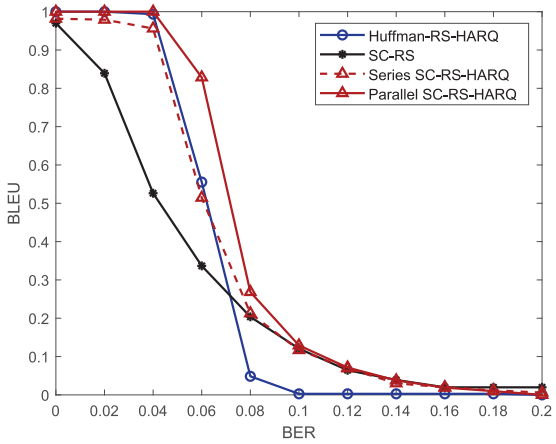


Fig. 6. BLEU performance of the proposed SC-RS-HARQ methods and the conventional Huffman-RS-HARQ method.

C. Performance of SCHARQ

In Fig. 7(a), the SCHARQ methods show their superiority when BER is between 0.04 and 0.2, and increasing the transmit bit limit helps improve the WER of SCHARQ. Considering that avoiding a tiny error is difficult when BER $\leq$ 0.04, we also transmit extra parity bits coded by the RS encoder to correct the estimated sentences similar to parallel SC-RS-HARQ. The SCHARQ with the RS code is called SCHARQ-RS, which has $\frac{5}{7}$ code rate for RS coding. Similarly, the SCHARQ methods always exhibit good capability to cope with high BER, as shown in Fig. 7(b). $n_R = 1000$ is adequate for SCHARQ to approach an almost-zero WER when BER $\leq$ 0.04, but its SER can only reach 0.05 at the most due to the tiny WER. The conventional RS code helps improve the performance and guarantees perfect transmission when the error in SCHARQ does not surpass the capability of the RS code. Thus, SCHARQ-RS can reach 0 WER and 0 SER when BER $=$ 0 and surpass SCHARQ when BER $\leq$ 0.1. However, the error that appears in the redundancy bits of the RS code may misapprehend the correct estimated sentences, making SCHARQ-RS a little worse than SCHARQ when BER is between 0.1 and 0.2. Parallel SC-RS-HARQ is the best method to use when BER $\leq$ 0.04, where SER is guaranteed to be zero; however, it performs worse than the SCHARQ methods when BER $>$ 0.06. The phenomena observed under the AWGN channels in Figs. 7(c) and (d) are similar. The RS-based HARQ methods always exhibit better performance than the others under high SNR, which reflects the superiority of the conventional error correction methods. SCHARQ-RS with $\frac{5}{7}$ RS code rate cannot surpass parallel SC-RS-HARQ when SNR is high. Decreasing the RS code rate of SCHARQ-RS is a choice to achieve better performance than the other competing methods. However, a low RS code rate for SCHARQ-RS may not be necessary because the few wrong words under high SNR have no impact on the semantic meaning.

