

Toward meaningful communication: Engineering semantics with the
geometry of conceptual spaces

by

Dylan T. Wheeler

A.S., Highland Community College, 2015

B.S., Ottawa University, 2018

M.S., Kansas State University, 2021

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Mike Wiegers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Abstract

As the demand for data transmission around the globe continues to rise, we face a challenge brought about by the fundamental limits of communication imposed by Claude Shannon’s information theory. Modern communication systems are able to operate with a rate close to the theoretical channel capacity, meaning we must increase this capacity in order to transmit additional data. This is achieved by either increasing the power of the signal or bandwidth of the channel, neither of which are desirable options. Alternatively, what if we could communicate more efficiently given the resources we already have? This is the goal of semantic communication, which is a paradigm that aims to communicate a given meaning rather than exactly reproduce bits at the receiver, potentially resulting in efficient semantic representations that can communicate the same meaning using less physical bits. However, modern approaches to semantic communication suffer from a lack of consensus on how to quantify and optimize a system around the notion of “meaning” and a reliance on black-box machine learning models that obscure the true functionality of semantic communication modules. In this dissertation, we introduce an approach termed “semantic communication with conceptual spaces,” which is based on the geometric conceptual spaces model of meaning. Conceptual spaces offer a general, expressive, and interpretable mechanism by which a system can utilize semantic knowledge and optimize its operation around this knowledge. We first lay the mathematical groundwork for such a system, and demonstrate its potential for efficient semantic communication via simulations. We then address a significant challenge of the conceptual space-based approach, which is obtaining the conceptual space model itself. To overcome this challenge, we develop a novel machine learning architecture capable of learning the complex domains of a conceptual space model using only high-level semantic property information. Next, we introduce a causal reasoning-based mechanism into the

proposed system, which allows the semantic communication system to determine which semantic elements are most important for performing a given task. This reasoning mechanism enables the system to achieve even greater efficiency by transmitting only the semantics that are important for the task at hand. Finally, we connect our approach to traditional information theory by deriving an upper bound for the goal-oriented rate distortion function. This bound is based on a fixed threshold for the semantic distortion within the system and the novel notion of distortion discrepancy. In short, this dissertation proposes, for the first time, a method of semantic communication based on a general and expressive model of meaning that allows for automated learning of semantics, causal reasoning-based optimization, and theoretical analysis similar to classical information theory.

Toward meaningful communication: Engineering semantics with the
geometry of conceptual spaces

by

Dylan T. Wheeler

A.S., Highland Community College, 2015

B.S., Ottawa University, 2018

M.S., Kansas State University, 2021

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Mike Wieggers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Approved by:

Major Professor
Dr. Balasubramaniam Natarajan

Copyright

© Dylan Wheeler 2024.

Abstract

As the demand for data transmission around the globe continues to rise, we face a challenge brought about by the fundamental limits of communication imposed by Claude Shannon’s information theory. Modern communication systems are able to operate with a rate close to the theoretical channel capacity, meaning we must increase this capacity in order to transmit additional data. This is achieved by either increasing the power of the signal or bandwidth of the channel, neither of which are desirable options. Alternatively, what if we could communicate more efficiently given the resources we already have? This is the goal of semantic communication, which is a paradigm that aims to communicate a given meaning rather than exactly reproduce bits at the receiver, potentially resulting in efficient semantic representations that can communicate the same meaning using less physical bits. However, modern approaches to semantic communication suffer from a lack of consensus on how to quantify and optimize a system around the notion of “meaning” and a reliance on black-box machine learning models that obscure the true functionality of semantic communication modules. In this dissertation, we introduce an approach termed “semantic communication with conceptual spaces,” which is based on the geometric conceptual spaces model of meaning. Conceptual spaces offer a general, expressive, and interpretable mechanism by which a system can utilize semantic knowledge and optimize its operation around this knowledge. We first lay the mathematical groundwork for such a system, and demonstrate its potential for efficient semantic communication via simulations. We then address a significant challenge of the conceptual space-based approach, which is obtaining the conceptual space model itself. To overcome this challenge, we develop a novel machine learning architecture capable of learning the complex domains of a conceptual space model using only high-level semantic property information. Next, we introduce a causal reasoning-based mechanism into the

proposed system, which allows the semantic communication system to determine which semantic elements are most important for performing a given task. This reasoning mechanism enables the system to achieve even greater efficiency by transmitting only the semantics that are important for the task at hand. Finally, we connect our approach to traditional information theory by deriving an upper bound for the goal-oriented rate distortion function. This bound is based on a fixed threshold for the semantic distortion within the system and the novel notion of distortion discrepancy. In short, this dissertation proposes, for the first time, a method of semantic communication based on a general and expressive model of meaning that allows for automated learning of semantics, causal reasoning-based optimization, and theoretical analysis similar to classical information theory.

Table of Contents

List of Figures	xiii
List of Tables	xvi
List of Nomenclature	xvii
Acknowledgements	xix
Dedication	xx
Preface	xxi
1 Introduction	1
1.1 Motivation	2
1.2 What is Semantic Communication?	4
1.3 Limitations of Current Work	5
1.4 Contributions	7
1.5 Organization	10
2 Literature Review	12
2.1 Outlook and Survey Articles	12
2.2 Overview of Semantics in Communication	15
2.3 Classical Semantic Information	19
2.3.1 Theory of Weakly Semantic Information	19
2.3.2 Theory of Strongly Semantic Information	22
2.3.3 Semantic Information with Truthlikeness	25

2.3.4	Extensions of the Theory of Weakly Semantic Information	27
2.3.5	Summary	32
2.3.6	Challenges and Opportunities	32
2.4	Knowledge Graph-Based Semantic Communication	33
2.4.1	Knowledge Representation with Knowledge Graphs	34
2.4.2	Semantic Similarity Measures on Knowledge Graphs	36
2.4.3	The Semantic Sensor Web	39
2.4.4	Knowledge Graphs for Semantic Communication	41
2.4.5	Working with Knowledge Graphs	44
2.4.6	Summary	45
2.4.7	Challenges and Opportunities	46
2.5	Machine Learning-Based Semantic Communication	47
2.5.1	Deep Learning Methods	47
2.5.2	DeepSC and its Variants	51
2.5.3	Reinforcement Learning Methods	54
2.5.4	Summary	56
2.5.5	Challenges and Opportunities	57
2.6	Significance-Based Semantic Communication	58
2.6.1	Age of Information and Value of Information	61
2.6.2	Semantic Sampling	63
2.6.3	Summary	65
2.6.4	Challenges and Opportunities	65
2.7	Chapter Summary	66
3	Semantic Communication with Conceptual Spaces: Part I	69
3.1	Related Work	70
3.1.1	Chapter Contributions	70
3.2	The Geometry of Meaning	71

3.3	Enabling Semantic Communication	72
3.3.1	The General Model	72
3.3.2	Toward Functional Compression	75
3.4	Experimental Results	76
3.5	Chapter Summary	79
4	Semantic Communication with Conceptual Spaces: Part II	80
4.1	Related Work	80
4.2	Chapter Contributions	81
4.3	Foundations	82
4.3.1	A Geometric View of Semantics	83
4.3.2	Toward Semantic Communication	86
4.4	Semantic Distortion and Error Analysis	88
4.4.1	Semantic Distortion in a Conceptual Space	88
4.4.2	Bounding the Probability of a Semantic Error	90
4.4.3	Practical Implications	94
4.5	Knowledge-Driven Semantic Communication	96
4.5.1	Data-Driven to Reasoning-Driven Intelligence	96
4.5.2	Communicating with Context	98
4.5.3	Ambiguity, Redundancy, and Scalability	99
4.6	Experimental Results	100
4.6.1	Problem Definition and System Description	101
4.6.2	Traditional Communication Components	105
4.6.3	Results and Discussion	105
4.7	Chapter Summary	109
5	Learning Conceptual Spaces	110
5.1	Related Work	110

5.1.1	Chapter Contributions	111
5.2	Problem Definition	111
5.3	Proposed Approach	114
5.3.1	Encoder Module	114
5.3.2	Decoder Module	115
5.3.3	Classifier Module	115
5.3.4	Putting it Together: Domain Learning Algorithm	117
5.4	Experimental Results	118
5.4.1	Experiment 1: MNIST	119
5.4.2	Experiment 2: CelebA	120
5.5	Chapter Summary	122
6	Improved Learning and Causal Reasoning for Semantic Communication	123
6.1	Related Work	124
6.1.1	Chapter Contributions	124
6.2	Background	125
6.2.1	Optimal Transport Theory	125
6.2.2	Causal Machine Learning	126
6.3	Improved Domain Learning	127
6.3.1	Variational Autoencoder	128
6.3.2	Learning With Wasserstein Distance	132
6.3.3	Putting It Together	134
6.4	Conceptual Semantic Communication with Causal Reasoning	135
6.4.1	Reasoning Module	137
6.4.2	The Four Steps of Causal Inference	138
6.4.3	The Semantic Decoder	143
6.5	Experimental Results	144
6.5.1	Setup	144

6.5.2	Domain Learning on Traffic Signs	146
6.5.3	Reasoning for Improved Communication	149
6.6	Chapter Summary	152
7	Semantic Rate Distortion Theory	154
7.1	Related Work	154
7.2	Chapter Contributions	155
7.3	Problem Definition	155
7.4	Supporting Results	157
7.5	Semantic-Functional Rate Distortion	160
7.6	Numerical Results	162
7.7	Chapter Summary	164
8	Conclusion and Directions for Future Work	165
8.1	Summary	165
8.2	Directions for Future Work	166
	Bibliography	168
A	Code Repositories	188
A.1	CSLearn	188
A.2	Chapter 6 Experiment Code	189
A.3	Chapter 7 Experiment Code	189

List of Figures

1.1	Global monthly traffic predictions from mobile data and fixed wireless access (FWA) [1].	2
1.2	Example of two semantically identical but technically distinct statements. . .	3
1.3	Hierarchy of communication goals.	4
2.1	The four existing approaches to SemCom.	16
2.2	Relationship between degrees of inaccuracy, vacuity, and informativeness. . .	25
2.3	Bipartite fact-conclusion graph model of a knowledge base.	31
2.4	An example KG.	35
2.5	Example of an <i>is-a</i> KG, where all edges represent an <i>is-a</i> relationship. . . .	36
2.6	Simplified diagram of the framework proposed in [2] for semantic document analysis.	42
2.7	Basic framework of the multi-modal GAN approach proposed in [3].	49
2.8	Architecture of the end-to-end SemCom system DeepSC proposed in [4]. Each box represents a different section of the network, which are all trained in a joint manner.	52
2.9	Reinforcement learning framework.	55
2.10	Overview of the significance-based semantic architecture proposed in [5]. . .	60
2.11	Example of AoI, where $t_n^{(g)}$ and $t_n^{(r)}$ are the generation and reception time of the n th packet, respectfully.	62
3.1	Color domain. Dashed lines indicate the three dimensions.	71
3.2	General SCCS framework, where the knowledge base represents a CS model.	73

3.3	Simulation results for the SCCS system. Comparison of semantic and syntactic probabilities of error (left) and average end-to-end semantic distortion $\delta(\mathbf{p}^*, \hat{\mathbf{p}})$ (right).	78
4.1	Visualization of the different levels of cognitive representation: associationism (top), conceptual (middle), and symbolic (bottom).	83
4.2	Example Voronoi tessellation with four concept prototypes on $\mathbb{R}^2$.	92
4.3	CS for using VRET to treat acrophobia. Properties, their prototype points and the Voronoi tessellation are included.	101
4.4	Visualization of the BaseCNN architecture used in the practical systems.	103
4.5	The three practical communication systems for VRET.	105
4.6	Plots of average semantic distortion vs. SNR (top) and probability of error vs. SNR (bottom) results for the proposed system with a theoretical semantic encoder under the AWGN channel (left) and Rician fading channel (right).	106
4.7	Probability of semantic error for the proposed system (filled circles) and a fine-tuned version (filled squares), baseline end-end semantic system (open triangles) and baseline non-semantic system (filled triangles), plotted against SNR.	108
5.1	Semantic similarity vs. semantic distortion.	113
5.2	Architecture of the domain learning network.	115
5.3	Architecture of the encoder and decoder used in the experiments.	119
5.4	The learned representations of the validation samples of the MNIST dataset for (a) a basic classifier (b) a basic AE and (c) the proposed domain learning framework.	120
5.5	The learned representations of the validation samples of the CelebA dataset, and decoded representations at the edge of the domain. The domain encodes smiling/not smiling primarily along the x-axis and female/male primarily along the y-axis.	121

6.1	Proposed domain learning framework.	129
6.2	Block diagram of the proposed goal-oriented SCCS framework with reasoning	136
6.3	Semantic decoder architecture.	144
6.4	Scattered 2D learned representations of the shape domain.	148
6.5	Semantic similarity heatmap between the learned property prototypes of the shape domain.	148
6.6	Visualized dimensions of the shape domain, obtained by decoding semantic representations along the axes. The top row of images corresponds to the first quality dimension and the bottom row to the second dimension.	149
6.7	Task accuracy vs. SNR with and without reasoning.	151
6.8	Task accuracy vs. SNR for the proposed and baseline systems for the AWGN channel with BPSK modulation.	152
6.9	Shape classification accuracy vs SNR for the proposed (message length = 2) and baseline systems (Rayleigh fading, 16-QAM modulation).	153
7.1	Flowchart of a $(2^{n_R}, n)$ semantic-functional code.	156

List of Tables

2.1	Summary of works in classical semantic information.	19
2.2	Some states of an example universe.	20
2.3	Summary of classical semantic information measures.	28
2.4	Summary of works in KG-based SemCom	34
2.5	Summary of KG-based semantic similarity measures.	38
2.6	Summary of works in machine learning-based SemCom.	48
2.7	Summary of DL-based SemCom loss functions	54
2.8	Summary of works in significance-based SemCom.	59
2.9	Summary of significance-based semantic measures.	60
3.1	Experimental concepts and prototype values for traffic sign data.	77
3.2	Rate (bits/transmission) and semantic error probability (SNR = 15dB). . . .	79
4.1	Concepts and prototype points ($\mathbf{z}_j$).	102
4.2	Practical systems: models and training details.	104
7.1	Simulation results for various parameter combinations. Dashes indicate that the value is the same as for the base case.	163

Nomenclature

AE	autoencoder
AI	artificial intelligence
AoI	age of information
ATE	average treatment effect
AWGN	additive white Gaussian noise
BCP	Bar-Hillel-Carnap paradox
BPSK	binary phase shift keying
CATE	conditional average treatment effect
CNN	convolutional neural network
CPU	central processing unit
CS	conceptual space
DAG	directed acyclic graph
DCT	dominated convergence theorem
DL	deep learning
DML	double machine learning
DNN	deep neural network
DRL	deep reinforcement learning
ELBO	evidence lower bound
GAN	generative adversarial network
GPU	graphics processing unit

IoT	internet of things
KG	knowledge graph
KL	Kullback-Leibler
LSTM	long short-term memory
ML	machine learning
OT	optimal transport
QAM	quadrature amplitude modulation
RGB	red-green-blue
RL	reinforcement learning
RNN	recurrent neural network
SCCS	semantic communication with conceptual spaces
SCCS-R	semantic communication with conceptual spaces with reasoning
SemCom	semantic communication
SFC	semantic-functional code
SNR	signal-to-noise ratio
SSW	semantic sensor web
TSSI	theory of strongly semantic information
TWSI	theory of weakly semantic information
UAV	unmanned aerial vehicle
VAE	variational autoencoder
VoI	value of information
VRET	virtual reality exposure therapy
WLLN	weak law of large numbers

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Bala Natarajan. Without his excellent guidance and endless support, this work would not have been possible. From him, I have learned many valuable lessons that I will carry with me long after my time at K-State. I would also like to extend a special thanks to my mentor and committee member, Dr. Erin Tripp. Her outstanding mentorship over the course of multiple internships and the ideas that have come from our discussions over the years have had a profound impact on my work and growth as a researcher. I would also like to express my warm gratitude to my committee members, Dr. Don Gruenbacher and Dr. Sanjoy Das, for their valuable feedback and support over the course of this work.

Another person without whom this work would not have been possible is my wife, Macey, whose unending love, support, and motivation kept me focused and made the journey an unforgettable one. I would like to express my sincere gratitude to my parents, Jenny, Matt, and Jennifer, whom I am blessed to be able to look to for inspiration and guidance. I would also like to acknowledge my siblings, Kadyne, Cooper, Camryn, Makayla, and Kennedy, who have been an immense source of joy during the often-difficult process of academic research.

Finally, I would like to thank all of my friends and colleagues I have had the pleasure of getting to know through the CPSWin research group, Dr. Hazhar Sufi Karimi, Dr. Sai Munikoti, Dr. Shweta Dahale, Dr. Mohammad Abujubbeh, Dr. James Ranjith Kumar, Dr. Deepesh Agarwal, Dr. Biswajeet Rout, Aabila Tharzeen, Dr. Amulya Sreejith, Prudhviraaj Dhanapala, Priyanka Gautam, Dr. Rahul Madbhavi, Rakhul Kumar, Niranjana Unnithan, Adwaitha Pradeep, and Reyhane Zare for their support and encouragement.

Part of this material is based upon work that was funded by Air Force Office of Scientific Research grant 21RICO035.

Dedication

To my wonderful wife Macey and sweet daughter Delilah, who give my life meaning and continue to teach me more about communication each and every day.

Preface

“It is we who attach meaning to the bits.”

—Richard Hamming

Communication is simple... or is it? At its core, it is nothing more than one entity sending some signal to be received and processed by another. As humans, we do this very naturally. In fact, more than just sending signals, we are exceptional at conveying *meaning* with these signals. We use expressive languages to convey extremely intricate and complex ideas, most of the time doing this without even thinking about it. We are able to create art that expresses an emotion without using any words. And yet, this second-nature ability is *extremely* difficult to engineer in an artificial system. We have no doubt come closer to achieving this due to the staggering advances in the field of artificial intelligence over the last decade or so, but we are still not there.

The problem at the heart of this dissertation is that of engineering a system to “say what it means,” *concisely* and in a way that the recipient *understands* the intended meaning. The importance of understanding is self-evident. Conciseness is growing in importance because the resources available to us for communication are currently limited (consider this as a “pipe” of a certain size) and the number of devices communicating (trying to send something down the pipe) is growing exponentially with no signs of slowing down. Therefore, concise communication (sending less down the pipe) is of utmost importance going forward.

This is a problem that has intrigued me greatly over the past few years, and this dissertation contains the culmination of my work up to this point aiming to contribute to the solution. It is my hope that it will serve as a useful tool and inspiration for those reading as we continue to strive for more meaningful communication.

Chapter 1

Introduction

Traditionally, communication has been viewed as the problem of one agent (the receiver) exactly reproducing the symbols sent by another (the transmitter). This problem is the basis for Claude Shannon’s momentous theory of communication [6], commonly known as information theory [7], and is referred to as the *technical* problem. However, Shannon and his collaborator Warren Weaver also defined additional levels of the communication problem [8]:

- A. How accurately can the symbols of communication be transmitted?
- B. How precisely do the transmitted symbols convey the desired meaning?
- C. How effectively does the received meaning affect conduct in the desired way?

Level A represents the technical problem, whereas levels B and C represent the *semantic* and *effectiveness* problems, respectively. To date, the engineering of communication systems has been concerned with solving the technical problem and reaching the limits imposed by Shannon’s information theory. As modern systems approach these limits, we seek additional methods to further improve the performance of such systems. A potential path forward involves operating at the higher levels of communication, i.e., the semantic or effective levels. In this introductory chapter, we first motivate the transition to the semantic level. We then briefly review the current state of the art and highlight some limitations. We then provide

an overview of the novel contributions of this dissertation and conclude by describing the organization of the manuscript.

1.1 Motivation

As the world becomes increasingly connected, data transmission demands continue to grow. According to a recent report published by Ericsson [1], monthly global traffic is predicted to grow exponentially over the next five years, as shown in Figure 1.1. Additionally, 5G subscriptions are projected to reach 5.6 billion by the end of 2029, with 9.3 billion total mobile subscriptions worldwide [1].

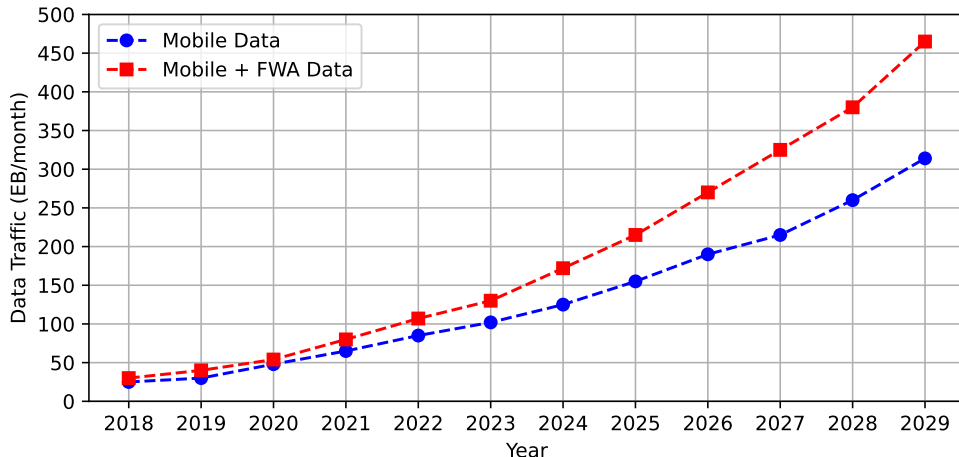


Figure 1.1: Global monthly traffic predictions from mobile data and fixed wireless access (FWA) [1].