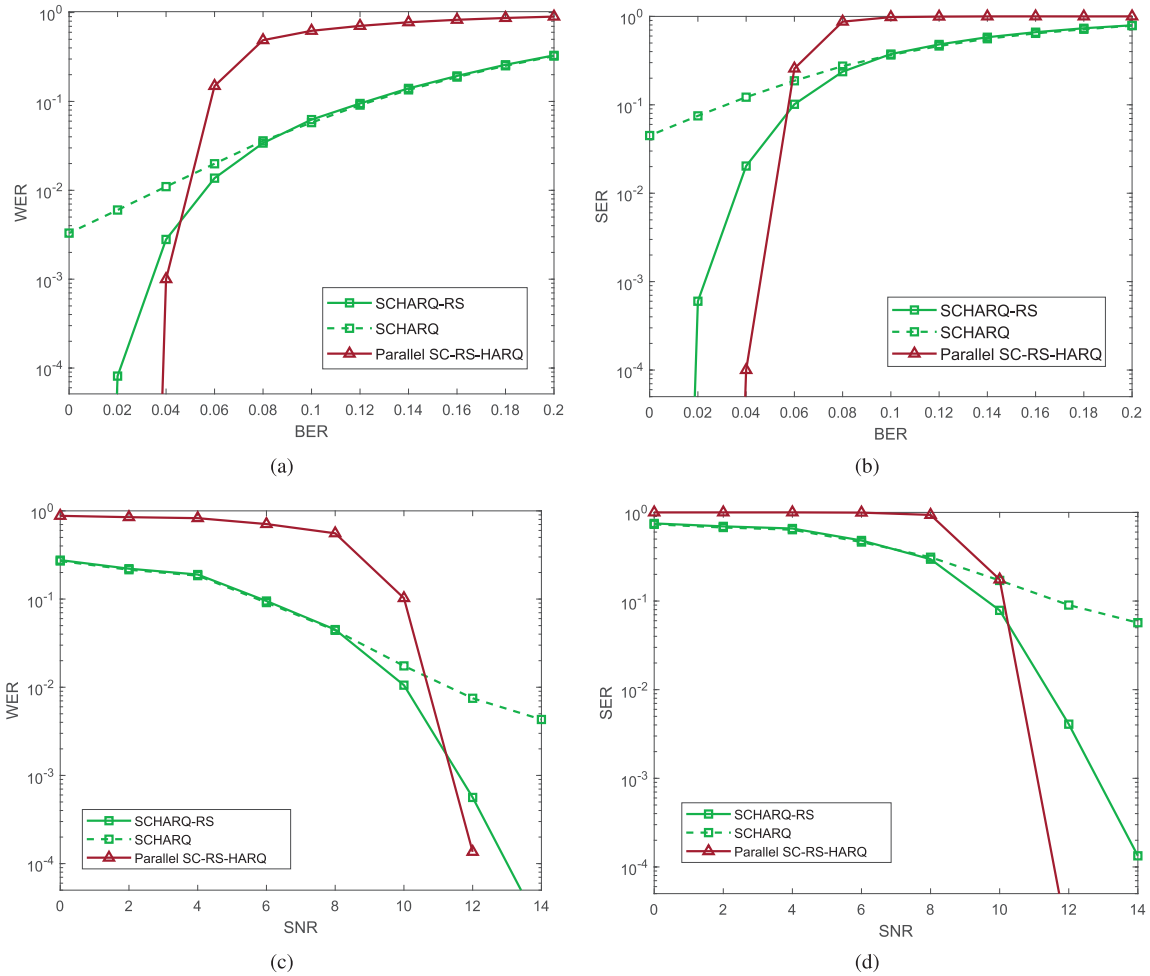


Fig. 7. Performances of the proposed SCHARQ and that of the competing methods. (a) WER under bit symmetric channels. (b) SER under bit symmetric channels. (c) WER under AWGN channels with 16-QAM modulation. (d) SER under AWGN channels with 16-QAM modulation.

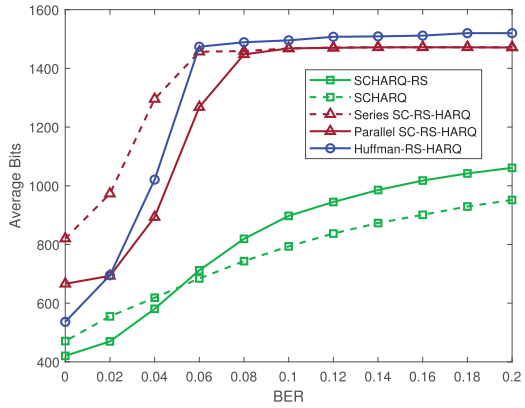


Fig. 8. Average bit consumption per sentence of different methods.

In Fig. 8, the average required number of bits for a sentence transmission are compared. Conventional Huffman-RS-HARQ and series SC-RS-HARQ reach the upper limit of transmission times at BER $=$ 0.06, and their SER is close to 1 when BER $\geq$ 0.08. By contrast, parallel SC-RS-HARQ performs well when BER is low and reaches the bit consumption limit at BER $=$ 0.08. Series SC-RS-HARQ requires more bits than the parallel one when BER $\leq$ 0.06 because the series one cannot reach zero SER, and some error sentences need to be retransmitted. The SCHARQ methods demonstrate the superiority of the joint design of semantic source-channel coding and HARQ in reducing transmit bit consumption, especially when BER is high. In particular, SCHARQ-RS needs fewer bits than SCHARQ when BER $\leq$ 0.05 even though it transmits extra redundancy bits. This phenomenon is due to the reduced times of retransmission with the help of the RS code when BER is low. However, the redundancy bits of the RS code cannot provide benefits for high BER but introduces additional errors (Fig. 7(b)); thus, SCHARQ-RS needs more bits than SCHARQ when BER $=$ 0.05-0.2.

The different value settings n_i of the i-th transmission are compared in Fig. 9. The robustness under fading channels is also shown under 16-QAM modulation. Specially, the channel for the i-th transmission is generated independently and the SNRs in different retransmissions may be different. Compared with the SCHARQ with three transmissions of $n_1 = 300, n_2 = 500, n_3 = 1000$, the SCHARQ with $n_1 = 300, n_2 = 1000$ requires more bits when SNR is low. Several transmissions of the SCHARQ with $n_1 = 300, n_2 = 1000$ require 1000 bits directly without the choice of the 500-bit

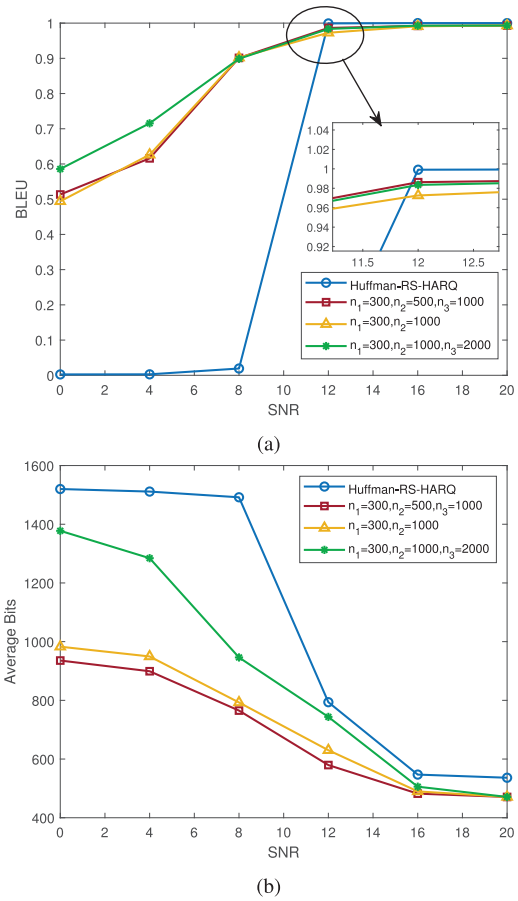


Fig. 9. Different settings of SCHARQ for each transmission. The methods are tested under Rayleigh fading channels with 16-QAM modulation. (a) BLEU performance. (b) Average bit consumption per sentence.

transmission. However, their BLEU performance is similar because the total transmit bits of the first and last transmissions are set to be the same, which means the correction capabilities are the same. The SCHARQ with $n_1 = 300, n_2 = 1000, n_3 = 2000$ has a better correction capability and requires more bits. In general, n_1 and n_R affect the curves of performance and bit consumption under extremely low and high SNR. The other ns only affect the shapes of these curves. The superiority of SCHARQ is still obvious under the different settings.

From this discussion, we find that the joint design of source-channel coding and HARQ has a significant advantage in reducing bit consumption and improving SER performance under high BER. The proposed methods exhibit their superiority at different BER scales. However, these designs cannot make full use of the semantic coder because the retransmission decision still relies on CRC that needs the estimated sentences to be error-free. Although the semantic method cannot outperform the conventional methods if BER is low and no error transmission is needed, it brings the possibility of protecting the sentence meaning when several words are incorrect. In the following part, CRC is replaced with similarity detection to study the pros and cons of semantic transmission.