In order to meet this growing demand in a sustainable way, we need to rethink the current paradigm of wireless communications. This is due to the information-theoretic fundamental capacity C of a band-limited noisy channel, also known as the *Shannon rate*, which corresponds to [7]

$$C = W \log \left(1 + \frac{P}{N_0 W} \right) \quad \text{bits per second,} \quad (1.1)$$

where W denotes the bandwidth of the wireless channel in hertz, $N_0/2$ is the power spectral density of the additive white Gaussian noise (AWGN) in watts per hertz, and the average

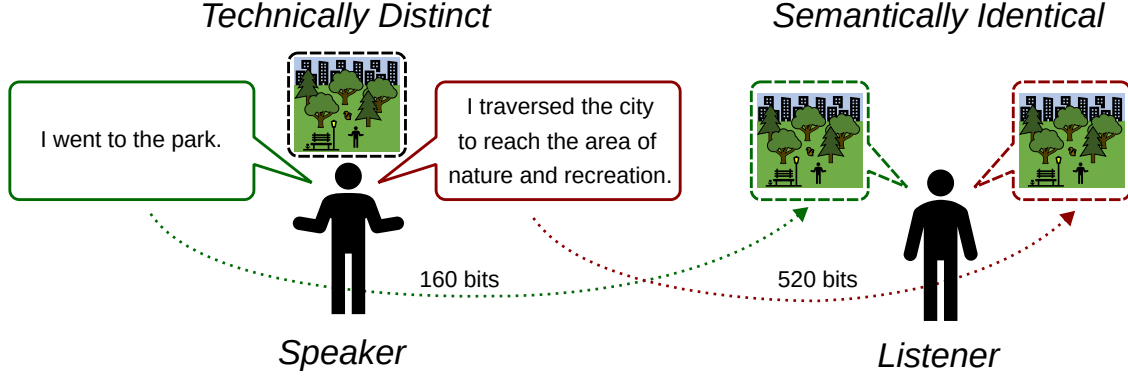


Figure 1.2: Example of two semantically identical but technically distinct statements.

power of the transmitted signal is less than or equal to P watts. From this fundamental limit, we see that classical information theory provides two options to increase the number of bits that can be sent over a channel. The first is to increase the power P of the transmitted signal. With the ever-important goal of designing sustainable systems, increasing power consumption to meet exponentially increasing demand is clearly not desirable. The second option is to utilize more of the wireless spectrum, i.e., increase the bandwidth W of the channel. This approach has been exploited by modern systems, with the advent of millimeter wave technology in 5G networks [9] which operates in the range of tens to hundreds of gigahertz. Moreover, communication over terahertz frequencies has become a popular research topic as of late [10]. However, operating at such high frequencies presents numerous challenges, such as poor propagation characteristics due to the frequency dependency of free-space path loss [11].

To address these challenges, one approach is to push past the information-theoretic limits of technical communication toward a novel paradigm of *semantic communication* (SemCom). As an intuitive example, consider the scenario depicted in Figure 1.2. We see that the two statements represent the same meaning, and are thus *semantically identical*, despite being technically distinct from one another. Suppose the goal of the speaker is to convey this meaning, i.e., perform SemCom. For either statement, the technical rate (in bits per second) is limited by (1.1). However, by choosing the longer statement, it is clear that the speaker is “wasting” bits by not choosing the more efficient technical representation of the intended

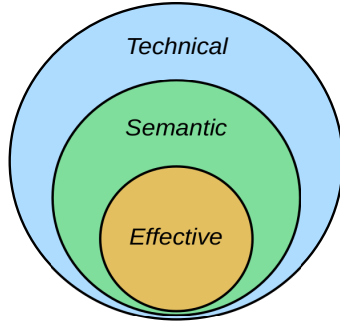


Figure 1.3: Hierarchy of communication goals.

semantics. Extrapolating this simple example to general communication, it follows that by extracting semantically minimal representations of data, we should be able to effectively increase the rate at which we communicate (with respect to the communicated semantics) without increasing either power or bandwidth usage. This intuitive observation is one of the key driving forces behind the recent interest in SemCom.

1.2 What is Semantic Communication?

Lately, SemCom has been a popular term used to describe a wide variety of approaches, and thus we will provide some clarification of what we mean when using this term here. First consider the hierarchy of communication goals shown in Figure 1.3. Note that all communication must be technical; some kind of signal must be sent for communication to take place. A subset of these communication goals will also be semantic, where the sender wants to convey some meaning to the listener. An even smaller subset will also be effective, where the sender has a desired action they would like the listener to take.

For example, suppose the sender transmits the text “The oven is in the kitchen” to the listener. If the goal is purely technical, then accurate transmission of each character is the sole objective. However, perhaps the sender is attempting to convey the layout of their home to the listener; then the meaning of the words is the focus, e.g., the listener should understand that the oven is in the kitchen and not the bedroom. But perhaps the sender wants the listener to turn the oven on; now the goal is effective as well.

When we refer to SemCom, we mean communication problems where the goal falls into the semantic level of this hierarchy. Therefore, we do not mean using semantics to exactly reproduce the original data, as this falls into the technical level. Likewise, we do not mean higher-level goal-oriented or task-oriented communication, although achieving SemCom brings us closer to realizing this kind of communication, as many tasks depend on the underlying meaning. When dealing with the effectiveness problem, we will explicitly refer to communication as goal-oriented or task-oriented SemCom. Otherwise, for basic SemCom we are concerned with problems where the primary goal is to convey some meaning.

1.3 Limitations of Current Work

In recent years, interest in SemCom has surged due to the unprecedented success of artificial intelligence (AI) and machine learning (ML) techniques. Numerous techniques and approaches have been put forth, however, SemCom remains a primarily theoretical development with some key limitations holding back its practical implementation. In this subsection, we provide an overview of some of the key works within the field of SemCom and discuss their limitations. A more thorough review of the current literature can be found in Chapter 2.

The first attempt at developing a theory of SemCom originates with Carnap and Bar-Hillel in 1954 [12], when they sought to extend classical notions of information theory to semantic information. The key difference was that their so-called theory of semantic information utilized *logical* probabilities rather than statistical characteristics of data. Over the years, some have attempted to extend this approach by utilizing the notions of truth [13] and “truthlikeness” [14]. More recently, there have been some other notable works taking the information-theoretic approach. The authors of [15] discuss semantics from the perspective of rate-distortion theory, where the distortion measure is assumed to capture the relevant semantics. They note, however, that “quantifying the semantic distortion measure for a particular information source is a formidable task.” The authors of [16] also study SemCom from a rate-distortion perspective, while the traditional logical probability-based approach is

revisited in [17]. The main difficulty in these approaches stems from the problem of defining an adequately expressive knowledge base from symbolic logic or the challenge of defining a semantic distortion measure.

The second category includes methods that center around representing meaning with a knowledge graph (KG) structure. With this approach, meaning is expressed through relations in the KG, for which distortion measures can be derived. The authors of [18] propose an edge-sharing KG to enable SemCom among intelligent edge devices in a network. In [19], the authors develop a communication system powered by a KG with subject-relation-object triples for transmission of images among unmanned aerial vehicles (UAVs). Furthermore, in [20] a framework exploiting a KG has been developed for cross-modal SemCom. Like the information-theoretic approach, the symbolic nature of KGs can make it difficult to define a fully expressive knowledge base. Defining semantic distortion in this setting can be difficult as well, although some link-based [21], node-based [22], and combinations of the two [23] do exist.

Third, the ML-based approach to SemCom has been by far the most popular approach as of late; some recent, notable publications include [24–28]. Most of the ML-based approaches to SemCom were inspired by the initial success of DeepSC [4], which is a system implementing SemCom of text data. The key idea with this approach is to train a neural network, often in an end-to-end manner, to “learn” how to best encode and transmit data with respect to some loss function. Therefore, this approach shares some similarities with the rate-distortion approach discussed earlier when the loss function is designed to minimize some semantic distortion. However, many works utilizing this approach (including DeepSC) train the system with the goal of exactly reproducing the initial data [4, 28], i.e., they are still addressing only the technical problem of communication. Others train the network to minimize some goal-oriented loss function [24, 29], which neglects the semantics of the problem. Both of these approaches are commonly referred to as SemCom, though they would be more accurately described as advanced technical or goal-driven coding schemes, since they are not primarily concerned with the conveyance of meaning.

Finally, a fourth approach to SemCom emphasizes the *significance* of information. This

significance-based communication is again more aligned with the effectiveness problem than the semantic problem, and is defined as the “provisioning of the right and significant piece of information to the right point of computation (or actuation) at the right point in time” [5]. A key idea within this approach is semantic sampling [30], in which devices have some control over their own traffic and produce only relevant information from the start [30]. Like the goal-driven ML-based schemes mentioned above, these approaches largely neglect the semantic problem of communication.

The major shortcomings of these approaches can be summarized as follows:

- *Limited expressivity*: Knowledge representation based on symbolic logic or KGs can be limited in the level of detail and nuance that can be captured. ML-based and significance-based approaches mostly neglect knowledge representation altogether and only implicitly address the underlying semantics through higher level goal-oriented techniques.
- *Lack of a formal definition of meaning*: The meaning of a symbol or piece of data is an ambiguous entity and has been difficult to work with. Despite this, if we wish to engineer meaning-oriented communication, we need a rigorous and concrete methodology of handling meaning.
- *Difficulty capturing semantic distortion*: Stemming from the trouble in adequately defining meaning, the discussed approaches have a hard time expressing semantic distortion or semantic similarity. Some attempts have been made for information-theoretic and KG-based approaches, though there does not seem to be any widely-accepted metrics. Existing measures for ML-based and significance-based approaches typically reflect either technical or goal-oriented performance.

1.4 Contributions

In light of the limitations discussed in the previous section, in this dissertation we propose a method of SemCom that brings the importance of modeling human-interpretable meaning

to the forefront. This method addresses each of the limitations discussed, i.e., it provides a formal, rigorous definition of meaning and semantic distortion, and is general enough to express complex notions of meaning. Specifically, we propose the use of the *conceptual spaces* (CSs) model of meaning, initially developed by cognitive scientist Peter Gärdenfors [31, 32], for representing the meaning in a SemCom system. Such a model allows for general expressivity of abstract notions, a formal geometric model for working with meaning, and a built-in notion of semantic distortion derived from the geometry of the space. In the following paragraphs, we provide an overview of our primary contributions as well as the original articles corresponding to each.

We survey the literature related to SemCom, reviewing existing techniques for modeling meaning, including classical semantic information-based, KG-based, ML-based, and significance-based approaches. The primary challenges associated with each approach as well as promising directions for future research are identified and discussed. More information can be found in Chapter 2 and in the following article:

- **D. Wheeler** and B. Natarajan, “Engineering Semantic Communication: A Survey,” in *IEEE Access*, vol. 11, pp. 13965-13995, 2023.

We introduce and formalize the novel *semantic communication with conceptual spaces* (SCCS) model of communication. A preliminary investigation is performed, simulating a basic SCCS system with CS-enabled semantic encoder and decoder, and demonstrate competitive semantic performance with massive reduction in rate as compared to a more traditional communication setup. Encouraged by these results, we rigorously define the components of the SCCS system and derive theoretical semantic performance bounds. Extensive simulations are conducted for a metaverse-inspired use case, again demonstrating large performance gains as compared to baseline approaches. More details can be found in Chapters 3 and 4, as well as the following articles:

- **D. Wheeler**, E. E. Tripp and B. Natarajan, “Semantic Communication With Conceptual Spaces,” in *IEEE Communications Letters*, vol. 27, no. 2, pp. 532-535, Feb. 2023.

- **D. Wheeler** and B. Natarajan, “Knowledge-Driven Semantic Communication Enabled by the Geometry of Meaning,” currently under review for publication in *Journal of Communications and Networks*.

The significant challenge of obtaining the CS model for SCCS is addressed by leveraging an autoencoder (AE)-based framework for learning domains of the CS. This framework requires only high-level semantic property information and balances data reconstruction and semantic classification loss terms (with semantic distance regularization) to learn the geometry of a domain. This framework is further improved by incorporating the theoretically-grounded variational AE (VAE) architecture and the Wasserstein loss function to allow for implicit regularization and the incorporation of prior knowledge. Simulations illustrate the ability of the proposed framework to learn representative and interpretable semantic domains. More information can be found in Chapter 5 and 6, as well as the following articles:

- **D. Wheeler** and B. Natarajan, “Autoencoder-Based Domain Learning for Semantic Communication with Conceptual Spaces,” *2024 Wireless Telecommunications Symposium (WTS)*, Oakland, CA, USA, 2024 (Best Graduate Student Paper award)
- **D. Wheeler**, B. Natarajan, “Conceptual Learning and Causal Reasoning for Semantic Communication,” *TechRxiv*, July 15, 2024, currently under review for publication in *IEEE Transactions on Cognitive Communications and Networking*.

A causality-based reasoning mechanism is developed for SCCS, allowing the system to determine which semantic elements of the data are most important for performing a task. By selecting only the most important semantics to transmit to the receiver, even greater efficiency is achieved. Simulations on a real-world image dataset highlight the practical potential of the proposed reasoning-based approach. These improvements are detailed in Chapter 6 and the following article:

- **D. Wheeler**, B. Natarajan, “Conceptual Learning and Causal Reasoning for Semantic Communication,” *TechRxiv*, July 15, 2024, currently under review for publication in *IEEE Transactions on Cognitive Communications and Networking*.

We explore the theoretical implications the SCCS system through the lens of rate distortion theory, developing a bound on the goal-oriented rate distortion function of a model SCCS system based on a semantic distortion threshold and the novel notion of distortion discrepancy. Numerical experiments are conducted to illustrate the behavior of a practical coding scheme based on these theoretical results. More details can be found in Chapter 7 and the following article:

- **D. Wheeler**, B. Natarajan, “Meaningful Goals: Toward a Semantic-Functional Rate Distortion Theory,” *TechRxiv*, August 15, 2024, currently under review for publication in *IEEE Communications Letters*.

1.5 Organization

The rest of this dissertation is organized as follows. Chapter 2 contains a detailed survey of different approaches to SemCom, divided by the way in which each handles the notion of meaning. Each section focuses on a specific approach, with tables summarizing the surveyed papers and mathematical measures related to that approach. Chapter 3 presents a preliminary investigation into our proposed SCCS approach, and Chapter 4 explores this approach further by improving the formality of the key pieces and proving some bounds on the probability of a semantic error under the proposed framework. Both Chapters 3 and 4 include empirical simulation studies to further illustrate the practical behavior of the proposed SCCS methodology. Chapter 5 addresses one of the primary challenges with the SCCS approach, namely, obtaining the CS model which serves as the knowledge base for communication. In this chapter, an AE-based ML architecture is proposed to learn the domains of a CS using high-level property information. Chapter 6 innovates on this learning approach by 1) replacing the AE module with a VAE module for improved regularization and 2) using a Wasserstein distance-based loss function to allow for the incorporation of prior domain knowledge. Chapters 5 and 6 also include empirical simulation studies to demonstrate the performance of the proposed learning methods. Additionally, Chapter 6 introduces a causality-based reasoning mechanism into the SCCS system which allows the

system to determine which semantic elements are most important for a given task, and only send these important elements to reduce communication overhead. Simulations are provided to demonstrate the benefit of this reasoning-driven approach. In Chapter 7, we investigate the theoretical limits of SemCom and derive an upper bound on the rate distortion function of a goal-oriented SemCom system inspired by the SCCS set up. In addition to the theory, we simulate a practical coding scheme based on these results and compare performance to that of an ideal, optimal coding scheme. Chapter 8 concludes the dissertation and discusses directions for future work leveraging the foundation laid throughout these pages.

Chapter 2

Literature Review

In this chapter we provide a review of recent literature that pertains to the topic of SemCom. We first discuss other related works, including those that outline a broad vision for SemCom and those that make an attempt to survey the field. We then provide a brief overview of the different approaches to utilizing semantics in communication systems. The following sections survey works related to these approaches, each concluding with a brief summary and a discussion of the challenges and opportunities related to that specific approach.

2.1 Outlook and Survey Articles

There have been several published works that attempt to provide a vision of what SemCom might look like, and even fewer that attempt to survey the field. While most of these have come about in recent years, one of the earliest examples was published in 1992 by Ouksel and Naiman [33], in which they discuss a SemCom protocol in heterogeneous database systems. They argue that a SemCom protocol provides a more flexible vehicle of communication and can support effective conflict resolution. Although they refer to this protocol as SemCom, their ideas align much more with the concept of the *semantic web*, which was first introduced by Berners-Lee *et al.* in 2001 [34]. While related, the idea of the semantic web is different from what we refer to as SemCom. The primary goal of the semantic web is to make the

information contained in pages on the internet machine-interpretable. This field has been well-studied over the years since its inception. The semantic web can be thought of as an attempt to achieve the second (semantic) level of communication between the web (source) and a machine (receiver), and thus represents a particular case of general SemCom.

One of the earliest works providing a vision for addressing the SemCom problem was given by Rodoplu *et al.* in 2007 [35]. They introduce their idea of a semantic domain, which includes semantic “atoms” and corresponding atomic operations that act on the atoms to generate objects. They then provide their vision of how one might characterize semantic information, making the important observation that the same object might have different semantic information measures in different domains. The vision provided in [35] leans on the idea of using KGs to represent prior knowledge bases at both the transmitter and receiver, falling within the realm of what we refer to as KG-based SemCom, which is the focus of Section 2.4.

Another vision article from 2013 [36] focuses specifically on the problem of semantic misunderstanding. An illustrative example given is the failure of the Mars Climate Orbiter, in which two collaborating teams of engineers were working in different unit systems (imperial and metric), leading to misunderstandings and failure of the mission. The article focuses more on the effectiveness problem rather than the semantic problem, and argues that the key theoretical notion for successful communication is the presence of *sensing*; which is described as feedback to the source indicating successful communication. The vision presented in this article describes agents in a communication system that can learn to achieve successful communication (in the effectiveness sense) despite some initial knowledge mismatch between source and receiver.

In [37], a brief history is provided on the quantification and transmission of information and intelligence. In addition to providing an informative summary on the history of communication techniques and theory, the article raises some important challenges for the design of future intelligent communication, namely:

- Can the formulation of channel capacity include a function of significance?

- How do we define error in the transmission of intelligence?
- How can the code set and signal shaping be defined to support optimal transmission of intelligence?
- How can the receiver be design to optimally accomplish reception of intelligence?

By interpreting the phrase “transmission of intelligence” as the nearly synonymous term of SemCom, these represent some of the grand challenges of practically achieving this higher level of communication. One possible avenue is proposed to address these challenges, which suggests the use of Bayes’ decision theory to quantify the “significance” of information.

Building on the idea of information significance, in Uysal *et al.* [5] envision semantics to mean just that: the significance of information as opposed to its meaning. They argue for an extensive cross-layer optimization of the end-to-end communication system, in a self-described *radical departure* from the well-established way of assessing communication networks. The key idea is that this optimization will yield SemCom by “the provisioning of the right piece of information to the right point of computation (or actuation) at the right point in time.” The development of semantic measures is called for to quantify what information is “right” such that the system can be properly optimized. The significance-focused view on semantics is also advocated for by the authors of [30], who describe a vision of goal-oriented SemCom, essentially combining the semantic and effectiveness problems. In addition to outlining this vision, utilizing these ideas is shown to greatly reduce robotic actuation error in a provided example.

A recent article [38] outlining nine challenges in AI for 6G communications points out the capabilities of recent learning techniques as a potential enabler of SemCom. The article outlines two challenges which must be met to develop SemCom, the first of which is the mathematical foundation of SemCom; while some attempts have been made at defining a mathematical framework on which to build SemCom, we are still lacking a comprehensive theory. The second challenge is the structure of SemCom systems, which is posed as a problem of choosing between a general deep neural network (DNN) or further exploring other structural levels of communication to facilitate SemCom. The authors of another

vision article [39] promote the use of ML in SemCom for 6G networks. This extensive article provides a complete view on the current vision for 6G networks, while also discussing details of both semantic and goal-oriented (effective) communications, with various examples and applications given for each. The authors then provide their vision of the 6G network as incorporating online learning-based communication and control, focusing on ML and applications of SemCom. Similar to the articles [38] and [39], the authors of [40] provide an overview of end-to-end SemCom based on DL. The discussion is broken into SemCom for different modalities, such as text, image and speech. Use cases, including internet of things (IoT) networks and smart factories are discussed, and open issues are presented.

The work that is perhaps the most similar to that presented throughout this chapter is the recent review published by Lan *et al.* [41], in which the authors review principles of SemCom, discuss existing system architecture designs, and provide an overview of designing SemCom systems based on KGs. Their discussion is divided into the categories of human-human, human-machine, and machine-machine communication, with example applications provided for each. While sharing many similarities, [41] differs from the work here in a few ways. First, the discussion contained in [41] is application-centric, while we aim to focus more on the general techniques and foundations of SemCom. This leads to a natural contrast in presentation; [41] breaks the discussion into human-human, human-machine, and machine-machine based on the *application* of SemCom, while we partition topics in our discussion based on the *definition* of semantics in communication. Furthermore, unlike [41], we discuss the classical approaches to quantify semantic information, in order to provide perspective on how the field has evolved since its inception.