D. Pros and Cons of Similarity Detection

We show the decision error rate of Sim32 in Fig. 10. We calculate the similarity in accordance with Eq. (23) with

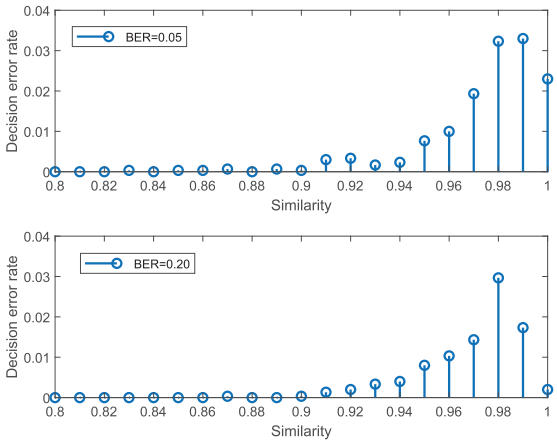
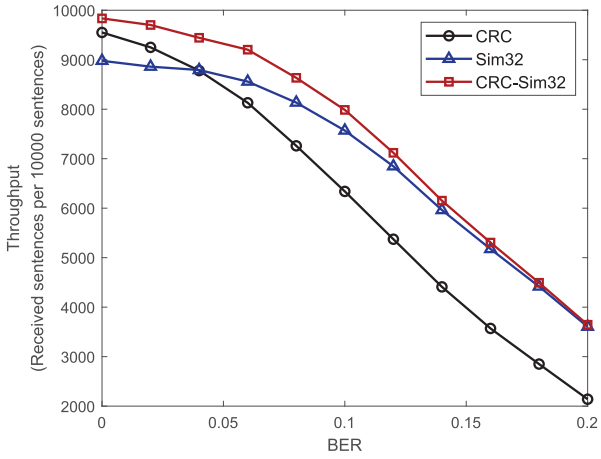


Fig. 10. Decision error rate of the Sim32 error detection method.

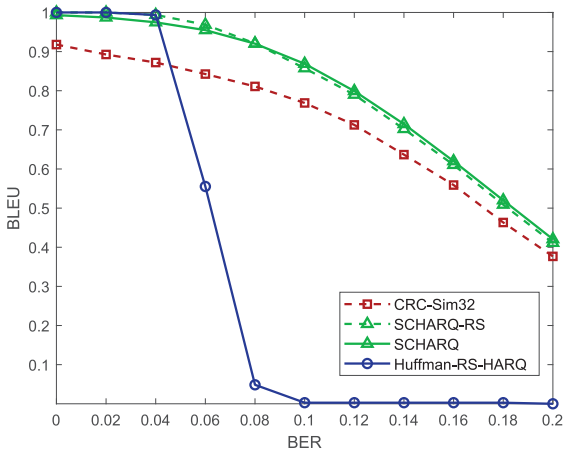
50,000 estimated sentences under two different BER settings. The decision error means that the estimated sentences with a similarity higher than 0.98 have a Sim32 decision of 0, and those with a similarity lower than 0.98 have a Sim32 decision of 1. As shown in Fig. 10, the error rate is high in the adjacent area of 0.98, demonstrating that Sim32 cannot obtain an accurate similarity because the semantic information of the true sentence is compressed into 32 bits in the transmission. In general, this similarity detection efficiently refuses the estimated sentences with a similarity less than 0.9 and is robust to the change in BER. However, approximately 2% of the correct sentences are mistaken as dissimilar sentences by Sim32 under BER $= 0.05$. To solve this issue, we only use Sim32 to find similar sentences after CRC detection to ensure that the correct sentences are directly received in the CRC process; this approach is called CRC-Sim32.

As shown in Fig. 11(a), the first two transmissions of SCHARQ still use CRC error detection, and the last transmission is changed into Sim32 and CRC-Sim32. Thus, the change in error detection has no influence on reducing the retransmission times. CRC-Sim32 uses CRC first then uses Sim32 to decide the sentences that fail in CRC. CRC-Sim32 needs 32 extra bits but can ensure that the correct sentences are not considered dissimilar sentences. The detected error sentences indicate that these estimated sentences cannot pass the decision after the last transmission, and this error rate is shown in Fig. 11. The performance of CRC can represent the SER performance of SCHARQ in Fig. 7(b). The Sim32 detection in the last transmission allows several estimated sentences with error words to be received as similar sentences. This method increases the number of received sentences, especially when BER is high. However, several correct sentences are mistakenly decided as dissimilar sentences. Thus, Sim32 receives fewer sentences than CRC when BER ≤ 0.04. CRC-Sim32 performs the best when BER is low, and it approaches Sim32 when BER $= 0.2$ because only a few correct sentences are mistakenly rejected by Sim32 when BER $= 0.2$.

Replacing CRC with CRC-Sim32 in all transmissions can reduce the bit consumption further by reducing the times of retransmission. For example, the received sentences can be increased by 18%, and the average bit consumption can be



(a)



(b)

Fig. 11. (a) Sentences from 10000 transmit sentences received by the different error detection methods. Only the error detection method in the last transmission is replaced. (b) BLEU performance of the proposed joint source-channel coding and HARQ methods. CRC-Sim32 refers to the BLEU performance of SCHARQ when the error detection methods of all transmissions are replaced with CRC-Sim32.

reduced by nearly 40 bits when CRC is replaced with CRC-Sim32 in all transmission at BER = 0.2.

The BLEU performance of the SCHARQ-based methods is compared in Fig. 11(b). SCHARQ-RS, SCHARQ, and Huffman-RS-HARQ use CRC error detection, whereas CRC-Sim32 is the SCHARQ that uses CRC-Sim32 for all the transmission times. Although SCHARQ-RS has an obvious SER performance gap with SCHARQ in Fig. 7(b), they have similar BLEU performance because the few error words that can be corrected by the RS code have little influence on the semantic information. Thus, CRC-Sim32 only needs the received sentences to be understood rather than a high BLEU score; thus, several sentences need fewer bits than the SCHARQ with CRC error detection. The performance gap between SCHARQ with CRC and CRC-Sim32 under low BER is larger than that under high BER because many received sentences have a chance to reach a high BLEU score under low BER.

The proposed similarity detection aims to adopt an estimated sentence with error words as long as the semantic meaning is unchanged. However, similarity detection is a difficult task because the true sentences are unknown to the receiver. Extensive work must still be done before similarity detection becomes reliable in sentence transmission.

E. Similar Sentences Received by Similarity Detection

Here, we analyze several common mistakes that appear in the similar sentences decided by the proposed methods. As shown in Table III, several similar sentences of SCHARQ are collected under four settings. CRC in brackets means that the first two transmissions are decided by CRC, and the last transmission is determined by CRC-Sim32. CRC-Sim32 means that all the transmissions are determined by CRC-Sim32. TX is the transmitted sentence, and RX is the received sentence after SCHARQ. Three pairs of TX and RX sentences are shown under different settings.

Most of the similar sentences under BER = 0 (CRC) contain only one or two error words. These mistakes only happen in long sentences because the 1000-bit limitation of the semantic network may not be enough for these sentences. The three received sentences demonstrate that the wrong nouns are difficult to judge for similarity detection because replacing a noun usually has no influence on the grammar. In Sentence 1, “acquis” is replaced with “aviation”, which may be associated with “travel freely without barrier”. In Sentence 2, the noun “instrument” becomes the pronoun “that”, and its meaning is vague. As for Sentence 3, its meaning has no change but its grammar contains a mistake. Pressure “reconciliation” can also be considered pressure “decrease”.

When BER = 0.2 (CRC), many wrong words appear in similar sentences, and the similarity detection produces mistakes. For example, Sentence 2 is decided as a similar sentence, but its meaning is changed thoroughly by the error words. Sentence 1 also has a wrong noun and its meaning is changed to some extent. Meanwhile, Sentence 3 can be understood.

The use of CRC-Sim32 as a retransmission decision saves bit consumption further but introduces more similar sentences with more wrong words compared with CRC. When BER = 0, Sentences 1 and 2 encounter a change in words, and these words may lead to a misunderstanding. For example, the year “2011” is replaced with “2008” in Sentence 1, and this mistake directly affects the behavior of the receiver. This phenomenon also exists when BER = 0.2. Thus, similarity detection is still un-reliable. Although similar sentences with error words replaced with synonyms are received, several sentences with changed semantic information are difficult to identify and reject.