2.2 Overview of Semantics in Communication

The ambiguity of the word “semantics” brings with it an inherent difficulty when attempting to provide a definition. Indeed, this is an issue that has drawn the attention of engineers and philosophers alike. In [42], a brief discussion regarding the philosophical context of SemCom is provided. There, it is noted that the idea that “communication must be considered as

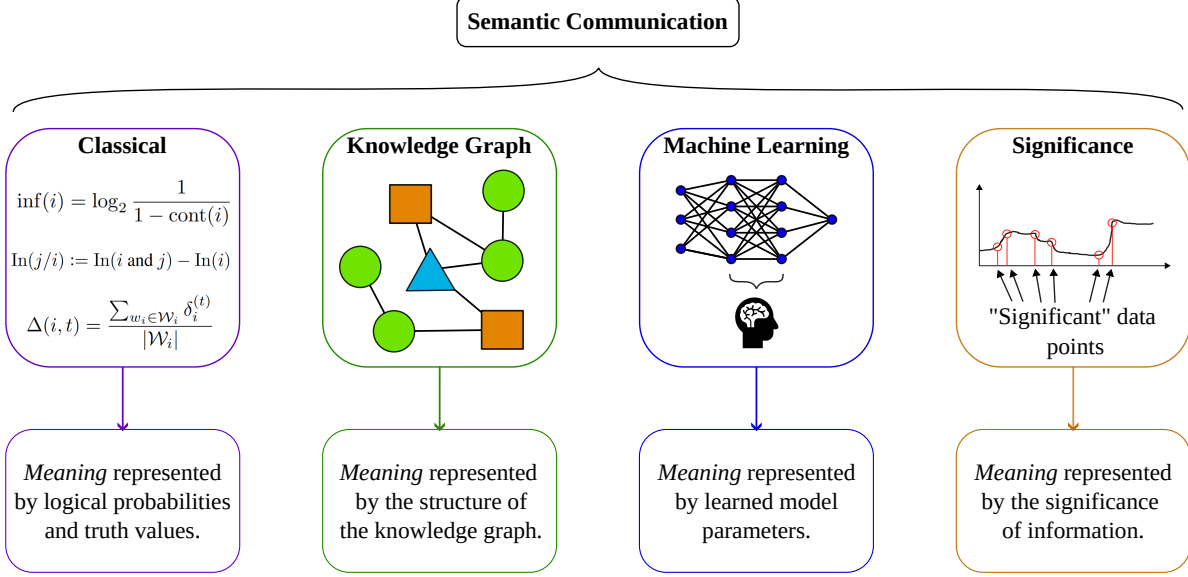


Figure 2.1: The four existing approaches to SemCom.

a means to an end” was first brought about by Dewey in 1925 [43], and later “brought to the forefront of philosophy” by Wittgenstein in 1953 [44]. Wittgenstein filled a short book, organized into a continuous flow of philosophical remarks, with his thoughts and reflections on the fundamental aspects of language. Clearly then, having been at the center of a great amount of philosophical thought, the definition of semantics is a complex one.

However, as we are interested in the engineering of SemCom, a clear definition is vital. Without it, we are left blind when attempting to create systems which may achieve this higher level of communication. The solution to any engineering problem requires full understanding of the problem itself, including any simplifications, assumptions, and constraints posed by the problem. Therefore, we dedicate this section to a brief introduction to the different engineering problems that have been posed when attempting to achieve SemCom. Inherent in each is the way in which this abstract idea of semantics is defined, which leads to different implications and solutions. Figure 2.1 provides a high-level view of these approaches and the ways that meaning is represented by each.

As previously mentioned, in their 1952 paper Carnap and Bar-Hillel attempt to outline a theory of SemCom [12] as a direct response to Shannon’s then-recently published “A Mathematical Theory of Communication” [6]. The core of their work is to base information

measures around *logical* probabilities rather than the *statistical* probabilities which underlie what we now call information theory. This definition of semantics is concerned with the so-called logical truth of a statement, from which information measures can be derived. We classify this definition and its derivatives *classical semantic information*. However, it is noted in [12] that this definition of “semantic information is a concept more readily applicable to psychological and other investigations than its communicational counterpart.” Regardless, there are still some important works that embrace this idea of semantic information for the engineering problem.

Perhaps the most pervasive method of defining semantics throughout the literature is to do so by using some sort of structured knowledge base. This structured knowledge base can take on many names, such as semantic network [45], taxonomy [21], ontology [46], and others. All of these essentially refer to the same idea, which is to use a graph structure, or *KG* to represent knowledge in the system [47]. Hence, we refer to these techniques as *KG-based SemCom*. Having close ties to the semantic web, it is clear why this approach is popular. By defining knowledge over a graph, it is relatively straightforward to define metrics of “semantic similarity,” which can then be analyzed using well-developed graph theory techniques. Furthermore, recent work on graph neural networks brings about the opportunity to incorporate modern learning techniques over such graphs [48].

Another prevalent approach that is seeing a surge of interest is the idea of using ML techniques to “learn” the semantics of a problem. Akin to model-based vs. data-driven approaches to general inference problems (see [49] for more on this), predefined KGs impose model-based semantics on the problem at hand while ML methods use data to determine these semantics. Borrowing techniques from natural language processing (NLP) and computer vision, deep networks can be taught to communicate in the most efficient manner while preserving semantic content [4]. Similarly, by implementing reinforcement learning (RL) methods, these networks can be refined over time and even adapt to dynamic changes in the communication problem. We refer to this approach as *ML-based SemCom*. Initial studies examining this approach to SemCom over several modalities have emerged in recent years [4, 50–52].

Finally, a fourth definition which differs significantly from those previously described is the one first mentioned in Section 2.1: the idea of semantics as the “significance” of information; we call this *significance-based SemCom*. While classified as an approach toward SemCom, this approach essentially addresses the third level of communication problem: the *effectiveness* problem. Rather than concern ourselves with the meaning of a message, advocates of this approach call for communication of the *right* information. Of course, what information is “right” will depend on the application and the desired outcome, therefore leading to the idea of *effective* or *goal-oriented communication*. This approach lends itself well to machine-machine communication, in which we are less concerned with the conveyance of “meaning” and more concerned with what the system accomplishes. One well-studied way of quantifying what information is “right” is the popular age of information (AoI) metric [53], which relates to generating and delivering information at the right time. By defining other metrics to quantify what is “right” for the problem at hand, a joint optimization can be carried out to achieve optimal communication.

These four approaches to defining semantics roughly partition the previous research regarding SemCom. Based on this analysis of the current literature, we broadly define semantics as *any definition of information or the transfer thereof that considers something beyond the statistical nature of the symbols used to represent that information*. This view unites the definitions discussed above, despite their differences in the “something beyond” that is considered by each. Of course, these definitions are not mutually exclusive. ML can be used to determine which information is “right,” just as classical semantic information metrics can be used to design and tune neural networks. Just as model-based deep learning (DL) incorporates both prior knowledge and data-driven techniques for inference, graph neural networks can be used to learn KGs for SemCom. However, by treating each of these definitions individually, a complete picture is given of the current state of SemCom. In the next sections, we dive into each definition, the engineering approaches that come as a result, and some of their potentials and challenges.

Table 2.1: Summary of works in classical semantic information.

Theory of Weakly Semantic Information	Carnap & Bar-Hillel [12]	Semantic information using logical probabilities
	Bao <i>et al.</i> [54]	Definition of semantic entropy and theorems
	Basu <i>et al.</i> [55]	Semantic compression
Theory of Strongly Semantic Information	Floridi [13]	Extension of TWSI using truth values
Truthlikeness	D’Alfonso [14]	Semantic information based on truthlikeness

2.3 Classical Semantic Information

As mentioned earlier, Carnap and Bar-Hillel’s paper *An Outline of a Theory of Semantic Information* attempts to tackle the problem of engineering SemCom through the quantification of semantic information [12]. This quantification is the first step towards efficient SemCom, as it allows us to consider ideas such as semantic compression and semantic error. Therefore, we begin our discussion with the ideas and methods related to classical semantic information. The works discussed in this section are summarized in Table 2.1.

2.3.1 Theory of Weakly Semantic Information

Following the convention of [13], we will refer to the theory laid out by Carnap and Bar-Hillel as the Theory of Weakly Semantic Information (TWSI), for reasons that will be discussed later. TWSI is defined over a language system $\mathcal{L}_n^\pi$ which contains n “things” (or individuals) and π primitive one-place predicates (descriptors). An *atomic* sentence is said to consist of a single predicate describing a single thing, while a *molecular* sentence is formed from two or more atomic sentences joined with some logical connective, including: or, and, if...then, if and only if. Any sentence can either be logically true, logically false, or logically indeterminate. Furthermore, *logical relations* are defined. For example, for sentences i and j , we have i *logically implies* j defined to mean that “if i then j ” is logically true. A *state description* is a sentence in which each of the π predicates is specified for each of the n individuals; thus completely specifying all aspects of the universe. Common set notation can be used to talk about “classes” of entities within the universe. For example, [12] describes a system consisting of three individuals, $\{a, b, c\}$, and two binary predicates, young or old (Y or O)

and male or female (M or F). Then an example of a state description could be given as “(a is F and Y) and (b is M and Y) and (c is M and O).” Some other possible states are given below in Table 2.2.

Table 2.2: Some states of an example universe.

a		b		c	
M/F	Y/O	M/F	Y/O	M/F	Y/O
F	Y	M	Y	M	O
F	O	M	O	M	O
M	Y	M	Y	M	Y
F	O	M	Y	F	Y

Note that, for binary predicates, a universe will consist of $2^{n\pi}$ possible state descriptions. Similar to the process in which Shannon developed entropy as a measure of information [6], Carnap and Bar-Hillel begin with axioms that semantic information (and its corresponding measures) must satisfy. Denoting the semantic information of a sentence as $\text{In}(\cdot)$, the first axiom is given as

$$\text{In}(i) \text{ includes } \text{In}(j) \iff i \text{ logically-implies } j, \quad (2.1)$$

meaning that i says everything that is said by j (and possibly more) if and only if j is implied by i within the logical framework. For example, take $i = (a \text{ is } F \text{ and } Y)$ and $j = (a \text{ is } F)$. Clearly, j is implied by i , and thus the semantic information of i includes that of j (though the converse is not true). This axiom requires us to treat information as a *set* or *class* of something; it is important to note that the *amount* of information is arbitrary at the moment, and must be defined on this set. Some theorems are derived from this axiom, and the concept of *relative information* is defined as

$$\text{In}(j/i) := \text{In}(i \text{ and } j) - \text{In}(i) \quad (2.2)$$

where $\text{In}(j/i)$ is again some set or class. Continuing the example above, we can see that $\text{In}(j/i) = \emptyset$, since $\text{In}(i \text{ and } j) = \text{In}(i)$. However, $\text{In}(i/j) \neq \emptyset$.

Based on (2.1) and (2.2), a concept of the information of a sentence is defined, and

is termed the *content* of a sentence. Content is derived from what is called a *content-element*, which is simply the negation of a state description. The content of a sentence i , denoted $\text{Cont}(i)$, is then defined as the set of content-elements logically implied by i . Intuitively, $\text{Cont}(i)$ can be thought of as the set of state descriptions in the overall state-space (i.e. universe described by our language system) that are *eliminated* with knowledge of the sentence i . Take $i = (a \text{ is } F \text{ and } Y)$ as before. Out of the $2^6 = 64$ possible state descriptions in the universe, this sentence eliminates all 48 in which a is not F or a is not Y , and $\text{Cont}(i)$ is composed of the negations of all such state descriptions. Clearly then, a self-contradiction will “say the most” (by eliminating all state-descriptions) and a tautology will “say the least” (by eliminating no state-descriptions). Similarly, a complete state description can be thought of as carrying a great amount of information, since it eliminates all other state descriptions. Note that this notion of information has nothing to do with the truth of a sentence, which is a point we will revisit. With this idea of information in place, the primary question is addressed: how shall the *amount* of information be defined?

The amount of semantic information carried by sentence i is denoted as $\text{in}(i)$, and the following requirements are stated:

$$\text{Cont}(j) \subseteq \text{Cont}(i) \Rightarrow \text{in}(i) \geq \text{in}(j) \quad (2.3)$$

$$\text{Cont}(j) \subset \text{Cont}(i) \Rightarrow \text{in}(i) > \text{in}(j) \quad (2.4)$$

$$\text{Cont}(i) = \emptyset \Rightarrow \text{in}(i) = 0 \quad (2.5)$$

Note the subtle difference between $\text{In}(\cdot)$ and $\text{in}(\cdot)$; $\text{In}(\cdot)$ represents the information itself, while $\text{in}(\cdot)$ is used to quantify *how much* information is given by $\text{In}(\cdot)$. Using $\text{Cont}(\cdot)$ as the information content of a sentence, these requirements are straightforward and make intuitive sense. By (2.3) and (2.4), a sentence containing all the information of another should have a greater than or equal amount of information, with equality only if the information carried is the same. By (2.5), the amount of information is zero if the information of a sentence is the empty set.

Based on these requirements, two measures are offered as the main contribution of [12]. The first is termed the *content-measure* of a sentence, denoted $\text{cont}(\cdot)$ (different from $\text{Cont}(\cdot)$), and is defined as any proper m -function of the negation of a sentence. We will not go into the details of what constitutes an m -function here (see [12, Section 6] for further reading), but suffice it to say that it satisfies (2.3)-(2.5), and defines a measure taking values between 0 and 1, thus representing a logical probability measure. However, a problem arises with this measure regarding another intuitive requirement not yet stated, namely additivity. Just as in classic information theory, we would like the information measure of two *independent* sentences to follow additivity, i.e. $\text{in}(i \text{ and } j) = \text{in}(i) + \text{in}(j)$ for i, j independent; it is shown that $\text{cont}(\cdot)$ does not satisfy this intuition [12, Thm. 6-15]. Thus, a second measure is proposed and is termed *measure of information*, denoted by $\text{inf}(\cdot)$, not to be confused with the infimum operator. This second measure is defined as

$$\text{inf}(i) = \log_2 \frac{1}{1 - \text{cont}(i)}. \quad (2.6)$$

Observe that this measure is analogous to the classical information-theoretic definition of entropy, making use of the logical probability $\text{cont}(i)$ instead of the statistical probability $p(i)$.

A comparison of these two measures is given in [12], and it is shown that both exhibit intuitively desirable properties, and likewise they both exhibit intuitively undesirable properties. The lack of additivity of $\text{cont}(\cdot)$ is one example. Another is that $\text{inf}(\cdot)$ lacks a counterpart to the property of $\text{cont}(\cdot)$ stating that $\text{cont}(i \text{ and } j) \leq \text{cont}(i) + \text{cont}(j)$. Thus, it is concluded that neither represent an ideal measure of semantic information, but rather that they both have specific strengths and weaknesses.

2.3.2 Theory of Strongly Semantic Information

A problem with TWSI occurs when presented with a sentence that constitutes a contradiction, e.g., “ i and not i .” As mentioned above, under the definition of $\text{Cont}(\cdot)$, this sentence

would carry with it maximum semantic information. Intuitively, however, we know that a contradiction *should* carry no information; it is obviously untrue and leaves the receiver no less informed than before the reception of the message. This ambiguity manifests itself in the mathematics of TWSI as well, and is known as the Bar-Hillel-Carnap Paradox (BCP). Therefore, in [13] Luciano Floridi proposes that *truth* lies at the root of this paradox, which can be solved by the incorporation of truthfulness considerations into TWSI.

The theory which follows is outlined in [13] and is deemed a Theory of Strongly Semantic Information (TSSI). Again, the goal is to develop a theory from which semantic information can be quantified, which would clearly be a useful theory for the engineering of SemCom. As a starting point, three desiderata are given:

D.1 Avoid any counterintuitive inequality comparable to BCP

D.2 Treat the alethic (truth) of a sentence not as a supervenient but as a necessary feature

D.3 Extend a quantitative analysis to the whole family of information-related concepts

The core of TSSI is the definition of degrees of vacuity and inaccuracy. The intuitive idea is that semantic information is related to “how true” and “how false” a sentence is. A highly vacuous sentence is one that is true, but carries with it little information. Similarly, a highly inaccurate sentence is false and also carries little information. This brings about the strange idea that a false sentence may carry more information than a true sentence. As an example, consider the system of three individuals and two predicates described earlier. The tautology $i = “(a \text{ is } F) \text{ or } (a \text{ is } M)”$ clearly provides no information, yet it is always true nonetheless. In contrast, consider the sentence $j = “(a \text{ is } F) \text{ and } (b \text{ is } M) \text{ and } (c \text{ is } F) \text{ and } (a \text{ is } Y) \text{ and } (b \text{ is } Y) \text{ and } (c \text{ is } O)”$ when c is actually M . While j is false, it is not too difficult to reason that it carries more information than i .

Mathematically, these concepts are formalized as a positive or negative degree of “semantic distance” of a sentence i from a fixed point, which is defined as the given situation w to which i is supposed to refer. True statements take on positive degrees between 0 and 1, while false statements take on negative degrees between -1 and 0. This mapping is denoted by

the function $f(i)$. For false statements, the *degree of inaccuracy* simply counts the number of false atomic statements e in i and divides by the total number of atomic statements, or the *length* l of i

$$f(i) = -e(i)/l(i), \quad (2.7)$$

where i is a false sentence. On the other hand, the *degree of vacuity* is more difficult to define since all atomic statements of a true sentence are true. Therefore, this degree is defined as the number of situations n (including the true situation) with which i is consistent divided by the total number of possible situations (s^l for a system with s predicates),

$$f(i) = n(i)/s^l, \quad (2.8)$$

where i is a true sentence, and a “situation” is nothing more than a complete state description as was defined for TWSI. Note the lack of symmetry between (2.7) and (2.8); this is one argument against TSSI.

Using these degrees, the *degree of informativeness* is defined as

$$\iota(i) = 1 - f^2(i). \quad (2.9)$$

Continuing the previous example, the tautology is consistent with all situations, and thus has a degree of vacuity $f(i) = 2^6/2^6 = 1$ and a degree of informativeness of $\iota(i) = 1 - 1^2 = 0$. Meanwhile the false statement j has degree of inaccuracy $f(j) = -1/6$ and a degree of informativeness $\iota(j) = 1 - (-1/6)^2 \approx 0.972$, and we see that $\iota(\cdot)$ is consistent with intuition. The relationship between degrees of vacuity and inaccuracy and degree of informativeness is given in Figure 2.2.

The *amount of vacuous information* is then defined as the integral of this curve from 0 to the degree of vacuity; the maximum amount of vacuous information is then simply the integral from 0 to 1, yielding $2/3$. Defining this maximum as α and the amount of vacuous information carried by i as $\beta(i)$, the amount of strongly semantic information carried by i is

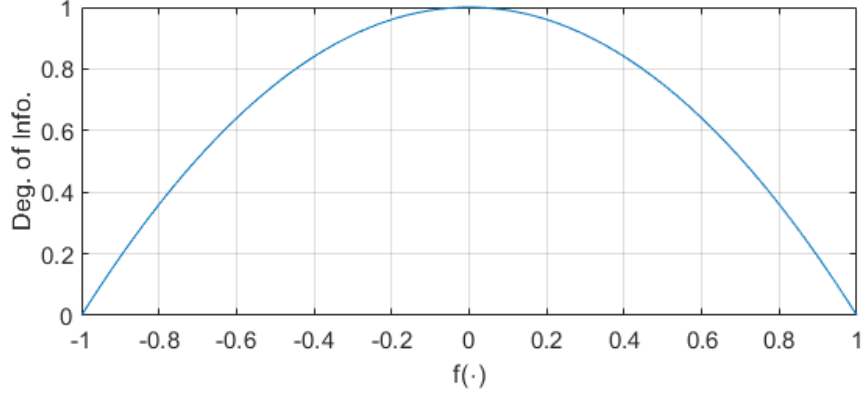


Figure 2.2: Relationship between degrees of inaccuracy, vacuity, and informativeness.

defined as

$$i^*(i) = \alpha - \beta(i) \quad (2.10)$$

It is finally argued that $i^*(i)$ provides a solution to the BCP, in that it shows that “semantic information about a situation presents an actual possibility that is inconsistent with at least one but not all other possibilities. A contradiction is not information-rich because it is not a possibility.” It is stated that in TSSI, a contradiction is simply a limit instance of “uninformation,” or lack of both positive and negative misinformation.

2.3.3 Semantic Information with Truthlikeness

TSSI is a step in the right direction, but still has some shortcomings with regards to quantifying semantic information. First, the degrees of vacuity and inaccuracy are inherently asymmetric; the former is a measure depending on the model, and the second is a measure based solely on the sentence at hand. In addition, it is unclear how to quantify vacuity and inaccuracy for more complex sentences beyond simple conjunctions of atomic statements.

In [14], Simon D’Alfonso builds on the foundations laid by TWSI and TSSI by expanding on Floridi’s idea of using “truthlikeness” for quantifying semantic information. Two existing approaches to quantifying truthlikeness are offered. The first, termed the Tichy-Oddie approach, computes truthlikeness as the complement of some distance function $\Delta(\cdot)$ from the

statement i and the true statement t [14]:

$$\text{Tr}(i) = 1 - \Delta(i, t) \quad (2.11)$$

where $\Delta(\cdot)$ takes values in $[0, 1]$. We can see that (2.10) is a particular case of (2.11) with $\Delta(i, t) = f^2(i)$. Let $\mathcal{W}_i$ denote the set of states in which statement i is true. Furthermore, define $\delta_i^{(t)}$ as the number of differing atomic statements in state w_i and the true state w_t , i.e., the number of false atomic statements in w_i . This value is weighted by the inverse of the number of propositions in the universe. The Tichy-Oddie approach [14] specifies the distance function as

$$\Delta(i, t) = \frac{\sum_{w_i \in \mathcal{W}_i} \delta_i^{(t)}}{|\mathcal{W}_i|} \quad (2.12)$$

Continuing with the example in the previous subsection, i agrees with all state descriptions, so $|\mathcal{W}_i| = 64$. After computing the sum in the numerator we have $\Delta(i, t) = 32/64 = 1/2$ regardless of the true state. Thus, a statement that is always true provides little information no matter the true state of the universe. For the second involving sentence j , we obtain $\Delta(j, t) = 1/6$ and $\text{Tr}(j) = 5/6$, again matching intuition that the false j carries more information than the true i .

In general, the function $\Delta(\cdot)$ can be any distance measure between the state specified by the sentence i and that of the true statement t . The Niiniluoto approach [14] makes use of a different distance metric, namely the min-sum measure

$$\Delta_{\text{ms}}^{\gamma\lambda}(i, t) = \gamma\Delta_{\text{min}}(i, t) + \lambda\Delta_{\text{sum}}(i, t) \quad (2.13)$$

where

$$\Delta_{\text{min}}(i, t) = \min_{w_i \in \mathcal{W}_i} \Delta(w_i, w_t) \quad (2.14)$$

$$\Delta_{\text{sum}}(i, t) = \frac{\sum_{w_i \in \mathcal{W}_i} \Delta(w_i, w_t)}{\sum_{w_b \in \mathbf{B}} \Delta(w_b, w_t)} \quad (2.15)$$

and $\mathbf{B}$ is the set of all states in the logical space. The state distance is defined as

$$\Delta(w_i, w_t) = \frac{\delta_i^{(t)}}{n} \quad (2.16)$$

where n is some atomic weight. Niiniluoto shows that the min-sum measure of (2.13) satisfies certain adequacy conditions.

Finally, D’Alfonso proposes a novel measure, termed the *value aggregate* measure, which is claimed to lie in between the Tichy/Oddie and Niiniluoto approaches. First, each state is assigned a value, with

$$\text{val}(w) = \frac{t^{(w)}}{n2^n} \quad (2.17)$$

where $t^{(w)}$ is the number of true atoms in state w , and n is the number of propositional variables in the logical space. The following algorithm is used to calculate information yield for a statement i :

1. Determine $\mathcal{W}_i$.
2. Place members of $\mathcal{W}_i$ into an array X_1 and order from lowest to highest value.
3. Create empty array X_2 of length 2^n . Fill the first $|\mathcal{W}_i|$ elements with the array X_1 .
Use the last (highest value) element of X_1 to fill the remain spaces of X_2 .
4. Sum the values of X_2 to get the information measure.

It is claimed without proof that the value aggregate measure satisfies many of the adequacy conditions listed by Niiniluoto. A summary of the measures discussed in this and the previous sections is provided in Table 2.3.

2.3.4 Extensions of the Theory of Weakly Semantic Information

These approaches to quantifying semantic information, namely TWSI, TSSI, and truthlikeness, all provide the groundwork for a theory of semantic information, which could enable SemCom. In [54], TWSI is used as a foundation for a general, abstract SemCom model. First a semantic information source is defined as a tuple (W_s, K_s, I_s, M_s) , where

Table 2.3: Summary of classical semantic information measures.

Method	Measure	Benefits	Drawbacks
TWSI [12]	$\text{cont}(i)$	• Simple measure satisfying (2.3)-(2.5) • Only dependent on logical probabilities	• Does not satisfy independent additivity • Trouble with contradictions (BCP)
	$\inf(i) = \log_2 \frac{1}{1-\text{cont}(i)}$	• Satisfies (2.3)-(2.5), independent additivity • Only dependent on logical probabilities	• Does not satisfy triangle inequality • Trouble with contradictions (BCP)
TSSI [13]	$f(i) = -e(i)/l(i)$ (i false), $f(i) = n(i)/s^l$ (i true)	• Avoids BCP by considering truth • Satisfies intuition that true and false statements carry information	• Asymmetric: different measures for true and false statements • Unclear how to define for more complex statements
Truth-likeness [14]	$\text{Tr}(i) = 1 - \frac{\sum_{w_i \in \mathcal{W}_i} \delta_i^{(t)}}{ \mathcal{W}_i }$	• Avoids BCP by considering truth • Provides flexibility through adjustment of atom weights	• Addition of false statement can increase information yield
	$\text{Tr}(i) = 1 - \gamma \Delta_{\min}(i, t) + \lambda \Delta_{\text{sum}}(i, t)$	• Avoids BCP by considering truth • Satisfies several adequacy conditions	• May not be applicable in certain scenarios

- W_s is the world model (potentially observable by the source),
- K_s is the background knowledge of the source,
- I_s is the inference procedure of the source,
- M_s is the message generator used by the source.

The semantic receiver is defined by the analogous tuple (W_r, K_r, I_r, M_r) . The source model is further specified by assuming that the world model W_s is a set of interpretations with probability distributions μ ; for the familiar example of propositional logic, an interpretation would be a set of positive propositions. The inference procedure I_s is then defined as a satisfiability reasoner for the propositional logic, and the message generator M_s employs a fixed coding strategy. The model entropy is then given by

$$H(W) = - \sum_{w \in W} \mu(w) \log_2 \mu(w) \quad (2.18)$$

Letting $m(x)$ denote the logical probability of a message x within this model, the *semantic*

entropy of x is defined as

$$H_s(x) = -\log_2(m(x)). \quad (2.19)$$

Note that this is equivalent to the TWSI measure given by (2.6) with $m(x) = 1 - \text{cont}(x)$. When the knowledge base K_s is included in this formulation, the set of possible worlds is restricted to a set compatible with K_s . This brings about the notion of *conditional semantic entropy*, where $m(x|K)$ now denotes the logical probability of x conditioned upon the background knowledge, and we have

$$H_s(x|K) = -\log_2(m(x|K)). \quad (2.20)$$

Letting X be a finite set of allowed messages with probability distribution $P(X)$, we know the classic Shannon entropy of X as

$$H(X) = -\sum_{x \in X} P(x) \log_2 P(x). \quad (2.21)$$

The following theorem relates the model (semantic) entropy (2.18) to the message (syntactic) entropy (2.21):

Theorem 1. $H(X) = H(W) + H(X|W) - H(W|X)$.

Proof. See [54]. ■

The main implication of Theorem 1 is a formal method of quantifying semantic uncertainty that is rooted in the TWSI and relating it to classic Shannon entropy. As entropy is used to quantify information in classic information theory, this gives a way of comparing the syntactic information and the semantic information under the present world model.

The authors in [54] then discuss the idea of using semantics for data compression. Intuitively, the idea is that some messages may be semantically equivalent without being syntactically equivalent. For example, many times we are able to understand the true meaning of text despite some minor errors in spelling or grammar. Hence, a syntactic error does not necessarily induce a semantic error. Based on this principle, the idea is that we can

achieve maximum compression by choosing the smallest semantically equivalent message for communication. Let $\overline{X}$ denote the smallest subset of X such that each $x \in X$ is semantically equivalent to some $\bar{x} \in \overline{X}$.

Theorem 2. *For a semantic source with interface language X , there exists a coding strategy to generate a semantically equivalent interface language X' with message entropy $H(X') \geq H(\overline{X})$. No such X' exists with message entropy $H(X) < H(\overline{X})$.*

Proof. See [54]. ■

Theorem 2 provides bounds on the maximum achievable data compression give a model as described above. Note that Theorem 2 is analogous to the classical source coding theorem, in that it gives existence of such a code but no insight into how to design the coding strategy.

The final major result of [54] is the so-called semantic channel coding theorem. Some notations used include:

- $I(X; Y) = H(X) - H(X|Y)$ is the traditional information-theoretic *mutual information* between X and Y
- $\overline{H_s(Y)} = -\sum_y p(y)H_s(y)$ is the average logical information of received messages

Theorem 3. *For every discrete memoryless channel, the channel capacity*

$$C_s = \sup_{P(X|W)} \{I(X; Y) - H(W|X) + \overline{H_s(Y)}\} \quad (2.22)$$

has the following property: For any $\epsilon > 0$ and $R < C_s$, there is a block coding strategy such that the maximal probability of semantic error is less than ϵ .

Proof. See [54]. ■

Similar to Theorem 2, Theorem 3 provides a result that parallels the classic channel coding theorem of information theory. This result gives a bound on the maximum amount of information that can be transmitted for some arbitrary probability of semantic error.

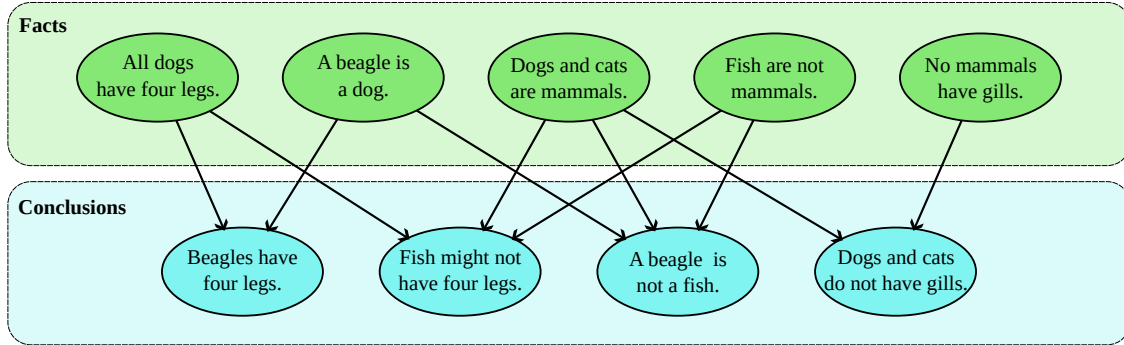


Figure 2.3: Bipartite fact-conclusion graph model of a knowledge base.

In [55], the general model and results of [54] are extended and practical semantic compression algorithms are given based on graph theoretic results. However, basic definitions, such as the semantic source and receiver, are different from those in [54], making it difficult to relate this work to previous results. Perhaps the most useful contribution of [55] is a discussion of some practical techniques for semantic compression.

The first suggested idea is to allow *non-uniquely decodable* codes. This fits the case in which a certain syntactic message represents two semantically equivalent states, i.e., message x could be decoded as either state a or b . Another idea is to extend the concept of erasure channel codes, such that some bits are intentionally “erased.” With this approach, only partial information may be recovered at the receiver, with some intentional semantic ambiguity.

A practical algorithm is then suggested for a system in which the source and the receiver share a knowledge base that is defined as a bipartite fact-conclusion graph, see Figure 2.3. The problem studied is that in which the source wants to convey a set of conclusions to the receiver in as few symbols as possible (where both facts and conclusions can be transmitted and are equally expensive). This problem reduces to computing the *minimum-vertex cover* which is solvable in polynomial time for bipartite graphs. This simple example represents a situation in which semantic compression is computationally feasible, given the assumption on background knowledge.

2.3.5 Summary

The classical semantic information approach attempts to mirror the path followed in the monumental development of information theory to define and quantify semantic information. TWSI attempts to achieve this through the use of logical probabilities rather than statistical probabilities. However, this theory contains a paradox in which a contradiction carries maximal semantic information. As a remedy, TSSI introduces the use of truth values to quantify semantic information, based on the idea that both false and true statements can carry varying degrees of information. This idea has been extended through the introduction of various truthlikeness measures for quantifying semantic information. Recently, some have attempted to extend the original TWSI, relating semantic entropy to traditional information-theoretic entropy and providing analogous source-coding and channel-coding theorems. All of these approaches seek to quantify semantic information of a sentence through the use of logical probabilities and truth values.

With regards to the motivating problem described in Section 1.1, namely, the trend of exponentially increasing global data traffic, the works described in this section provide no quantitative results addressing this issue. This is due to the fact that these works are more concerned with the development of semantic information theory, rather than the subsequent use of this theory for SemCom. That is not to say that the presumed benefits are nonexistent, however, and future work in this area should seek to confirm this potential.

2.3.6 Challenges and Opportunities

Given the ability to quantify semantic information, development of optimal coding techniques (source and channel) should follow. However, as can be seen from the previous discussion, this quantification is no trivial task. A definition of semantic information itself is elusive, and attempts to quantify it become mired with paradoxes and counter-intuitive results. However, should such a complete theory exist, the benefits that would follow are clear, thus presenting a major opportunity.

Previous work seems to point to the idea of truthlikeness as the best approach to quan-

tifying semantic information. The main challenge with this approach is the need for a large knowledge base. Simple examples are given using propositional logic models consisting a few objects and propositions, but the size of these models explodes for more realistic and practical systems. Furthermore, these models must be predefined and known at both the source and receiver, which introduces further complications. Another challenge is selection of the correct information measure. As we have seen, there is no single measure for semantic information; the available measures all come with their respective strengths and shortcomings.

A clear opportunity lies in the further development of the theory of semantic information. While significant progress has been made on the subject, a unifying and ubiquitous theory does not yet exist. Another promising opportunity is the incorporation of existing semantic information measures into practical systems. In particular, *neurosymbolic AI* has become a field of great interest as of late [56]. By combining the strengths of symbolic logic and DL, neurosymbolic AI could enable powerful learning systems that are capable of logical reasoning. By applying neurosymbolic AI to the problem of SemCom, perhaps the logical model at the source and receiver could be learned from data. Then, the DL architecture could be used to efficiently facilitate SemCom. As all of the previous work on semantic information quantification relies to some degree on propositional logic, the application of neurosymbolic AI seems to be a perfect fit.

2.4 Knowledge Graph-Based Semantic Communication

SemCom necessarily requires a knowledge base at both the transmitter and receiver. The engineering of SemCom thus requires some form of knowledge representation to encapsulate the knowledge at the transmitter and receiver. One prevalent method of representing knowledge is through the use of a knowledge graph (KG). Although a particular KG may be defined in different ways, in general a KG can be said to use a graph structure to model a given knowledge base. For example, nodes of the graph could represent objects, while edges represent relations between the objects. By referencing the KG, a system can then

Table 2.4: Summary of works in KG-based SemCom

Knowledge Representation	Delgado-Frias & Moore [45]	“Semantic network” as a KG
	Mertoguno & Lin [46]	Evolution of a distributed knowledge base
	Swartout <i>et al.</i> [57]	Using ontologies to share knowledge
Semantic Similarity	Rada <i>et al.</i> [21]	Link-based similarity measure on KGs
	Resnik [22]	Node-based similarity measure on KGs
	Jiang & Conrath [23]	Similarity measure considering both links and nodes
	Sathya & Uthayan [58]	Measure of quality for an entire ontology
Semantic Sensor Web	Sheth <i>et al.</i> [59]	Semantic sensor web
	Gyrard <i>et al.</i> [60]	Semantic sensor web for machine-to-machine communication
	Chun <i>et al.</i> [61]	IoT directory for the semantic sensor web
	Bhajantri & Pundalik [62]	Data-processing for the semantic sensor web
	Schachinger & Kastner [63]	Semantic interface for building automation
	Lakka <i>et al.</i> [64]	Interoperability of semantic systems
Semantic Communication	Jeong <i>et al.</i> [65]	Semantic error correction of spoken queries
	Zhang <i>et al.</i> [66]	Model for semantic NLP
	Li [2]	Text analysis and character recognition with a KG and ML
	Wang <i>et al.</i> [67]	Hybrid KG/ML model for explanation of recommendations
	Aumayr <i>et al.</i> [68]	Recommendation system for wireless network operation
	He <i>et al.</i> [69]	Model for general goal-based communication
	Güler <i>et al.</i> [70]	Optimal SemCom with game theory
Working with Knowledge Graphs	Wei <i>et al.</i> [71]	Reasoning over large KGs
	Zheng <i>et al.</i> [72]	Embedding of large-scale KGs
	Zhu <i>et al.</i> [73]	Fusion of KGs

perform communication that is semantically-aware. The works discussed in this section are summarized in Table 2.4.

2.4.1 Knowledge Representation with Knowledge Graphs

One of the earliest works making use of this idea was published in 1989, which proposes a so-called “semantic network architecture” for AI processing [45]. Arguing that knowledge representation and manipulation is required for AI, a multi-processor architecture is proposed. This semantic network takes the form of a KG, in which nodes are defined as *concepts* and edges are defined as *relationships* between concepts. Figure 2.4 gives a simple example of a KG. The architecture is composed of a grid of processing elements which compute the corresponding links and nodes associated with the semantic network. By build-

ing the architecture in such a manner, it is argued that the system can perform intelligent actions, such as labeling the scene of a given image (computer vision).

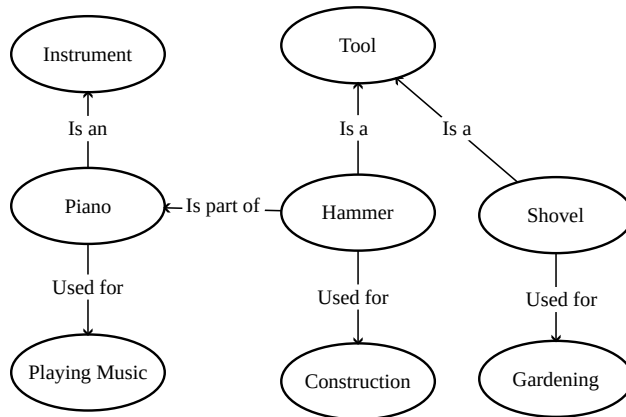


Figure 2.4: An example KG.

Other early studies into knowledge representation with KGs include [46] and [57], published in 1996 and 1997, respectively. In [46], the problem of a *distributed* knowledge base is studied using KGs. Specifically, the challenges of evolving the distributed knowledge and controlling this evolution are considered. This distributed knowledge base is framed as a multi-agent system in which some communication exists between the agents. Within the system, each agent possesses a simple KG and agents collaborate to perform global inferencing. Participating agents receive distributed rewards after correct inferences, resembling a multi-agent RL scheme. A crossover operator is adopted from genetic algorithms to facilitate knowledge evolution. Distributed knowledge bases are also the focus of [57], which uses the idea of a large-scale shared ontology. Here, an ontology is defined as “a hierarchically structured set of terms for describing a domain,” from which a knowledge base can be constructed. The key idea is that if two knowledge bases are formed from the same ontology, knowledge can be easily shared between the two. An analogy would be two people from the same culture; the shared cultural norms and ideas would serve as the so-called ontology, and though their knowledge bases will not be the exact same, intuitively it will be easier for them to communicate given their shared background. A set of desiderata for ontologies is given, essentially proposing that ontologies should represent large-scale, living documents from which we can define smaller, application-specific knowledge bases. This idea is the

basic premise behind the Web Ontology Language standards that shape the semantic web [74].

2.4.2 Semantic Similarity Measures on Knowledge Graphs

As was seen in the previous section, an important idea for SemCom is the idea of semantic similarity. In the sense of classical semantic information, logical probabilities and truthlikeness were used to define this similarity. How should one quantify semantic similarity given a KG representation? This is a question that some began to address around the time KGs emerged as a way of representing knowledge. In [21], a metric of semantic similarity, or distance, is proposed and simply termed Distance. The authors assume a KG that consists of an *is-a* hierarchy, see Figure 2.5 below. This Distance metric is defined for *sets* of concepts

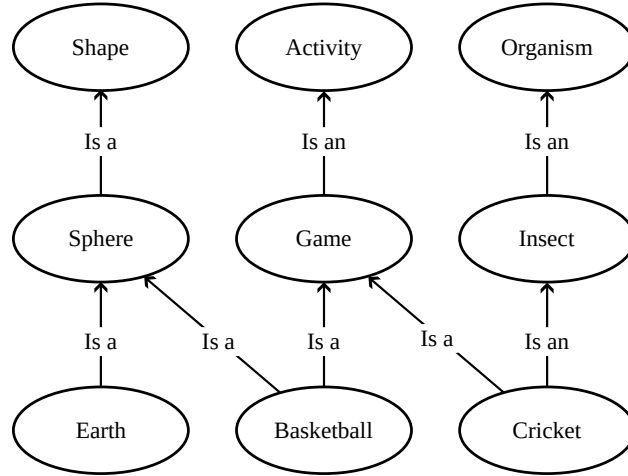


Figure 2.5: Example of an *is-a* KG, where all edges represent an *is-a* relationship.

and is dependent upon the path lengths between nodes in a KG; formally, it is defined as the average minimum path length over all pairwise combinations of nodes between two sets of nodes, i.e., for sets of nodes $\mathcal{X}$ and $\mathcal{Y}$,

$$\text{Distance}(\mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{X}||\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} d(x, y), \quad (2.23)$$

where $d(x, y)$ is the shortest path between x and y . For example, consider the distance between the sets $\{\text{Sphere}, \text{Earth}\}$ and $\{\text{Basketball}\}$ in the KG shown in Figure 2.5. The

shortest path between Sphere and Basketball is 1, while the shortest path between Earth and Basketball is 2, yielding a total distance of $3/2$. Repeating for the sets $\{\text{Sphere, Earth}\}$ and $\{\text{Cricket}\}$, we obtain a distance of $7/2$, and thus we can conclude that the concepts $\{\text{Sphere, Earth}\}$ are more similar to the concept $\{\text{Basketball}\}$ than $\{\text{Cricket}\}$. It is shown through experimental results that (1) Distance can simulate human assessments of conceptual distance and (2) Distance can evaluate some cognitive aspects of semantic networks. However, it is also found that Distance is less applicable to nonhierarchical KGs.

An alternate measure is presented in [22], again for an is-a taxonomy KG. First, it is noted that link-based measures, such as (2.23), suffer from the fact that links in the taxonomy are assumed to represent uniform distances, while in reality some linked concepts may be “closer” than others. Indeed, for the average person in the United States, $\{\text{Basketball}\}$ probably shares a stronger intuitive link to $\{\text{Game}\}$ than $\{\text{Cricket}\}$ does, while this association may differ elsewhere. Therefore, a *node-based* measure is proposed based on the notion of information content. First, the KG is augmented with a function $p : \mathcal{C} \rightarrow [0, 1]$ where $\mathcal{C}$ is the set of all nodes in the graph. $p(c)$ can be thought of as the *probability* of encountering a concept c ; thus, concepts higher in the taxonomy will have greater probability. Then we can define the semantic similarity between two concepts as

$$\text{sim}(c_1, c_2) = \max_{c \in S(c_1, c_2)} -\log p(c) \quad (2.24)$$

where $S(c_1, c_2)$ is the set of concepts that subsume *both* c_1 and c_2 . Intuitively, this measure computes the log-inverse probability of the *most-specific* node (farthest “down” in the taxonomy) which branches to both concepts; therefore, the farther down this node, the smaller it’s probability, and the greater the similarity measure. For example, in Figure 2.5, this would imply that $\{\text{Cricket}\}$ is more similar to $\{\text{Basketball}\}$ than to $\{\text{Earth}\}$.