In general, the proposed Sim32 is a good attempt to exploit the capability of semantic architectures, which can protect the semantic information when encountering mistakes and attempting to repair the sentences according to the semantic relation. However, several similar sentences decided by Sim32 may still lead to misunderstanding

F. Complexity Analysis

The majority of the observed complexity is from the Transformer-based architectures. We use more Transformer-based architectures than [30] did; thus, our methods have

TABLE III
THE SENTENCES PASS THE SIM32 BUT CONTAIN ERROR WORDS

	BER=0 (CRC)
1	RX: this **aviation** means a lot to eastern and central european countries also for historical reasons because it provides the opportunity to travel freely without barriers TX: this **acquis** means a lot to eastern and central european countries also for historical reasons because it provides the opportunity to travel freely without barriers
2	RX: today we must welcome an important result which is that we consider the citizens of bulgaria and romania to be among those who can benefit from this fundamental **that** TX: today we must welcome an important result which is that we consider the citizens of bulgaria and romania to be among those who can benefit from this fundamental **instrument**
3	RX: the problem must be treated at its roots by providing better employment and education opportunities for them this internal migration pressure will also **reconciliation** within the european union TX: the problem must be treated at its roots by providing better employment and education opportunities for them this internal migration pressure will also **decrease** within the european union
	BER=0.2 (CRC)
1	RX: the **disputes** just do not back this idea up either TX: the **facts** just do not back this idea up either
2	RX: romania has invested more vote eur 1 billion and the results have undertaken countries positive in all **this communities are** TX: romania has invested more than eur 1 billion and the results have definitely been positive in all **the evaluation reports**
3	RX: no one **are** interested **freely** in a two track europe TX: no one **is** interested **nowadays** in a two track Europe
	BER=0 (CRC-Sim32)
1	RX: it was already established in 2007 that once the technical **negotiations** were fulfilled bulgaria and romania would join the schengen area in **2008** TX: it was already established in 2007 that once the technical **criteria** were fulfilled bulgaria and romania would join the schengen area in **2011**
2	RX: fellow members bulgaria and romania have completed their **development** and we are building on **greater** security systems in cooperation with **our** schengen partners TX: fellow members bulgaria and romania have completed their **job** and they are building on **these** security systems in cooperation with **their** schengen partners
3	RX: what is the situation when it comes to the judicial reform and anti corruption measures that are still **achieved** TX: what is the situation when it comes to the judicial reform and anti corruption measures that are still **needed**
	BER=0.2 (CRC-Sim32)
1	RX: bulgaria and romania are absolutely not **preparations** for schengen TX: bulgaria and romania are absolutely not **ready** for Schengen
2	RX: we **are on** double standards TX: we **cannot allow** double standards
3	RX: what **president** is the commission giving that this **rapporteur** will be **resolved** effectively TX: what **guarantees** is the commission giving that this **problem** will be **tackled** effectively

TABLE IV
THE COMPLEXITIES OF THE PROPOSED METHODS

Methods	Parameters	Flops	Memory
SC	4.96M	205M	14.2MB
SCHARQ	14.4M	613M	42.1MB
SCHARQ with Sim32	16.7M	689M	48.8MB
[30]	3.33M	127M	12.3MB

higher complexity. The parallel and series SC-RS-HARQ methods have similar complexities as SC. SCHARQ-RS has a similar complexity as SCHARQ. The number of floating-point multiplication additions (Flops), parameters, and required memory are compared in Table IV. All three transmissions of SCHARQ have semantic-based encoders and decoders, and SCHARQ has nearly three times more complexity than SC. Although SCHARQ demonstrates better performance and requires fewer bits than the conventional HARQ methods under high BER, its complexity is increased dramatically. However, SCHARQ is still valuable for wired channel environments where conventional methods cannot transmit any information. In addition, the model-compression methods studied in [30] are a good choice because they can reduce 90% of the parameters with only a small performance loss.

Although excellent state-of-the-art methods, such as SentencePiece [16] and abstractive summarization, can be used to enhance the SC part to save bits, the results of the transmission comparison of these frameworks are not changed. Semantic methods with RS-based HARQ show better performance than conventional coding, but their complexities are increased. End-to-end semantic-based HARQ methods entail more complexity but perform much better than RS-based ones under wicked channel environments.