Combining (2.23) and (2.24), The authors of [23] develop a new measure for the is-a taxonomy KG that provides an even higher correlation with human similarity judgment benchmarks. It involves a link-based calculation which takes into consideration node-based

Table 2.5: Summary of KG-based semantic similarity measures.

Method	Measure	Benefits	Drawbacks
Link-Based [21]	$\text{Distance}(\mathcal{X}, \mathcal{Y}) = \frac{1}{ \mathcal{X} \mathcal{Y} } \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} d(x, y)$	• Simple to compute • Intuitive	• Each link assumed equi-distant • No node-based information
Node-Based [22]	$\text{sim}(c_1, c_2) = \max_{c \in S(c_1, c_2)} -\log p(c)$	• Links not necessarily equi-distant • Interpretable, based on probabilities	• Requires additional function p
Node/ Link-Based [23]	$\text{Dist}(c_1, c_2) = \min_{\text{path}(c_1, c_2)} \sum_c \text{wt}(c, p_c)$	• Incorporates both notions • Achieves highest human correlation	• Requires additional function p • Complicated to compute

edge weights. First, information content is defined in the same way as in [22]:

$$\text{IC}(c) = -\log p(c). \quad (2.25)$$

Then it is argued that the *strength* of a child link is dependent on the information content of the parent node,

$$\text{LS}(c, p_c) = \text{IC}(c) - \text{IC}(p_c), \quad (2.26)$$

where p_c is the parent node of c . This link strength, among other factors, is used to compute an overall edge weight $\text{wt}(p_c, c)$ between the child c and parent p_c . Finally, the semantic similarity, denoted as Dist , is defined as summation of edge weights along the shortest path linking two nodes

$$\text{Dist}(c_1, c_2) = \min_{\text{path}(c_1, c_2)} \sum_{c \in \{\text{path}(c_1, c_2) - \text{LSuper}(c_1, c_2)\}} \text{wt}(c, p_c) \quad (2.27)$$

where $\text{LSuper}(c_1, c_2)$ is the lowest super-ordinate of c_1 and c_2 . Note that in [23], $\text{Dist}(w_1, w_2)$ is used, where w_1 and w_2 are introduced to address the scenario when one node belongs to multiple inheritances. An evaluation of this metric shows a correlation value of 0.828 with human similarity judgments, higher than the node-based and link-based measures alone. This implies that considering link strength within a KG can lead to much more reasonable similarity judgments. A summary of the KG-based semantic similarity metrics described here are given in Table 2.5.

While (2.23), (2.24), and (2.27) all give ways to measure semantic similarity within a KG, the authors of [58] have recently proposed a semantic metric to assess the quality of an entire ontology. It is argued that most existing metrics for ontology assessment consider only structural properties, and ignore the semantics of the ontology. The proposed metric is termed the “relationship deviation metric” and is determined by the number of breadthwise and depthwise relationships in the KG. It is shown that the proposed metric captures the quality of some example ontologies.

2.4.3 The Semantic Sensor Web

One prevalent use of KGs for semantics is the so-called semantic sensor web (SSW) proposed by the authors of [59], which uses ideas from the semantic web. Namely, metadata is captured along with the desired data from each sensor, such that better sense can be made of the observations (similar to how metadata can be used to determine what is contained within a webpage). For example, providing the total lifetime of a sensor may lead to more informed decision-making; a sensor that has been operating well past its expected lifetime may be more likely to produce faulty measurements. With regards to communication, this metadata can provide the *context* necessary for semantically efficient communication within the network. The suggested core set of attributes in [59], as adopted from the RDFa language [75], are

- *about*: a triple that specifies the resource metadata is about
- *rel* and *rev*: specify a relationship or reverse-relationship with another resource
- *href*, *src*, and *resource*: specifies the partner resource
- *property*: specifies a property for the content of an element
- *instanceof*: optional, specifies the RDF type of the object

Furthermore, [59] advocates for the use of ontologies along the three types of semantics associated with sensor data: spatial, temporal, and thematic. Once these ontologies have

been defined, rule-based reasoning can be implemented to provide better inferences from sensor observations.

Building on this idea of an SSW, [60] proposes a semantic-based approach to automatically combine, enrich, and reason about machine-machine data to support IoT applications. One key idea is that the *meaning* of new information is pre-defined in an ontology, and therefore the ontology can facilitate the fusing of cross-domain knowledge. Suppose we possess an ontology which stores weather-related knowledge and soil-related knowledge and the relations between the two; by leveraging this shared knowledge, it might be possible to achieve smarter fusion of the data for agricultural decision-making. A concept called “Linked Open Rules” is defined as a means of sharing and reusing semantic rules, and some examples are given that demonstrate the proposed concept, including a weather monitoring application. In [61], an IoT directory, called IoT-DS, is proposed to support semantic description, discovery, and integration of new objects as an alternative approach to building a SSW. One key difference is that IoT-DS distinguishes static and dynamic components, based on whether other attributes vary with time. It is shown that IoT-DS provides a 40% reduction in communication overhead as compared to a naive approach.

Some other recent works that look at the idea of a SSW are [62–64]. In [62], a survey of data processing techniques for SSW is provided, as well as another take on the architecture of a SSW, which is partitioned into physical, semantic, application and controller layers. The authors of [63] consider the problem of a semantic interface for machine-machine communication, this time with the intended application of building automation. Requirements are defined for the building automation problem, from which an interface is developed and an ontology formed. The authors of [63] provide an example of how KG-based semantics can be utilized to engineer a SSW for a specific application. Finally, the authors of [64] study the general problem of semantic interoperability, or the ability to interact and exchange data with shared meaning, between systems. An interoperability mechanism termed SEMIoTICS is proposed, in which an IoT application request is received by a directory which then connects the corresponding sensor and actuator to fulfill the request.

2.4.4 Knowledge Graphs for Semantic Communication

The works discussed in the previous section are similar to the idea of the semantic web, in that they focus more on how to semantically describe objects rather than the task of SemCom itself. One of the first works utilizing a KG for SemCom is [65], which focuses on semantic error correction for spoken query processing. Spoken query processing, or question answering, has become a hot topic as of late (more on this in Section 2.5). The authors of [65] propose the use of two KGs: a *domain* dictionary and an *ontology* dictionary. The first represents application-specific knowledge, while the second contains the pure general knowledge of the world. In the agricultural setting mentioned above, the ontology dictionary might contain general information that does not vary between crops or location, while the domain dictionary might consist of site-specific information. For a spoken query, the semantic recovery stage involves the use of a semantic confusion table based on the domain knowledge to replace semantic errors. Lexical, or syntactic, recovery is then performed based on the corrected semantic phrase. Experiments performed on the domain of in-vehicle telematics show that the technique yields a 37% reduction in term-error-rate as compared to baseline models. This decreased error rate bodes well for traffic reduction in communication systems, as less information will need to be re-transmitted due to errors.

In recent years, some have sought to apply KG-based semantics to text-based communication. The authors of [66] look to apply KGs to the problem of NLP, and proposes an enhanced language representation model termed ERNIE, which is an enhancement of the popular NLP model BERT [76]. The model operates by first recognizing entities in some text, and matching these to entities in a pre-defined KG. The KG representation is then embedded using known algorithms, such as TransE [77], and then used in conjunction with standard text embeddings as the input to an aggregator. DL encoder-decoder techniques are then used to perform common NLP tasks. In [2], a hybrid KG-ML approach is proposed to perform text analysis through character recognition. The model first uses DL to perform character recognition over two bodies of text, then uses semantic measures to quantify how similar the bodies of text are. To quantify the semantic similarity, each word is modeled as

a word in a graph, and corresponding distance metrics are proposed. The general framework is illustrated in Figure 2.6. Through experiments carried out on Dickinson’s Portfolio, it is concluded that the performance of the proposed technique can meet real-time recognition requirements.

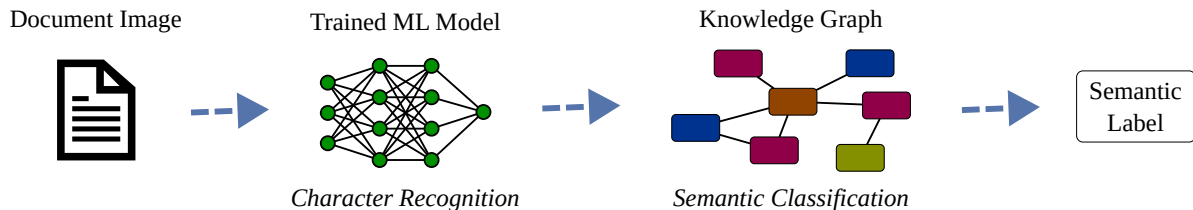


Figure 2.6: Simplified diagram of the framework proposed in [2] for semantic document analysis.

Another application of KGs for SemCom involves recommendation systems, in which the goal is for an automated system to make the best possible recommendations to some user. One way that KGs have been used for this task is to enhance explainability of the recommendations [67]. Explainability can enhance a user’s experience when receiving recommendations, e.g., Amazon suggesting products to a user “based on previous purchases.” In [67], a model termed *Knowledge Path Recurrent Network* is proposed as a hybrid KG-ML model. This model works on a KG which contains objects in the recommendation system and performs reasoning based on the paths in the KG to infer user preference. In the Amazon example, products would be modeled as nodes in the KG, and the software might suggest to a user a product with the shortest path length to that just purchased by the user. A long short-term memory (LSTM) network is adopted to model the sequential dependencies of objects and relations, from which a pooling operation is used to obtain the prediction. By modeling the sequential dependencies, the system can offer an explanation for each prediction. One scenario where a recommendation system is useful is wireless network management, where an automated system can recommend the best course of action to an operator. The growing complexities of these networks can make management difficult for a human operator, while automated methods might not always make the best decisions. Thus, a hybrid approach has some benefits. The authors of [68] focus on gathering context from a wireless network and correlating it with useful information from documents in the network provider’s domain

using KGs. The KG is formed from two types of documents: product troubleshooting manuals and incident/troubleshooting reports by technicians. The first provides well-structured problem solving instructions, while the second provides important links between symptoms and issues. This KG is then used to reason about new problems that arise, and recommend a course of action for the operator. Experiments show a decrease of up to 91% of the documents that are presented to an operator, drastically reducing the amount of information involved in the communication process.

Finally, there has been some work on the more general scenario where two agents strategically communicate to achieve some goal, in which KGs are used to facilitate communication. The authors of [69] study a symmetric collaborative dialogue setting in which two agents communicate to achieve a common goal. Each agent possesses private knowledge. The model, termed Dynamic Knowledge Graph Network, models the dialogue state as a KG which evolves as the conversation advances. The graph contains three types of nodes, namely item, attribute, and entity nodes. These nodes are embedded, and used as an input to a LSTM network. Provided this embedding and an embedded utterance from another agent, the network generates an utterance in response. Experiments show that Dynamic Knowledge Graph Network is able to hold a coherent and strategic conversation with a human, and that the number of entities and attributes uttered to achieve the goal are reduced by 47% and 17%, respectively, when compared to baseline rule-based communication strategies. Goal-oriented communication naturally lends itself to game-theoretic analysis, which is the focus of [70]. In goal-oriented communication, as in game theory, there are two or more agents that seek to achieve some goal, and can employ multiple strategies to reach said goals. Similar to [69], in [70] communication is modeled as taking place between two agents, with the addition of a third agent who could aim to either improve/deteriorate communication performance. The optimal transmission policies are characterized, where optimality is defined as minimizing end-to-end semantic error. This error is derived from the semantic similarity measures proposed in [21–23]. The interaction is modeled as a Bayesian game, where uncertainty is introduced about the characteristics of other agents. It is shown that, in the static scenario, that finding encoding/decoding strategies to minimize average semantic error is an NP-hard

problem, and two algorithms are proposed. In addition, it is demonstrated that when the third agent signals its true nature to the communicating agents, a sequential equilibrium is attainable, i.e., when sufficient information is available regarding the intentions of agents involved in the communication, efficient SemCom can be achieved. It is shown that judicious transmission policies can greatly reduce semantic errors.

2.4.5 Working with Knowledge Graphs

In the previous sections, we have seen how KGs can be used to facilitate SemCom. If our goal is to engineer SemCom through the use of KGs, then we must develop effective methods of working with KGs. Reasoning over, i.e., deriving knowledge from, a KG is a challenge that becomes more difficult with increasing scale of the KG. In [71], a reasoning system is proposed for large-scale KGs. This system, termed KGRL, is based on the web ontology language 2 rules logic which was developed for the semantic web. Using the rules defined by this logic, the iterations of the reasoning procedure are reduced based on dependency relations and multiple applications of these rules. Experiments show that KGRL is able to greatly increase reasoning efficiency as compared to state-of-the-art reasoning systems.

Many of the works previously discussed combined KG methods with DL methods, which requires an embedding of the KG into some vector space. This embedding essentially aims to preserve the knowledge represented by the graph in the embedding space. Similar to reasoning, this task becomes difficult at large scales. The authors of [72] study the problem of training KGs at scale, proposing a technique termed DGL-KE to efficiently perform KG embeddings. DGL-KE provides optimized embeddings for three types of hardware configurations: (1) many-core central processing unit (CPU) machines, (2) multi-graphics processing unit (GPU) machines, and (3) a cluster of CPU/GPU machines. For each hardware type, the DGL-KE takes advantage of parallel processing to fully utilize the computing hardware. The allocation of memory resources throughout the process is specifically designed for each hardware type, and mini-batch training is utilized to perform the embedding. Other optimization techniques employed by DGL-KE are graph partitioning, negative sampling, data

access to relation embeddings, and applying gradients to global embeddings. Experiments for hardware types (1) and (2) demonstrate improved efficiency compared to other methods.

Another interesting problem is that of KG fusion. For example, say that when a new agent joins the network, we wish for it’s knowledge to be merged with the overall knowledge of the network. If both knowledge bases are represented by KGs, how should we fuse them together? One way is by *instance matching*, which establishes a semantic link between instances in KGs. The authors of [73] propose a method called Follow-the-Regular-Leader Instance Matching (FTRLIM), which is able to match instances between large-scale KGs with approximately linear time complexity. The FTRLIM framework is based on an blocking algorithm called MultiObj, which divides instances into blocks and is also developed in [73]. Through various experiments, FTRLIM is shown to perform effective and scalable KG fusion.

For more information on the field of KGs, we direct the interested reader to [47], which provides a recent and comprehensive survey covering (1) KG representation learning, (2) knowledge acquisition and completion, (3) temporal KGs, and (4) knowledge-aware applications (such as SemCom).

2.4.6 Summary

Following the large amount of research that has been dedicated toward the semantic web, there have been many works which seeks to utilize KGs for SemCom. This idea stems from the fact that some kind of knowledge representation is required for SemCom, and a common way of representing knowledge is with KGs. Furthermore, some have proposed similarity measures for concepts within a KG, which can be used to quantify semantic similarity. This form of knowledge representation has been proposed for use in the so-called semantic sensor web, which extends the semantic web to physical sensor networks. Various works have attempted to leverage KGs for different SemCom scenarios, including recommendation systems and general goal-oriented communication. Methods such as hybrid ML-KG systems, game theoretic techniques, and others have been applied to achieve this communication. Finally, some have studied the specific problem of working with the KG itself, which will

be important when implementing these systems at scale. The key idea behind KG-based SemCom is that knowledge, and therefore meaning, can be captured by a KG and utilized by the semantic system.

Some of the works mentioned above have provided quantitative results demonstrating improved performance with regards to communication efficiency. As mentioned, results in [61] indicate a reduction of around 40% in overall network traffic. Results from [65] show decreased error rates using semantic techniques, which in turn has implications for reduced communication traffic. Experiments in [69] show that goal-oriented SemCom can reduce the amount of entities and attributes communicated by 47% and 17% respectively. In a more specific example, results from [68] demonstrate that SemCom can reduce the amount of total information conveyed in a recommendation system by up to 91%. These results, stemming from diverse examples and use-cases, show that KG-based SemCom can address the issue of increasing data demands with more efficient communication.

2.4.7 Challenges and Opportunities

KGs are a popular way of representing knowledge in a system, which can then be used to facilitate SemCom. However, this approach does not come without its challenges. First of all, as knowledge in the system grows, the KG can become massive. As was discussed in the previous subsection, working with KGs becomes difficult as they grow larger. Scalability is a challenge that must be addressed for efficient SemCom. This difficulty is amplified in systems with stringent communication requirements, such as those requiring real-time operation. Another challenge is that a KG must be predefined with some prior knowledge of the system, reducing the ease of deployment. Building these graphs can be time-consuming and therefore costly.

These challenges present opportunities as well. Further development of scalable methods, such as those in [71–73], will be required for the practical use of large (and thus more expressive) KGs. Furthermore, techniques such as transfer learning present a promising approach toward reusing existing KGs for new applications, such that a KG does not need

to be built from scratch for each deployment. Another exciting field is of ML and DL on graphs. These techniques can be employed to learn optimal embeddings and relations from existing data, and bring with them all of the benefits that have been achieved with DL in other domains.

2.5 Machine Learning-Based Semantic Communication

The field of ML has seen an explosion of activity in recent years. At its core, ML seeks to learn from data in order to better perform some task. The availability of massive amounts of data and advanced algorithms have enabled the practical implementation of ML in many domains, including NLP [78], computer vision [79], and others. Over the past few years, some have sought to utilize the power and flexibility of ML to develop SemCom systems. The general idea behind these approaches is to “learn” the semantics of the problem. Just as in image processing, where the key features for classifying an image may be hidden to us, the semantic aspects of communication may be unknown. Through the use of ML methods, these latent semantic features can be learned through training and added to the system’s knowledge base automatically. Thus, by utilizing learning methods, these systems address one of the key challenges inherent to KG-based SemCom, namely the requirement of a predefined knowledge base. In this section, we will take a closer look at some of these works to illustrate the ML-based approach to SemCom. The works discussed in this section are summarized in Table 2.6.

2.5.1 Deep Learning Methods

DL is a subset of ML which utilizes DNNs to perform prediction and decision-making tasks [90]. These networks are trained through an iterative update of the network parameters, which is typically accomplished through some gradient-based method. Modern DNNs come in many different forms, such as the classic multi-layer perceptron [91], convolutional neural network (CNN) [92], recurrent neural network (RNN) [93], etc. As with any DL problem, a

Table 2.6: Summary of works in machine learning-based SemCom.

Deep Learning	Lu <i>et al.</i> [80]	Ensuring mutual understanding using a LSTM-RNN
	Hua & Du [3]	GAN for cross-modal retrieval
	Huang <i>et al.</i> [81]	Semantic coding of images with a GAN
	Qiao <i>et al.</i> [82]	CNN, RNN for scene text recognition
	Tong <i>et al.</i> [83]	Audio-based SemCom with CNN and federated learning
	Zhou <i>et al.</i> [84]	Text-based SemCom using a transformer
	Sana & Strinati [85]	End-to-end communication with semantic symbols using a transformer
	Xie <i>et al.</i> [4]	Text-based SemCom with DeepSC
	Weng <i>et al.</i> [51]	Speech-based SemCom with DeepSC-S
	Xie <i>et al.</i> [52]	Visual question answering with MU-DeepSC
	Xie & Qin [50]	SemCom for IoT with L-DeepSC
Reinforcement Learning	Lu <i>et al.</i> [86]	General SemCom with arbitrary similarity function
	Wang <i>et al.</i> [87]	Reinforcement learning over a KG for text communication
	Lotfi <i>et al.</i> [88]	Collaborative deep reinforcement learning with heterogenous agents
	Yun <i>et al.</i> [89]	Reinforcement learning for air-to-ground SemCom

key consideration is how to choose the *loss function*, which will determine how the parameters are tuned. For more on information on DL, please see [90].

Regarding SemCom, DL can be used in many different ways, often depending on the modality of communication (text, images, speech, etc.). One of the first works employing DL for SemCom provides an illustrative example. In [80], aviation radiotelephony communication is considered, where a statement spoken by one party is repeated by the other to ensure understanding. In this scenario, clear and reliable communication is crucial, as misunderstandings can lead to accidents and crashes. To minimize semantic errors (i.e., misunderstandings between pilots and air traffic control), a LSTM RNN is proposed. The LSTM-RNN takes as its input a pilot-air traffic control sentence pair. To train the network, statement pairs are assigned a similarity R , ranging from $R = 0$ to $R = 1$, with 1 indicating the strongest semantic similarity. Clearly, a strong similarity is desired; we want the pilot and air traffic control to be on the same page. The network is then trained to minimize the *cross-entropy* error:

$$L = - \sum_{n=1}^N R^* \log(R) + (1 - R^*) \log(1 - R) \quad (2.28)$$

where R^* is the true label value and R is the predicted consistency value. After training,

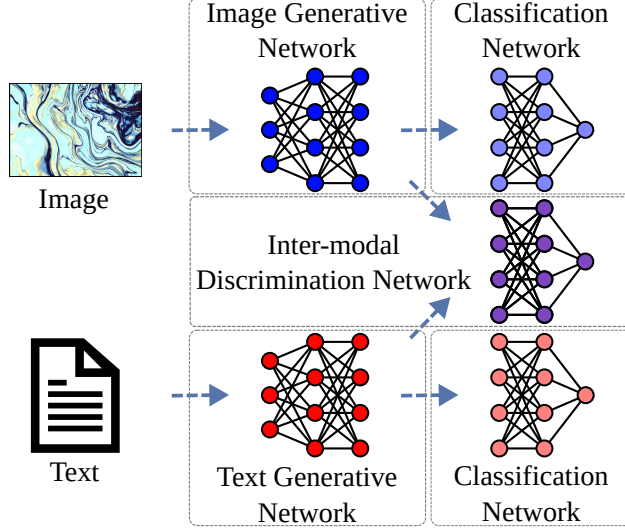


Figure 2.7: Basic framework of the multi-modal GAN approach proposed in [3].

the network can be used to classify new statement pairs; if a pair is found to be semantically inconsistent, a signal can be given to either party to correct the misunderstanding.