V. CONCLUSION

We have investigated semantic coding in this study. We combined it with conventional RS channel coding and IR-HARQ and developed two different frameworks, namely, series SC-RS-HARQ and parallel SC-RS-HARQ. By comparing the two proposed frameworks and conventional methods, we found that the semantic encoder has good performance when faced with high BER. However, it cannot guarantee an error-free transmission. Parallel SC-RS-HARQ exploits the different advantages of the semantic architecture

and the conventional method and outperforms the conventional IR-HARQ method. We have also designed a joint source-channel coding and HARQ framework called SCHARQ. This framework is highly flexible and efficient because it can transmit incremental bits to solve the issues of different sentence lengths and varying channel conditions. Thus, it has the best performance among all the competing methods when BER is high, but a little weaker when BER is low. To exploit the full potential of the semantic coder, we proposed a similarity detection approach called Sim32 to detect the semantic error in the estimated sentences and combined it with CRC; the resulting scheme is called CRC-Sim32. The proposed error detection methods allow similar sentences to be received so that many sentences can be transmitted, especially when BER is high. However, several sentences with changed semantic information are still mistakenly received. In the future, further work is needed to improve reliability.

REFERENCES

[1] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[2] Z. Qin, H. Ye, G. Y. Li, and B.-H.-F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019.

[3] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 77–83, Oct. 2019.

[4] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[5] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based joint pilot design and channel estimation for multiuser MIMO channels," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1999–2003, Nov. 2019.

[6] A. M. Elbir, K. Vijay Mishra, M. R. Bhavani Shankar, and B. Ottersten, "A family of deep learning architectures for channel estimation and hybrid beamforming in multi-carrier mm-wave massive MIMO," 2019, *arXiv:1912.10036*.

[7] M. Wenyan, Q. Chenhao, Z. Zhang, and J. Cheng, "Sparse channel estimation and hybrid precoding using deep learning for millimeter wave massive MIMO," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2838–2849, Feb. 2020.

[8] J. Guo, C.-K. Wen, and S. Jin, "CAnet: Uplink-aided downlink channel acquisition in FDD massive MIMO using deep learning," 2021, *arXiv:2101.04377*.

[9] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning-based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[10] H. Ye, L. Liang, G. Y. Li, and B.-H. F. Juang, "Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3133–3143, May 2020.

[11] H. Ye, G. Y. Li, and B.-H. Juang, "Deep learning based end-to-end wireless communication systems without pilots," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 3, pp. 702–714, Sep. 2021.

[12] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, 2020.

[13] N. Strodthoff, B. Göktepe, T. Schierl, C. Hellge, and W. Samek, "Enhanced machine learning techniques for early HARQ feedback prediction in 5G," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2573–2587, Nov. 2019.

[14] G. Qiu, M.-M. Zhao, M. Lei, and M.-J. Zhao, "Throughput maximization for polar coded IR-HARQ using deep reinforcement learning," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–6.

[15] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[16] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," 2018, *arXiv:1808.06226*.

[17] E. Calvanese Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107930.

[18] B. Guler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 4, pp. 787–802, Dec. 2018.

[19] G. Shi *et al.*, "A new communication paradigm: From bit accuracy to semantic fidelity," 2021, *arXiv:2101.12649*.

[20] W. Tong and G. Ye Li, "Nine challenges in artificial intelligence and wireless communications for 6G," 2021, *arXiv:2109.11320*.

[21] C. E. Shannon, W. Weaver, and N. Wiener, "The mathematical theory of communication," *Phys. Today*, vol. 3, no. 9, p. 31, 1950.

[22] J. Bao *et al.*, "Towards a theory of semantic communication," in *Proc. IEEE Netw. Sci. Workshop*, Jun. 2011, pp. 110–117.

[23] E. Bourtsoulatze, D. Burth Kurka, and D. Gunduz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.

[24] C. Lee, J. Lin, P. Chen, and Y. Chang, "Deep learning-constructed joint transmission-recognition for Internet of Things," *IEEE Access*, vol. 7, pp. 76547–76561, 2019.

[25] F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, "Joint source-channel coding for video communications," in *Handbook of Image and Video Processing*, A. Bovik, Ed., 2nd ed. Burlington, MA, USA: Academic, 2005.

[26] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, Aug. 2021.

[27] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2326–2330.