Another interesting problem involving SemCom is that of cross-modal retrieval, in which the format of the query is different than that of the information being queried, e.g., a voice request for certain textual document in a database. In [3], a generative adversarial network (GAN) is proposed to address this problem by performing *semantic correlation* on multi-modal data, specifically image and text data. A GAN is a network composed of two networks, namely a generator and a discriminator. The generator learns by attempting to “fool” the discriminator, while the discriminator is optimized to discern between outputs of the generator. The authors of [3] propose a framework in which the generator takes both an image and a body of text as inputs and learns their representations. The goal of the discriminator is to distinguish between the two modalities. The basic framework of this approach is illustrated in Figure 2.7. After training, the generator will have learned the representations of the heterogeneous modalities in a common space, in which data with similar semantics will be close, i.e., the *meaning* common to both the text and the image is captured in this space. The proposed method is shown to outperform both traditional and deep methods with respect to the mean average precision metric.

Another work utilizing a GAN is [81], in which DL is implemented with the goal of

semantic coding of images. The goal of semantic coding is to minimize the bit rate of transmission while preserving the semantic information of the image. Again, the semantics of interest here are unknown, and the goal is to learn them throughout the training process. Here, the generator network is used to learn and restore semantic information which is used as a “base layer” of the image. The generator loss function is formed as a rate-perception-distortion trade-off, including a combination of the VGG loss [94] and LPIPS [95]. Then, Better Portable Graphics residual coding is used to refine the image. The overall network explores different strategies to optimize the rate-perception-distortion tradeoff, and is shown to exhibit similar performance to baselines models while achieving a 2-4 $\times$ reduced bit rate.

Similar to the work in [66], the authors of [82] look to enhance standard encoder-decoder NLP techniques (this time without the use of a KG), specifically for the problem of recognizing some text within an image, or *scene text recognition*. For example, in the case of a self-driving vehicle, it would be beneficial for the vehicle to be able to recognize the text located on street signs within the field of vision, as a human driver would. In their model, termed SEED, semantic information is used in both the encoder module for supervision and the decoder module for initialization. The semantic information is predicted from the image features which are first extracted with a CNN and RNN. To predict this information, a simple fully-connected DNN is trained with a dual cross-entropy and cosine embedding loss function. Then, the image features and the semantic information are both fed to the decoder to perform text recognition. Using ASTER [96] as an exemplar to demonstrate the framework, it is shown that the semantic enhancement improves performance of the model.

A CNN along with federated learning is proposed to facilitate audio-based SemCom in [83]. Federated learning is a branch of DL in which distributed, local models are trained individually and then combined to form a global model. In [83], federated learning is implemented in a system consisting of a single server and many devices. The devices train local models based on data, to reduce communication overhead to the server. Each local model consists of an with a convolutional layer for extracting the semantic information of the audio. Normalized root mean square error is used as a loss function to evaluate the quality of semantic reconstruction, and experiments show that the proposed architecture can achieve

around $100\times$ improved performance (with respect to the mean square error) with around 1/3 the transmitted data of traditional methods.

Yet another type of DNN is the *transformer*, which has seen wide success in the field of NLP [78]. Transformers rely on the idea of *attention*, which provides different weights to different features of the input data, similar to how our brains pay more attention to certain perceptual inputs. In [84], an adaptive universal transformer is implemented for text-based SemCom. The optimization goal is “to minimize the semantic errors while facing different communication situations.” The universal transformer is able to accomplish this by adding a circulation mechanism which can dynamically allocate greater computation time to semantically complex statements. Intuitively, this is similar to how a human might read some text. Passages with a simple meaning are easier to understand and thus can be read faster, while passages with complex meaning demand more thought (equivalently, computation time). Cross-entropy is used as the loss function for training the network, and simulations performed on the standard proceedings of the European Parliament [97] show an improved performance over traditional methods.

The transformer has also been proposed for SemCom in [85]. In this work, an end-to-end architecture is proposed which performs communication with *semantic symbols*, which are used to represent semantics. The SemCom model is defined, including the encoder, decoder, channel and noise. The transformer model is then designed for this model, including both source and channel coding in a joint architecture. This architecture is trained using the cross-entropy loss, and experiments in a NLP setting demonstrate that the proposed system can achieve a similar performance to traditional techniques with a 21% decrease in the number of symbols.

2.5.2 DeepSC and its Variants

One particular DL model for SemCom that has been the subject of various studies was first proposed in [4], where it was given the name DeepSC. DeepSC is also based on the transformer DL model, and the model proposed in [84] takes its inspiration from DeepSC.

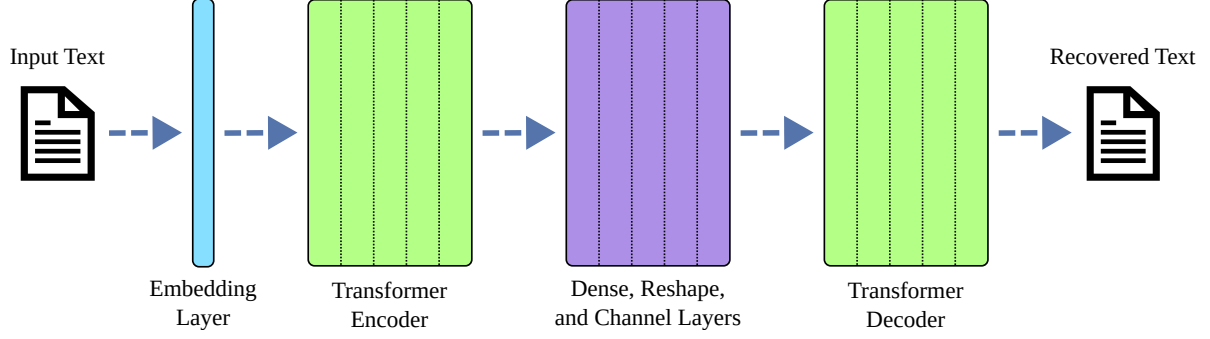


Figure 2.8: Architecture of the end-to-end SemCom system DeepSC proposed in [4]. Each box represents a different section of the network, which are all trained in a joint manner.

Similar to other works, [4] defines the model for the SemCom system, which consists of a transmitter performing both semantic encoding and channel coding, and a receiver performing the inverse operations. The architecture of the proposed network is shown in Figure 2.8. Both transmitter and receiver possess some background knowledge. The goal is stated as simultaneously minimizing semantic errors (measured with the cross-entropy loss function) and transmitted symbols. This is accomplished through an end-to-end transformer network, which uses a self-attention mechanisms for extracting semantic information from text; here, the meaning is captured by this attention mechanism and which input text is emphasized by the model. Various metrics are used to demonstrate the superior performance of DeepSC compared with traditional communication methods.

There have been a few variants inspired by DeepSC, which all aim to facilitate SemCom within different modalities. One example is [51], where the authors propose a SemCom system for speech signals, termed DeepSC-S. It is claimed that the end-to-end communication system “learns and extracts the essential speech information.” This is a very intuitive idea, as it is clear how non-verbal qualities of speech can impact a conversation (tone, volume, etc.). Specifically, DeepSC-S employs an attention-based CNN for speech coding and a CNN for channel coding, and the mean-squared error loss function is used for training. It is confirmed through simulations that DeepSC-S outperforms traditional systems with equivalent bit rates under AWGN, Rayleigh and Rician channels.

Another variation of DeepSC is proposed in [52], where the authors consider a multimodal

communication system and is termed MU-DeepSC. The task-oriented system is implemented for visual question answering, where the transmitter sends an image and a question about that image (text) and the receiver aims to correctly answer the question (e.g., is there a red ball present?). The transmitter consists of two networks; one employs ResNet-101 [98] and a CNN for semantic image encoding, and the other utilizes Bi-LSTM [99] and a dense DNN for semantic text encoding. These encodings are transmitted over the channel, where the receiver then implements a memory, attention, and composition network to generate the answer. The end-to-end network is trained with the cross-entropy loss, and simulation results using the CLEVR dataset [100] demonstrate accuracy gains of up to 80% compared to traditional methods with a 70% reduction in transmitted symbols.

As we’ve seen in the discussion of the semantic sensor web, a prevalent idea is the use of SemCom in IoT applications. In [50], a “lite” version of DeepSC, named L-DeepSC, is proposed for use with IoT networks. In L-DeepSC, the SemCom model is trained and updated in the cloud and distributed to IoT devices. These devices then implement the model to perform semantically aware data collection and text transmission with low complexity. The system first uses a least-squares estimator to obtain channel state information, and then a deep de-noise network to refine the channel state information estimates. The trained model is then compressed through sparsification and quantization and broadcast to the IoT devices. The devices then use this model to perform SemCom with text data, uploading new data to the cloud. It is shown that the proposed system performs competitively with traditional methods, especially in the low signal-to-noise ratio (SNR) domain. Moreover, L-DeepSC reduces the model parameters of the original DeepSC network by around 60%, which translates to reduced communication upon model distribution to IoT devices.

As ML-based SemCom is based on *learning* the semantics of the problem, explicit semantic metrics are not defined. However, the chosen *loss function* affects how the semantics are learned. Table 2.7 provides the loss functions used in each of the discussed DL approaches.

Table 2.7: Summary of DL-based SemCom loss functions

Loss Function	Expression	Description
Cross-entropy [4, 50, 52, 80, 84, 85]	$L = -\sum_{n=1}^N R^* \log(R) + (1 - R^*) \log(1 - R)$	Promotes semantic similarity or probability values R consistent with labels R^*
Cross-modal generative [3]	$L_{gen} = \lambda L_{adv} + L_{cls}, L_{adv} = -\sum_{i=1}^N [\log(1 - D(G_I(x_i))) + \log D(G_T(y_i))],$ $L_{cls} = -\sum_{i=1}^N \log [C(G_I(x_i))^T c_i + \beta C(G_T(y_i))^T c_i]$	Aims to promote similar image embeddings $G_I(x_i)$ and text embeddings $G_T(y_i)$ given semantically similar inputs x_i and y_i
Rate-perception-distortion [81]	$\mathcal{L}_{E,G} = \mathcal{L}_G + \lambda_1 (\mathbb{E}[d(x, G(\hat{w}, s))] + \mathbb{E}[\ G(\hat{w}, s) - x\ _1]) + \lambda_2 (R(\hat{w}) + R(s) + R(r'))$	Simultaneously minimizes the rate, perception and distortion with hyperparameters λ_1 and λ_2
Cross-entropy/cosine embedding [82]	$L = L_{rec} + \lambda[1 - \cos(S, em)]$	Combination of traditional cross-entropy L_{rec} and cosine embedding loss
Normalized root mean squared error [83]	$\mathcal{L}_{NRMSE} = \frac{\sum_{t=1}^T (a_t - \hat{a}_t)^2}{\sum_{t=1}^T a_t^2}$	Seeks to minimize the difference between recovered audio data $\hat{a}_t$ and actual data a_t
Mean squared error [51]	$\mathcal{L}_{MSE} = \frac{1}{W} \sum_{w=1}^W (s_w - \hat{s}_w)^2$	Seeks to minimize the difference between recovered speech data $\hat{s}_w$ and actual data s_w

2.5.3 Reinforcement Learning Methods

Another popular approach to ML is reinforcement learning (RL) [101]. In RL, the model is viewed as an agent in a state space. From its current state, the agent can take some action, for which it is provided a reward. The goal of the agent is to discover the action-taking policy which will maximize long-term rewards from any given state. Rather than use existing data to determine this policy, in RL the model is trained by letting the agent “explore” different policies; the model takes some sequence of actions, and then tunes the parameters based on the rewards received. A simple illustration of this learning process is given in Figure 2.9. Due to the agent/reward set up of RL, it is a method which is well-suited for learning to play different games, such as chess and Go. Indeed, learning models including both RL and DL have been used to create artificial players which outperform human world champions in both games [102].

Recently, some have turned to using RL as a means to achieve SemCom. In [86], a RL solution is proposed to carry out general SemCom. It is argued that the objective functions of many of the ML-based approaches to SemCom demonstrate a “semantic blindness,” and

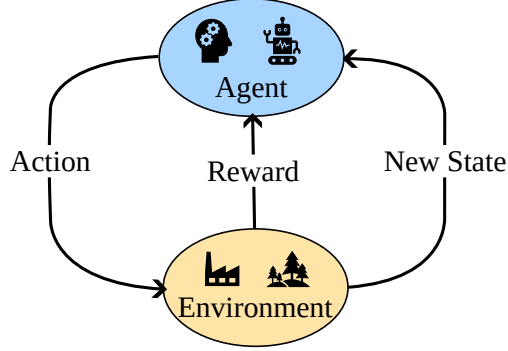


Figure 2.9: Reinforcement learning framework.

are still biased toward bit-level accuracy. The proposed joint source channel coding solution, termed SemanticRL-JSCC, is formulated using a Markov decision process framework. The reward function is based on any *general* semantic similarity function. This method differs from those DL methods discussed, as the semantics of the problem are defined by the chosen similarity function, and thus SemanticRL focuses on *how* to best communicate given some semantics, rather than learning *what* those semantics are. A distinct feature of SemanticRL as opposed to other ML approaches is that this similarity function is not necessarily differentiable. In the training of the model, a *self-critic* approach [103] is taken, resulting in a quicker and simpler solution. Experiments carried out over the European parliament dataset demonstrate superior performance with regard to common metrics, as well as a stable learning trajectory.

The authors of [87] combine a KG representation of semantic information with RL for SemCom using text data. First, a KG is extracted from a body of text, and this KG is treated as the semantic information of that text. Based on this KG representation, two metrics, namely semantic *accuracy* and *completeness* are derived. Combining these two metrics gives the overall metric of semantic similarity. SemCom is then formed as an optimization problem which seeks to maximize the semantic similarity of the text at the transmitter and receiver through resource allocation and information transmission, under a delay constraint. Using an attention-based RL framework to solve this optimization problem, it is shown that the proposed SemCom solution outperforms a traditional RL scheme as well as typical wireless communication techniques.

Collaborative RL is a form of RL in which multiple agents are present in the system, and they collaborate to determine the optimum policy. In [88], collaborative deep RL (DRL) is used to train a group of heterogeneous agents over a wireless cellular network. First, the algorithm selects the best subset of semantically relevant DRL agents for collaboration. This semantic relevance between two agents is based on their policies; if a target agent returns a large average reward under a source agent’s policy, the target is said to be similar to the source. Here, the semantics are captured by the policies of each of the agents; similar meaning is implied by a similar policy. Once the similar subset of agents is obtained, the training loss and wireless bandwidth are jointly minimized to obtain the optimal policies for each agent. Simulations of the proposed technique show improved training performance compared to other collaborative DRL methods and classic DRL. It is also shown that the proposed approach is able to use resources more efficiently, demonstrating better performance with fewer resource blocks than other approaches.

Looking to implement RL-based SemCom for a specific application, the authors of [89] propose a DRL framework for air-to-ground ultra-reliable low latency communication using UAVs. Similar to the work of [88], they propose the use of a multi-agent DRL framework, coined graph attention exchange network, for SemCom. Self-attention is used to determine the attention a UAV gives to other UAVs in the network, and based on this attention, training is performed in a centralized manner. Once the optimal policies have been obtained, the central unit distributes the model to the UAV’s and actions are carried out in a decentralized manner. It is shown that the proposed GAXNet achieves more efficient training than the state-of-the-art centralized training and decentralized execution algorithm QMIX [104], and is better able to avoid collisions between UAVs.

2.5.4 Summary

ML-based SemCom is an approach that has seen a spike of interest with the recent boom in AI technology. In ML-based SemCom, the semantics of the problem are not predefined as in classical and KG-based SemCom, but rather they are learned through data-driven

training. This learning is performed either through DL or RL. Many different DL models have been proposed to facilitate SemCom, include the transformer, GAN, CNN, and others. One notable example is DeepSC, which was originally proposed as a text-based SemCom system using a transformer-based network. Variants of DeepSC have been proposed in recent years which target different modes of communication. Though not as prominent, some RL methods have been proposed to learn semantics as well. One benefit of this approach is the loss function need not be differentiable. In ML-based SemCom, meaning is characterized by the parameters of a model which are learned in a data-driven manner.

Many of the discussed works provide quantitative results demonstrating efficient SemCom. The results of [81] show a 2-4 $\times$ reduction in bit rate, while experiments in [83] indicate large performance gains over traditional systems for around 1/3 the original bit rate. [85] demonstrates a 20% reduction in transmitted symbols to achieve a similar performance as baseline systems. DeepSC and its variants also indicate potential for traffic reduction; DeepSC-S [51] can achieve improved speech performance for similar bit rates, and MU-DeepSC [52] achieves superior accuracy with a 70% reduction in transmitted symbols. Finally, L-DeepSC [50] reduces traffic in another way, by drastically reducing the parameters of a ML model which is to be broadcast to IoT devices. All in all, as in KG-based SemCom, existing works in ML-based SemCom indicate the potential for this approach to address the issues raised at the outset.

2.5.5 Challenges and Opportunities

One of the challenges with ML-based approaches to SemCom was pointed out in [86], which is the difficulty in working with semantic metrics. As most DL techniques use gradient-based methods for optimization, we must necessarily work with differentiable metrics, while many semantic similarity metrics are non-differentiable. Another critical challenge is the black-box nature of many DL models. A challenge within the field of DL as a whole, this quality of deep networks obscures our ability to analyze and evaluate *why* a model does or does not perform well. Finally, the improved performance that DL has enjoyed due to big data is approaching

a limit, where further gains can only be reached through massive training, which imposes a huge computational burden [49].

Opportunities can be found in the solutions to these challenges. First, there is certainly no ubiquitous metric of semantic similarity, and it is likely that many metrics will be application-specific. Development of such metrics is of critical importance to the advancement of SemCom. As each approach to SemCom we’ve seen thus far (classical, KG-based, ML-based) employs its own unique metrics, it is likely that a convergence of these ideas and combination of metrics could yield novel implementations and improved results. Furthermore, development of methods similar to [86] that relax the differentiability requirement of the semantic metric is a promising avenue for further study.

Regarding the lack of interpretability and ever-growing appetite for data associated with deep networks, one promising solution is the development of *model-based DL* techniques [49]. Model-based methods are those that derive some inference rule based on prior knowledge of the problem, while data-driven methods (including DL) rely solely on data to form the inference rule. Model-based methods are typically more interpretable and efficient than data-driven methods, while data-driven methods are more expressive and robust in new situations. Techniques utilizing the strengths of both methods to create a single model are referred to as model-based DL [49], and are another promising method of addressing some of the challenges facing ML today.

2.6 Significance-Based Semantic Communication

One last view on SemCom has been proposed only recently, and involves defining the semantics of information as the *significance* of this information [5]. Recall “The effectiveness problem” defined in Chapter 1: “How effectively does the received meaning affect conduct in the desired way?” Defining the semantics of information as significance of information essentially addresses this problem, as significance is inherently determined by what one is trying to achieve with communication, or the *goal*. As a simple example, if the goal of communication is to control a robot performing remote surgery (a necessarily real-time application),

Table 2.8: Summary of works in significance-based SemCom.

Age of Info.	Uysal <i>et al.</i> [5]	Age of information as a significance-based semantic metric
	Uysal <i>et al.</i> [105]	Practical evaluation of the age of information metric
	Beytur <i>et al.</i> [106]	Age of information performance of UDP and TCP protocols
	Ayan <i>et al.</i> [107]	Comparison of AoI/VoI impact on cellular control system performance
Value of Info.	Uysal <i>et al.</i> [5]	Value of information as a significance-based semantic metric
	Molin <i>et al.</i> [108]	Value-based information management for state estimation
	Ayan <i>et al.</i> [107]	Comparison of AoI/VoI impact on cellular control system performance
Semantic Sampling	Uysal <i>et al.</i> [5]	Semantic sampling as part of a SemCom architecture
	Kountouris <i>et al.</i> [30]	Semantic sampling as part of a SemCom architecture
	Bacinoglu <i>et al.</i> [109]	Semantic sampling for tracking of a stochastic process
	Dommel <i>et al.</i> [110]	Semantic sampling for goal-oriented communication over a shared medium

information that was just obtained will be much more significant than information obtained 10 seconds ago. Based on this general idea, the authors of [5] call for “a redesign of the entire process of information generation, transmission and usage in unison.” In this section, we survey the few recent works that support this idea of significance-based SemCom. The works discussed in this section are summarized in Table 2.8.

In [5], some examples of measures which relate to the significance of information are given. The first of these is information *freshness*, which is determined by the time taken since the information was generated to when it was received. AoI is a measure which captures this idea of freshness, and has been well-studied over the past decade or so. We will discuss AoI further in the following subsection as a prime example of a significance-based semantic measure; the authors of [5] present results indicating energy savings of different age-aware protocols ranging from 10-64%. Another example of a semantic measure is *relevance*. Consider a process that is being sampled; consecutive samples that capture little change in the process are typically of less interest than those for which sudden changes occur. We could say that the latter samples are more relevant than the former. As an extension of relevance of information, a more powerful example is given as the *value* of information (VoI), which is defined as the difference between the benefit of a sample and the cost of its transmission. It is argued in [5] that developing metrics that capture these ideas is critical for achieving SemCom. Those mentioned here are summarized in Table 2.9.