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

[29] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Syst.*, Dec. 2017, pp. 5998–6008.

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

[31] M. Rao, N. Farsad, and A. Goldsmith, "Variable length joint source-channel coding of text using deep neural networks," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5.

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

[33] G. Caire and D. Tuninetti, "The throughput of hybrid-ARQ protocols for the Gaussian collision channel," *IEEE Trans. Inf. Theory*, vol. 47, no. 5, pp. 1971–1988, Jul. 2001.

[34] S. B. Wicker and M. J. Bartz, "Type-II hybrid-ARQ protocols using punctured MDS codes," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 1431–1440, Feb. 1994.

[35] M. L. B. Riediger and P. K. M. Ho, "Application of Reed–Solomon codes with erasure decoding to type-II hybrid ARQ transmission," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 1, Dec. 2003, pp. 55–59.

[36] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1532–1543.

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[38] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Jul. 2002, pp. 311–318.

[39] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," 2017, *arXiv:1703.00395*.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[41] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," *MT Summit*, vol. 5, pp. 79–86, Sep. 2005.
[42] V. Tilavat and Y. Shukla, "Simplification of procedure for decoding Reed–Solomon codes using various algorithms: An introductory survey," *Int. J. Eng. Develop. Res.*, vol. 2, no. 1, pp. 279–283, 2014.

Peiwen Jiang (Graduate Student Member, IEEE) received the B.S. degree from Southeast University, Nanjing, China, in 2019, where he is currently pursuing the Ph.D. degree in information and communications engineering. His research interests include deep learning-based channel estimation, signal detection, and semantic transmission in communications.

Chao-Kai Wen (Senior Member, IEEE) received the Ph.D. degree from the Institute of Communications Engineering, National Tsing Hua University, Taiwan, in 2004. He was with the Industrial Technology Research Institute, Hsinchu, Taiwan, and MediaTek Inc., Hsinchu, from 2004 to 2009, where he was engaged in broadband digital transceiver design. In 2009, he joined the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, where he is currently a Professor. His research interest includes the optimization of wireless multimedia networks.

Shi Jin (Senior Member, IEEE) received the B.S. degree in communications engineering from the Guilin University of Electronic Technology, Guilin, China, in 1996, the M.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003, and the Ph.D. degree in information and communications engineering from Southeast University, Nanjing, in 2007. From June 2007 to October 2009, he was a Research Fellow with the Adastral Park Research Campus, University College London, London, U.K. He is currently with the Faculty of the National Mobile Communications Research Laboratory, Southeast University. His research interests include space time wireless communications, random matrix theory, and information theory. He and his coauthors have been awarded the 2011 IEEE Communications Society Stephen O. Rice Prize Paper Award in the field of communication theory and the 2010 Young Author Best Paper Award by the IEEE Signal Processing Society. He served as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE COMMUNICATIONS LETTERS, and *IET Communications*.

Geoffrey Ye Li (Fellow, IEEE) is currently a Chair Professor at Imperial College London, U.K. Before moving to Imperial in 2020, he was a Professor with the Georgia Institute of Technology, USA, for 20 years and a Principal Technical Staff Member with AT&T Labs - Research, Middletown, NJ, USA, for a period of five years. His research interests include statistical signal processing and machine learning for wireless communications. In the related areas, he has published over 600 journals and conference papers in addition to over 40 granted patents and several books. His publications have been cited over 50,000 times with an H-index over 100 and he has been recognized as a Highly Cited Researcher, by Thomson Reuters, almost every year. He was awarded IEEE Fellow and IET Fellow for his contributions to signal processing for wireless communications. He won several prestigious awards from IEEE Signal Processing, Vehicular Technology, and Communications Societies, including IEEE ComSoc Edwin Howard Armstrong Achievement Award in 2019. He also received the 2015 Distinguished ECE Faculty Achievement Award from the Georgia Tech. He has organized and chaired many international conferences, including the Technical Program Vice-Chair of the IEEE ICC 2003, the General Co-Chair of the IEEE GlobalSIP 2014, the IEEE VTC 2019 (Fall), and the IEEE SPAWC 2020. He has been involved in editorial activities for over 20 technical journals, including the founding Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Special Series on ML in Communications and Networking.