Table 2.9: Summary of significance-based semantic measures.

Measure	Expression	Description
Age of Info.	$\Delta(t) = t - u(t)$	Characterizes the <i>freshness</i> of information generated at time $u(t)$
Value of Info.	$v(x) = b(x) - c(x)$	Difference between the benefit of a sample and cost of its transmission
Relevance of Info.	$r(x) = f(x_n, x_{n-1})$	Quantifies the amount of change in a process since the previous sample

The authors of [5] also present a vision for an end-to-end SemCom architecture. This architecture takes into considerations the elements of freshness, relevance, and value to optimize the entire system. A simplified flowchart of the proposed architecture is provided in Figure 2.10. Specifically, semantic sampling is implemented to relax the assumption that data arrives in an uncontrolled manner, i.e., only significant information is generated in the first place. Semantic channel encoding, multiuser scheduling, channel access and flow control are proposed to increase the efficiency and effectiveness of each of these processes. It is stated that the realization of this architecture will involve an entirely new paradigm shift that is incompatible with previous designs of communication systems. A few specific examples are envisioned, which include SemCom for networked control systems, smart cities, and massive machine type communications/IoT systems.

The authors of [30] also focus on the idea of significance-based SemCom. They define semantics of information as “the significance and usefulness of messages.” A similar argument to that of [5] is given, stating that simple generation and communication of data often leads to reception of stale or irrelevant information at the receiver and wasted resources. This again brings about the need for a *goal-oriented* communication system, one that addresses the “effectiveness problem.”

A new concept presented in [30] is the idea of defining semantics at different *scales*. At the

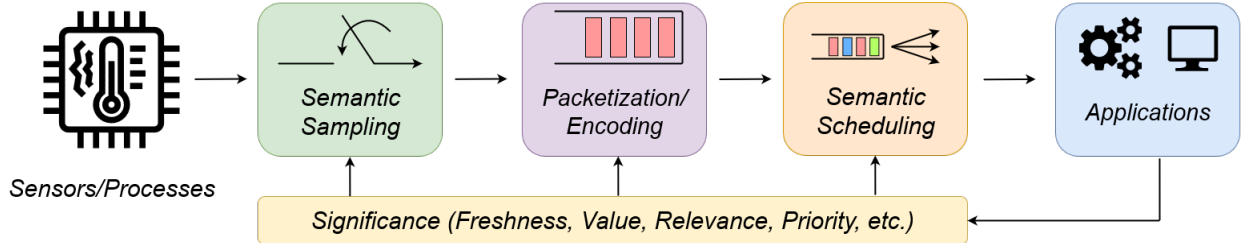


Figure 2.10: Overview of the significance-based semantic architecture proposed in [5].

microscopic scale, specific pieces of information from the source may be of different significance, e.g., that a safety risk is present or not. The *mesoscopic* scale is the intermediate level, which takes into consideration link-level semantics. This includes both innate (objective) measures such as freshness (AoI) and precision, and contextual (subjective) measures such as timeliness and completeness. Finally, the *macroscopic* scale takes system-level semantics into consideration, specifically looking at end-to-end distortions and delays that affect the end goal.

An end-to-end semantic architecture is proposed, which is strikingly similar to that envisioned in [5], including semantic sampling and semantics-aware signal processing blocks. A specific example of an end-to-end communication system is given, involving a remote actuation application. The source monitors the actual state of a robotic arm, while the receiver aims to construct and maintain a digital twin of this robotic arm. Communicating over a wireless erasure channel, it is shown that an end-to-end semantics approach performs much better than other, more semantically-unaware approaches in terms of real-time reconstruction error and cost of actuation error.

2.6.1 Age of Information and Value of Information

Two proposed measures that capture the significance of information are AoI and VoI. In this subsection, we will discuss some of the work that has been done with regards to these two measures to illustrate their use as semantic measures.

The authors of [105] provide a compilation of some recent works examining AoI in practical scenarios and looks at issues such as synchronization, transport layer protocols, congestion, and the use of ML. First, the status *age* of an information flow (characterized as a flow of data packets) is defined as the difference between the current time and the generation time of the most recently received data packet, which is illustrated in Figure 2.11. Other derivative metrics are defined as well, such as *average* AoI and *peak* AoI. Depending on the application at hand, one may desire to minimize either the average or peak AoI to best maximize information freshness.

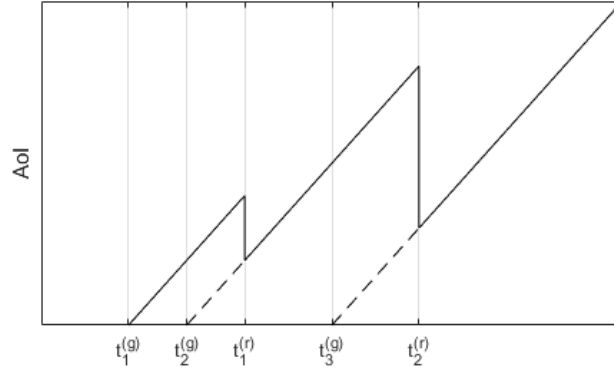


Figure 2.11: Example of AoI, where $t_n^{(g)}$ and $t_n^{(r)}$ are the generation and reception time of the n th packet, respectively.

Next, the authors of [105] discuss the measurement of AoI in practical systems. It is noted that first and foremost, accurate timing information is required for age measurement. Second, synchronization is required at the transmitter and receiver. Once these have been established, AoI can be computed at the transmitter, receiver, or centrally and used for network optimization. It is also shown how timing imperfections, such as clock bias, can affect AoI measurement.

The authors then look at a specific work which examines the AoI performance of some modern transport layer protocols. In [106], the AoI performance of User Datagram Protocol and Transmission Control Protocol are examined for different testbed setups. It is shown that User Datagram Protocol is able to maintain a lower average AoI at higher data rates than Transmission Control Protocol for a multi-hop network testbed. Both protocols tend to perform relatively well up to a certain rate, above which the system becomes “panicked” and AoI performance becomes poor. In contrast, evaluated with respect to an IoT testbed, the opposite holds true, and Transmission Control Protocol is shown to achieve a slightly better AoI performance. Overall, [105] provides a useful summary of how AoI can be practically integrated to evaluate communication systems.

As an example application of VoI, the authors of [108] propose a value-based method of information management of a networked system for state estimation. Essentially, this system will allocate a time slot to the estimator with the highest-priority information, where

priority is determined by the VoI. The VoI of each estimator is computed as a function of the expected overall weighted squared error, given the current data at that local estimator. Therefore, by choosing the data which minimizes this error, the system is essentially choosing the information with the highest value to the task at hand. To illustrate the performance of the proposed system, an automated driving scenario with multiple vehicles is simulated. It is shown that the VoI-based scheme is able to avoid collisions with very high probability, while a simple time-triggered scheduling approach resulted in a collision in 19.7% of the experiments.

Another article directly compares the impact of AoI and VoI on the performance of a cellular networked control system [107]. Here, VoI is defined as quantifying the amount of reduction in uncertainty of a stochastic process at the recipient. It is interesting to note the similarity between this definition and that of Shannon’s definition of entropy [6], which also quantifies uncertainty reduction. Here, VoI is concerned with the content of a new update, while AoI focuses only on the timeliness of this update. The AoI is defined in the usual fashion, while a VoI metric is proposed for both uplink and downlink transmissions, and in both cases is a function of the expected squared-error. Simulations demonstrate that a system implementing a VoI-based scheduler is able to achieve a lower absolute error as opposed to a system with an AoI-based schedule for the cellular networked control system.

2.6.2 Semantic Sampling

Another important aspect of a significance-based SemCom system, as proposed by the authors of both [5] and [30] is the notion of *semantic sampling*. Basically, the aim of semantic sampling is to generate information at the source in a “smart” way, such that only necessary information is generated and transmitted over the system. As a general example of this, the authors of [109] consider the problem of tracking an unstable stochastic process by using causal information of another stochastic process. Essentially, by using some information related to the process of interest, we can determine when to take “significant” samples that allow for accurate tracking, thereby implementing semantic sampling. This work can be

seen as contributing to a theory of semantic sampling as discussed in [5]. In [109], necessary conditions are provided for tracking integer-valued sources using causal information. These results are expressed in terms of the Rényi entropy and information density; essentially, the information density between the two processes must be greater than a threshold that is set by the Rényi entropy of the process being tracked. Furthermore, the authors of [109] also provide sufficient conditions for tracking integer-valued sources using causal information. The first of these is based on the maximum *a posteriori* estimator of the source information based on the causal information, and the second is based on a different estimator which considers a notion of distance. With regards to semantic sampling, the results of [109] imply that one could perhaps only sample and transmit the causal information over the channel instead of the source information itself. If the causal information results in fewer transmitted symbols, this would theoretically increase the efficiency of our system while preserving reconstruction fidelity at the receiver.

Looking at a more practical implementation of semantic sampling, the authors of [110] address the problem of semantics-aware active sampling and transmission over a shared communication medium. The goal of the system is to use joint sampling and transmission to compute the probability of a quantity of interest at the receiver. In this work, semantics-aware communication refers to a system in which the receiver aims to recover the aggregated information of interest, rather than the individual messages. For example, perhaps the average measurement from a number of sensors is of interest; in this case, meaning is captured by the aggregated value rather than the individual data. An active sampling scheme is adopted, such that each device takes samples according to a Bernoulli distribution, where the parameter of this distribution can vary between devices. To obtain the empirical probability of the quantity of interest, each device transmits over a time slot, and the receiver averages over the time slots. The estimation technique is shown to perform well with respect to both mean squared-error and Kullback-Leibler Divergence metrics.

2.6.3 Summary

One final approach to SemCom is significance-based communication, which addresses both the semantic and the effectiveness problems of communication. This recent approach makes use of metrics which quantify the significance of information, such as freshness, value, relevance, and others. Two prime examples of metrics corresponding to significance-based SemCom are AoI and VoI. AoI is a popular metric which has been well-studied compared to VoI, and some work has been done comparing the two measures. Furthermore, a key idea of this approach is semantic sampling, which has received some attention as well. Overall, significance-based SemCom looks to solve the semantic problem by first solving the effectiveness problem, and assigning meaning to information based on what impact that information will have at the receiver.

As the most recent of the discussed methods, quantitative results demonstrating the potential of significance-based SemCom for data traffic reduction are few. In [5], results are presented demonstrating energy savings, resulting in decreased transmission, of 10-64% for different age-aware protocols. While there are not many quantitative results in this area, the potential is clear. Significance-based communications center around the reduction of insignificant information, and thereby inherently work to reduce traffic in a communication system.

2.6.4 Challenges and Opportunities

One clear challenge faced by significance-based SemCom, similar to other methods of SemCom, is the development of applicable metrics. AoI and VoI are two examples which have been the focus of prior work, however we anticipate that other useful metrics will be proposed as this approach to SemCom is being developed in the literature. The quest for appropriate metrics presents a rich opportunity for future research. Clearly these measures are highly application-specific, and thus the utility of one measure may vary greatly from one scenario to another.

Another challenge relates to the general progression to such a SemCom system. As stated

in [5], this approach entails a radical departure from the ways in which current communication systems operate. However, to be a viable path forward for modern communication systems, a certain level of backward-compatibility must be present, such that the vast existing wireless infrastructure need not be replaced from scratch. Methods which involve some degree of compatibility with legacy systems will be important for the progression to SemCom.

Many opportunities lie in the development of significance-based SemCom systems. With the advancement of IoT and cyber-physical systems, wireless communication is increasingly being used for highly specific goal-oriented tasks. Designing systems which communicate as efficiently as possible under the constraints imposed by the specific task at hand will be important for optimizing the efficiency of wireless technologies, as well as improving performance of these technologies.

2.7 Chapter Summary

There are four primary existing approaches to SemCom. Classical semantic information-based approaches attempt to extend the ideas of information theory to capture the semantics of information, and are based on ideas of logical probabilities and truthlikeness. Two prominent theories are TWSI and TSSI, and truthlikeness-based approaches extend these theories. The key idea of this approach is to follow the path of classical information theory by first quantifying semantic information, and developing results based on this quantification.

KG-based SemCom focuses on the aspect of a knowledge base at a semantic source and receiver, using a KG to model such knowledge bases. This approach follows from the extensive work revolving around the semantic web. By representing knowledge in a graph structure, semantic similarity measures can be devised and forms of reasoning can be performed. This form of structure and working with knowledge is the key driver of this approach.

ML-based SemCom uses modern learning techniques to carry out SemCom in a data-driven manner. This includes DL methods, which learn semantics through neural network structures, and RL techniques, which learn semantics with an action-reward framework.

Unlike the classic and KG-based approaches, this approach assumes no formal structure of the semantics to be learned, and puts the burden of learning these semantics on the model itself. Consequently, meaning is captured by the tuned model parameters that are found as a result of data-driven training.

Finally, significance-based SemCom takes a goal-oriented approach and look to communicate in a way that best achieves the goal. Significance is quantified by metrics pertaining to different qualities of information, such as freshness and value. Semantic sampling is also a critical point of this approach, which seeks to generate only information that is pertinent to the task at hand. By addressing the effectiveness problem of communication, this approach circumvents the semantic problem and inherently assumes that the semantics will be addressed within the effectiveness solution.

Regarding the problem of data traffic reduction, various works in KG-based and ML-based SemCom have demonstrated quantitative results illustrating the potential to address this growing issue for diverse applications and use-cases, from simple text-based speech to network operations recommendation systems. For the classical and significance-based approaches, these results are fewer. One conclusion that can be drawn upon examination of these various results, is the need for standard evaluation procedures across the different approaches. If efficient communication is to be one of the main goals of SemCom, standard metrics capturing this idea should be chosen as the field develops to facilitate the comparison between competing methods.

For each approach, we provide some challenges and opportunities that could inspire future work in the field. For the classical approach, a clear opportunity lies in the further development of a theory of semantic information. The KG-based approach includes many existing methods which can be extended; particularly, scalability and learning methods are two areas which require further improvements. A major drawback of the ML-based approach is a lack of interpretability, and model-based DL and neurosymbolic AI are two potential solutions to this issue. The significance-based approach is the most recent of the four, and thus provides many exciting opportunities for future work, particularly in the development and study of significance-oriented metrics. The most important challenge to the success of

SemCom is the ability to define and work with “meaning.”

Chapter 3

Semantic Communication with Conceptual Spaces: Part I

From the example in Section 1.1, we see that SemCom holds the potential to improve efficiency over traditional technically-focused systems. Furthermore, focusing on the technical level may bring with it inefficiency stemming from the fact that a syntactic error does not necessarily induce a semantic error. For example, suppose the sender transmits the sentence “All communication is semantic,” but the receiver is presented with “All communication is semantic.” Erroneous transmissions are often addressed by re-transmission schemes or error-correcting codes, requiring additional valuable resources. However, viewed from the semantic level, this is a minor error and most likely will preserve the semantic information intended by the transmitter. We observe that semantic-oriented communication may be more robust to errors and, in turn, more efficient than traditional systems. Then the question is this: *how can we engineer semantics in a communication system?* In Chapter 2, we survey various previous attempts to answer this question in detail, and discuss their respective challenges. Looming over this issue is an even more fundamental question: *how should we define semantics?* For this question, we adopt the CS theory of semantics proposed by Peter Gärdenfors [31, 32]. This geometric representation of meaning can provide the foundation for efficient SemCom systems. To address the former question, we propose a system with a semantic en-

coder/decoder based on this theory, termed *semantic communication with conceptual spaces* (SCCS). Moreover, we propose the use of *functional compression* to optimize the semantic system, in which data from one or more sources is compressed such that a function of this data can be accurately computed [111]. We show that encoding data to preserve semantic information can be viewed as a particular example of functional compression.

3.1 Related Work

There has been a recent surge of interest in SemCom, sparked by the success of modern AI. One of the most prominent approaches is termed DeepSC, which uses DL to “learn” the semantics of text communication [4]. Another idea regards semantics as the *significance* of information [5], where the meaning of information is captured by metrics like AoI and VoI. We provide a detailed survey of literature related to SemCom in Chapter 2.

Functional compression is a generalization of the well-known source coding problem [8]. In [111], Doshi et al. extend this work to general functions of two discrete, finite sources. The problem is represented as a graph coloring problem, with feasibility conditions and optimality bounds derived. While this can be solved exactly for some nice functions, this problem is NP-hard in general.

Since Gärdenfors’ proposal of CSs in 2000, there have been some works that attempt to extend fundamentals of the proposed theory. Rickard, Aisbett, and Gibbon propose the addition of fuzzy logic to the theory and formalize many of the abstract concepts in [112], and extend this work to type-2 fuzzy spaces in [113]. In addition, some have used CSs to facilitate a diverse set of applications, from robot learning [114] to space event characterization [115].

3.1.1 Chapter Contributions

In this chapter, we show how the theory of CSs can be deployed to develop a SemCom system and demonstrate the benefits of such a system. The main contributions of this chapter include:

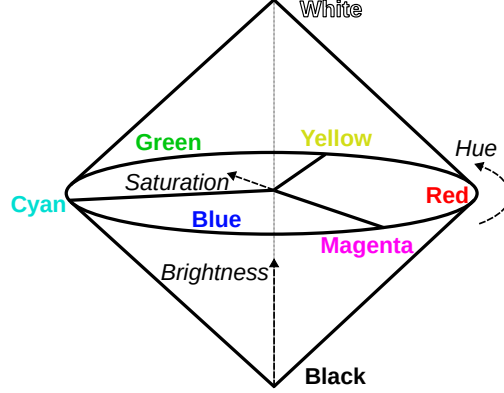


Figure 3.1: Color domain. Dashed lines indicate the three dimensions.

- A novel approach to designing SemCom systems based on CS semantics, termed SCCS.
- The introduction of a formal approach to the study of such systems using functional compression techniques.
- Confirmation of the intuitive benefits of SemCom through simulation of image transmission, demonstrating faithful transmission of meaning with 99.79% reduction in rate.

3.2 The Geometry of Meaning

In this section, we attempt to answer the latter of the two questions posed in the introduction to this chapter: *how should we define semantics?* The theory of semantics based on CSs, aptly referred to as the “geometry of meaning” by Gärdenfors, provides a convenient solution. Proposed as a cognitive model of the way humans conceptualize ideas, CSs offer a geometric view of concepts.

Summarizing the key components introduced in [31] and [32], a CS is essentially the knowledge base of an agent. A CS can be characterized as a collection of *domains*; each domain is then made up of integral *quality dimensions* that define the geometry of the domain. Based on these definitions, a *property* is defined as a convex region of a single domain, and a *concept* is defined as a collection of regions across domains within the CS.

To illustrate, consider the example of the color domain. Studies indicate that humans perceive three dimensions regarding color: hue, saturation, and brightness [31]. In the

theory of CSs, these dimensions are regarded as *integral*, because all three must be present to describe a color, and thus they make up a domain. These dimensions form the color spindle shown in Figure 3.1. An example property could be *red*, which corresponds to the convex region of the spindle with a red hue value. Another property is *dark*, which describes colors in the bottom half of the spindle. Now add a second domain to the space, e.g. the shape domain. In this domain, we have a convex region corresponding to *cube*. Then the *red* region of the color domain with the *cube* region of the shape domain define the concept of a *red cube*.

The major implication of this theory for SemCom is the formalization of *meaning as a geometric concept*. With this formalization, semantic similarity becomes a simple distance measure (provided a metric exists on the domains), and semantic error can be easily quantified. Indeed, this framework elegantly captures the idea that a syntactic error does not always induce a semantic error. When communicating an idea (a point in the CS), perhaps a syntactic error moves the idea to a different point in the space; as long as the point remains within the regions defined by the concept, a semantic error will not occur. For example, perhaps a communication error causes the transmitted color of a red cube to switch to a different shade of red. It makes no difference, as the intended meaning is preserved.

3.3 Enabling Semantic Communication

With our definition of semantics in place, we now turn to the first question posed in the introduction to the chapter: *how can we engineer SemCom?*

3.3.1 The General Model

The block diagram of our proposed approach is given in Figure 3.2. Formally, suppose there is some concept to be communicated, denoted by $\mathbf{z}$. The system begins with some data $\mathbf{x}$ describing $\mathbf{z}$, which is first passed through the *semantic encoder*; this is represented by some functional mapping $\mathbf{p} = e(\mathbf{x})$. The function e can be thought of as an *embedding* function,

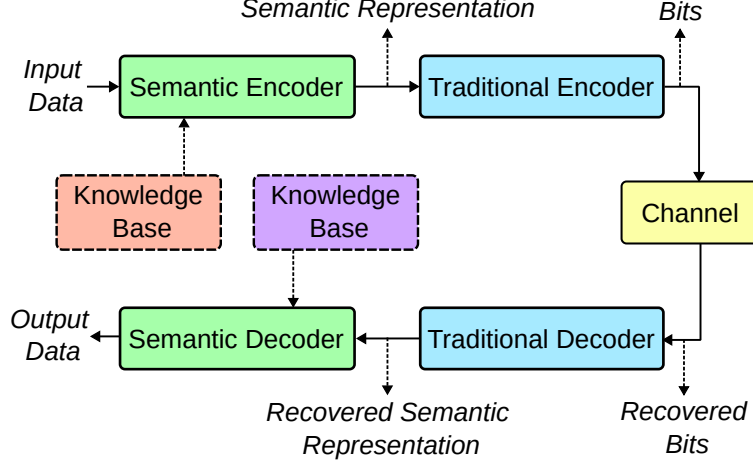


Figure 3.2: General SCCS framework, where the knowledge base represents a CS model.

which maps the description to some point $\mathbf{p}$ in the CS, where $\mathbf{p}$ is a vector of coordinates. This mapping can be either analytically derived or learned with a neural network. The coordinates $\mathbf{p}$ are then used as the input to the traditional encoder, which performs functions such as quantization, source coding, etc. and outputs a string of bits $\mathbf{b}$; in functional terms, we have $\mathbf{b} = g(\mathbf{p})$. These bits are then modulated (in the case of wireless communication) and sent over the channel. The semantic source can then be thought of as a composition of functions,

$$\mathbf{b} = g(e(\mathbf{x})). \quad (3.1)$$

The receiver begins with an input of bits $\hat{\mathbf{b}}$ which are potentially distorted by the communication channel, i.e., $\hat{\mathbf{b}} = h(\mathbf{b})$. These bits serve as the input to the traditional decoder, which performs standard functions such as source and channel decoding. Denote these functions by f , so that we have $\hat{\mathbf{p}} = f(\hat{\mathbf{b}})$, where $\hat{\mathbf{p}}$ represents the (possibly distorted) coordinates of the CS. Finally, let d denote the function de-mapping the CS representation to the concept it represents. Then the semantic receiver can be written as

$$\hat{\mathbf{z}} = d(f(\hat{\mathbf{b}})), \quad (3.2)$$

and the end-to-end semantic system is written as

$$\hat{\mathbf{z}} = d(f(h(g(e(\mathbf{x})))) = \psi(\mathbf{x}). \quad (3.3)$$

If $\hat{\mathbf{b}} \neq \mathbf{b}$, we say a *syntactic error* has occurred. If $\hat{\mathbf{z}} \neq \mathbf{z}$, then we call this a *semantic error*. Note that $\hat{\mathbf{b}} = \mathbf{b}$ (perfect syntactic communication) yields the minimum probability of semantic error. Conversely $\hat{\mathbf{b}} \neq \mathbf{b}$ does not necessarily imply a semantic error, matching our intuition of a SemCom system. Therefore, with the goal of accurate SemCom, *it may be possible to relax the requirements of the technical system while meeting the requirements of the semantic system*. First observe that g, h , and f account for the technical accuracy, which in turn will impact the semantic accuracy to some degree. Moreover, the semantic accuracy also depends on the ability of e to extract semantic representations from the initial data, and on the ability of d to recover concepts from potentially distorted semantic representations. The benefit of using CS representations is in the ability to simply quantify this semantic distortion as a distance in the CS; let this distance be denoted by $\delta(\mathbf{p}_i, \mathbf{p}_j)$. If we consider $\mathbf{p}^*$ as the central point, or prototype, of the true concept (recall a concept is composed of convex regions), then we can characterize the semantic distortion incurred throughout the overall system as

$$\delta(\mathbf{p}^*, \hat{\mathbf{p}}) = \Delta(\delta(\mathbf{p}^*, \mathbf{p}), \delta(\mathbf{p}, \hat{\mathbf{p}})), \quad (3.4)$$

where Δ is a general function of the distortions resulting from semantic encoder and syntactic error. If we further assume δ to be a metric satisfying the triangle inequality, we can write

$$\delta(\mathbf{p}^*, \hat{\mathbf{p}}) \leq \delta(\mathbf{p}^*, \mathbf{p}) + \delta(\mathbf{p}, \hat{\mathbf{p}}). \quad (3.5)$$

Thus, by bounding the two right-hand terms in (3.5), we can effectively bound the overall semantic distortion and ensure reliable SemCom. Moreover, if the semantic encoder is known to have low distortion, we can relax constraints on the technical system while maintaining an acceptable level of overall semantic distortion, and vice versa.

Communication as presented above can be thought of in two ways. The first is *general*

SemCom, which requires a broad knowledge base. An example would be human-to-human voice communication; the semantic encoder would need to be familiar with a wide array of concepts and their corresponding representations in speech. In contrast, a simpler application is *task-based SemCom*, which deals with a small set of concepts and requires a relatively narrow knowledge base. For example, suppose we want to implement a color classifier using data from wireless color sensors. The sensors observe data corresponding to some unknown color (i.e., the concept). This data propagates through the system as in (3.3), and a decision is made regarding the present color. This offers two benefits. The first is, again, that a syntactic error does not necessarily induce a semantic error. The second is that the CS representation *may be more efficient for the task at hand*. In the color example, if we only desire a decision on the color, then all that needs to be transmitted is the hue coordinate, rather than the full red-green-blue (RGB) triplet. We recognize the fact that *general SemCom* may require a complex CS, and obtaining such a model is a research challenge. We believe that SCCS will have the largest initial impact in task-specific, goal-oriented scenarios, where the domain knowledge is relatively simple to model.

3.3.2 Toward Functional Compression

As with traditional communication, we would like to optimize SemCom to be as efficient as possible through data compression. The objective of traditional source coding is to compress data for exact reconstruction:

$$\min_{g,f} E[\ell] \quad \text{s.t.} \quad \ell = \text{length}(g(\mathbf{x})) \quad \text{and} \quad f(g(\mathbf{x})) = \mathbf{x}, \quad (3.6)$$

where $\text{length}(g(\mathbf{x}))$ counts the number of (typically binary) symbols used to encode $\mathbf{x}$, and $E[\cdot]$ denotes expectation. Functional compression seeks to compress data with respect to a function:

$$\min_{g,f} E[\ell] \quad \text{s.t.} \quad \ell = \text{length}(g(\mathbf{x})) \quad \text{and} \quad f(g(\mathbf{x})) = \varphi(\mathbf{x}). \quad (3.7)$$

In particular, this can be done by identifying equivalence classes of the target function φ

to eliminate redundancies in the encoded data. An equivalence class is a subset of the input space for which φ remains constant; or, from the point of view of the receiver, a set of inputs which are indistinguishable. For example, suppose $\varphi(x) = x \bmod 2$ for x drawn uniformly from $\{0, 1, 2, 3\}$. Traditionally, 2 bits are needed to represent each value, but by identifying the equivalence classes $\{0, 2\}$ and $\{1, 3\}$, this can be reduced to 1 bit.

As mentioned above, this problem is NP-hard in general. Here, we propose approaching the problem by first taking a step back. Rather than considering general functions, we consider high-level tasks and then model these tasks using CSs. This corresponds to task-based communication as in the previous subsection, with target function $\mathbf{z} = \varphi(\mathbf{x})$. For the SemCom system, we can define the following functional compression problem:

$$\min_{e,g,f,d} E[\ell] \quad \text{s.t.} \quad \ell = \text{length}(\mathbf{b}) \quad \text{and} \quad \psi(\mathbf{x}) = \mathbf{z}. \quad (3.8)$$

In particular, by assuming a minimum-distance decoder d on the underlying CS, (3.8) can be equivalently written as

$$\min_{e,g,f} E[\ell] \quad \text{s.t.} \quad \ell = \text{length}(\mathbf{b}) \quad \text{and} \quad \delta(\mathbf{p}^*, \hat{\mathbf{p}}) \leq \tau, \quad (3.9)$$

where τ is some distortion threshold. Constructing the target function with CSs allows us to implicitly include equivalence relations based on the concepts within the space, for which efficient semantic encodings can be obtained.

3.4 Experimental Results

We examine SemCom of images using the German Traffic Sign Recognition Benchmark dataset [116], inspired by an autonomous driving application. For our tests, we define a CS with two domains, namely the color domain (Figure 3.1) and the domain of regular polygons. For the polygon domain, we consider a single dimension quantified by the ratio r of the maximum distance from the center of a shape to its boundary to the minimum of

such distance. For a triangle $r = 2$ and for a circle $r = 1$. Concepts considered in this experiment are: yellow square, red triangle, red octagon, red circle, and blue circle. The prototype values for these concepts are shown below in Table 3.1.

Table 3.1: Experimental concepts and prototype values for traffic sign data.

<i>Concept</i>	<i>Shape Ratio</i>	<i>Hue</i>	<i>Saturation</i>	<i>Brightness</i>
Yellow Square	1.4142	0.1667	1	0.9714
Red Triangle	2	0	1	0.5323
Red Octagon	1.0824	0	1	0.5323
Red Circle	1	0	1	0.5323
Blue Circle	1	0.6667	1	0.3230

We wish to compare the proposed system to a more traditional system. For the SCCS system, we train a CNN with three convolutional layers and two fully connected layers for a total of 1,692 parameters to learn the semantic encoder function e from the image space to the CS. As such, the prototype values of each concept are used as the training labels. Since hue is a circular dimension, a custom loss function is required to capture distance within the space. Denoting the coordinates of the space as $\mathbf{p} = (r \ h \ s \ b)^\top$, the loss function of the CNN is defined as

$$\delta(\mathbf{p}, \hat{\mathbf{p}}) = \frac{1}{4} \left((r - \hat{r})^2 + (s - \hat{s})^2 + (b - \hat{b})^2 + \gamma(h, \hat{h})^2 \right), \quad (3.10)$$

where

$$\gamma(h, \hat{h}) = -\frac{1}{\rho} \ln \left(\frac{1}{2} e^{-\rho|h-\hat{h}|} + \frac{1}{2} e^{-\rho(1-|h-\hat{h}|)} \right)$$

is an approximation of the circular distance given by $\min(|h - \hat{h}|, |1 - (h - \hat{h})|)$ with parameter $\rho > 0$. The output of the semantic encoder is then quantized to n_b bits/value to form a $4n_b$ -bit packet, which is modulated using binary phase-shift keying (BPSK) and transmitted over a Rayleigh fading channel. At the receiver, BPSK demodulation is performed and a minimum-distance decoder is used to recover the concept.

We also simulate a traditional system which transmits the entire 25x25-pixel image over the channel. Each pixel is represented as three n_b -bit values, which undergo identical modulation and channel effects as above. At the receiver, a CNN with identical structure to that of the semantic system classifies the concept from the recovered image. This network is

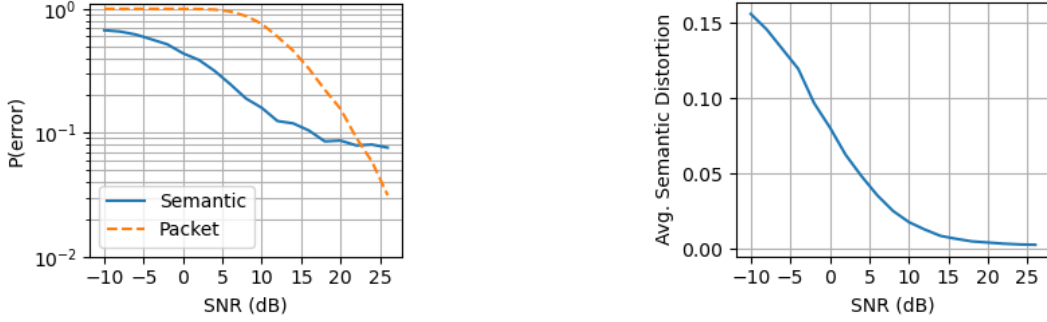


Figure 3.3: Simulation results for the SCCS system. Comparison of semantic and syntactic probabilities of error (left) and average end-to-end semantic distortion $\delta(\mathbf{p}^*, \hat{\mathbf{p}})$ (right).

trained using the classic cross-entropy loss function, using one-hot encoded vectors as labels.

First, we demonstrate that a syntactic error does not necessarily induce a semantic error. We define a semantic error as an incorrectly recovered concept ($\hat{\mathbf{z}} \neq \mathbf{z}$), whereas a syntactic error occurs when the transmitted packet contains one or more bit errors ($\hat{\mathbf{b}} \neq \mathbf{b}$). Figure 3.3 (left) shows the performance for the semantic system with $n_b = 8$ when the SNR is varied. It is observed that the system is able to perform well *semantically* despite significant syntactic errors; for example, at 15dB, approximately 40% of transmitted packets contain errors, but the meaning is still accurately conveyed 90% of the time. As the channel conditions improve, the performance becomes limited at an SNR of approximately 20dB due to the nonzero semantic distortion floor illustrated in Figure 3.3 (right). By improving the semantic encoder, the high-SNR performance of the overall system will improve accordingly.

Next we compare the performance of the two systems as a function of the communication rate; these results are given in Table 3.2. Here, the rate of the semantic system is $4n_b$ bits/transmission, while the rate of the traditional system is $(25 \times 25 \times 3)n_b$ bits/transmission, and n_b is varied to obtain different rate values. Thus, the semantic system achieves a rate reduction of 99.79% over the traditional system. Furthermore, we observe that the semantic system is able to make better use of fewer bits, as the meaning conveyed is significantly more accurate at lower values of n_b than for the traditional system. Conversely, as the number of quantization bits increases, the traditional system is shown to outperform the semantic system at the cost of dramatically greater communication rate.

Table 3.2: Rate (bits/transmission) and semantic error probability (SNR = 15dB).

n_b	<i>Semantic System</i>		<i>Traditional System</i>		<i>Rate Reduction</i>
	<i>Rate</i>	$P(\hat{\mathbf{z}} \neq \mathbf{z})$	<i>Rate</i>	$P(\hat{\mathbf{z}} \neq \mathbf{z})$	
2	8	17.1%	3750	70.9%	99.79%
5	20	10.7%	9375	50.5%	
8	32	10.9%	15000	4.4%	

3.5 Chapter Summary

In this chapter, we introduced a novel approach to SemCom through the use of CS theory. We described the problem of SemCom, using a simple example to illustrate some of the potential benefits. We then formed our definition of semantics utilizing the theory of CSs, which provides a geometric representation of meaning. We described the model of a SemCom system under this theory, termed SCCS, and proposed functional compression as a method of obtaining optimal encoding schemes for the semantic system. Through simulation of image transmission, we provided quantitative results illustrating the ability of the SCCS system to faithfully convey meaning with a massive reduction in communication rate.

Chapter 4

Semantic Communication with Conceptual Spaces: Part II

In the previous chapter, we presented the SCCS system for SemCom, utilizing the theory of CSs for representing meaning. The results obtained from simple preliminary experiments demonstrated the potential of SCCS for robust SemCom with massive reductions in the technical communication rate. Motivated by these results, in this chapter we take a deeper dive into the specifics of the proposed SCCS framework. We rigorously define components of the system in greater detail, introduce for the first time various theoretical and practical aspects of our approach, and further demonstrate its potential through the use of extensive simulations of a more realistic application scenario than that of Chapter 3. Additionally, we make the case for the CS model of meaning as an ideal candidate for a knowledge-driven foundation for higher-level reasoning-driven systems.

4.1 Related Work

Similar to the SCCS approach, the notion of a CS was also recently utilized in [117] as a component of the authors' proposed SemCom framework. However, their treatment is limited to a space with only traditional Euclidean dimensions, while our model allows for

more general dimensions that may better match human knowledge [31]. Moreover, the focus of [117] is not on the model itself, but rather the development of a reasoning mechanism given such a model. In this chapter, we take a deeper look into the CSs framework in relation to SemCom and retain the requirement that the dimensions of such a space should model some human-interpretable qualities, as initially defined in [31]. This is in stark contrast to the common approach of training DNNs in an end-to-end fashion [4], in which the encoder maps data to some vector space with uninterpretable features. The framework of CSs presented here models semantic information directly such that the semantics of the raw data can be quantified and interpreted, unlike most of the approaches in the current literature.

4.2 Chapter Contributions

In this chapter, we aim to formalize key theoretical aspects of the SCCS framework in order to further demonstrate its potential for future SemCom systems. We further provide insight on how our approach can lay the groundwork for higher-level reasoning-based systems, such as the one proposed in [118]. We provide thorough simulation results from a metaverse-inspired problem setup to confirm the theoretical developments and demonstrate the potential technical benefit of our approach regarding syntactic efficiency and semantic performance for future wireless applications. The main contributions of this chapter can be summarized as:

- We uncover key theoretical aspects of the SCCS framework. Specifically, we provide formal definitions of the various elements (domain, property, etc.) and examine the central concept of semantic distortion and semantic error under these definitions.
- We further analyze this notion of semantic distortion, and derive bounds on the probability of semantic error under our proposed framework. The derived bounds are simple to obtain under reasonable assumptions and are based on the level of semantic distortion introduced at the semantic encoder and the channel, and can thus be used to inform robust system design.

- We show how the CS model of semantics can serve as a foundation of knowledge for building future reasoning-driven SemCom systems such as that proposed in [118]. Specifically, we show how the notion of a *semantic language* [118] naturally arises from a CS and how contextual information can be easily incorporated into the CS model.
- We conduct extensive simulations of the proposed approach on a metaverse-inspired problem. We examine the problem of virtual reality exposure therapy (VRET) [119] and simulate a SCCS system designed around this problem.
 - We simulate a system with a theoretical semantic encoder able to attain an arbitrary level of performance. We show that the end-end performance is inevitably limited by the quality of the semantic encoder, and that the proposed system can be semantically robust under a fading channel model. Specifically, comparing performance under Rician fading to that of the AWGN channel, we find that in many cases an increase of only 2-3dB in SNR is needed under the fading channel model to achieve similar semantic performance to that of the AWGN model.
 - A practical SemCom system is simulated, training a CNN as the semantic encoder. We show that the proposed approach can reduce the communication rate by over 99.9% as compared to a more traditional system, while achieving similar semantic performance at extreme SNR values. Additionally, the proposed technique exhibits significant performance gains in the mid-high SNR regions as compared to an alternative SemCom technique, decreasing the probability of semantic error by around 0.25. Finally, we show that even greater gains can be achieved by fine-tuning the semantic encoder to the specific task at hand.

4.3 Foundations

In this section, we provide a detailed background on the theory of CSs and formally define the key elements used in our SCCS approach to SemCom. We then present the SCCS framework first discussed in Chapter 3 in greater detail.

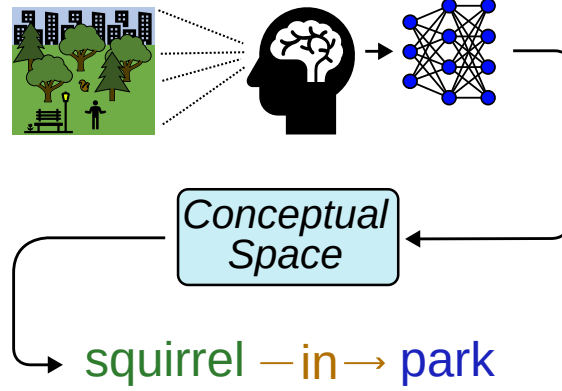


Figure 4.1: Visualization of the different levels of cognitive representation: associationism (top), conceptual (middle), and symbolic (bottom).

4.3.1 A Geometric View of Semantics

In [31] Peter Gärdenfors describes his theory of CSs, which is proposed as a cognitive model for how ideas and thoughts take shape in the human mind. There, Gärdenfors defines three levels of cognitive representations. The *associationism* level models cognitive representations as associations between different elements. He names *connectionism* as a special case, where these associations are modeled by artificial neuron networks. In contrast, the *symbolic* level uses discrete symbols as the primary cognitive representation and treats cognition as symbol manipulation. These two representations can be seen in the approaches to SemCom described in Chapter 2.

Gärdenfors proposes a third level of representation, termed the *conceptual* level, which acts as a bridge between these two. This kind of representation is “based on geometrical structures, rather than symbols or connections among neurons” [31, p. 2]. Rather than oppose each other, these different levels come together to form human thought. This idea is aligned with the recent interest in *neurosymbolic AI* [56], and is shown in Figure 4.1. The elements of the conceptual level of representation are originally described somewhat qualitatively [31], so we will now provide formal definitions for each of them.

The basic building block of a CS is a *quality dimension*, which is used to quantify some “quality” of an object or idea in some domain. We formalize this notion as a set.

Definition 1. (Quality Dimension) A quality dimension is a set of scalar values quantifying

some quality of an idea or object. A key point here is that these dimensions should be interpretable; typical ML-generated features in many popular approaches to SemCom do not fit this criterion. We denote a quality dimension by $\mathcal{Q}$, where specific values along a quality dimension are denoted by q , such that $q \in \mathcal{Q}$.

Quality dimensions are organized into *domains*. Quality dimensions that share a domain are said to be *integral*; Gärdenfors describes integral quality dimensions as being those such that “one cannot assign an object a value on one dimension without giving it a value on the other.” [32, p. 22]. Basically, domains collect related qualities and combine them to form a geometric space. Thus, we can use quality dimensions to build a set representing the domain.

Definition 2. (Domain) A domain $\mathcal{D}$ is a set that is constructed from the Cartesian product of integral quality dimensions, i.e., $\mathcal{D} = \times_{k=1}^K \mathcal{Q}_k$. A point within a domain is specified by a column vector of quality values $\mathbf{d} = (q_1 \ q_2 \ \cdots \ q_K)^\top$.

We use $^\top$ to denote the transpose operator. With a slight abuse of terminology, we refer to K as the dimensionality of domain $\mathcal{D}$. Furthermore, we can think of ideas and objects as having *properties*. These properties can be thought of as grouping similar objects and ideas together, and are defined with respect to a single domain.

Definition 3. (Property) A property $\mathcal{P}$ of a single domain $\mathcal{D}$ is defined as a convex subset of that domain, i.e., $\mathcal{P} \subset \mathcal{D}$ where, for $\lambda \in [0, 1]$ and any $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{P}$, $\lambda \mathbf{p}_1 + (1 - \lambda) \mathbf{p}_2 \in \mathcal{P}$.

The intuition underlying the convexity requirement is straightforward; if two objects possess a property, objects that are conceptually “in between” these two objects should also possess the same property. Allowing for multiple domains gives rise to a CS, which allows us to represent the meaning of ideas and objects in a structured manner.

Definition 4. (Conceptual Space) A conceptual space $\mathcal{Z}$ is a set which is formed from the Cartesian product of $M \geq 1$ domains, i.e. $\mathcal{Z} = \times_{m=1}^M \mathcal{D}_m$. An idea or object is uniquely specified within the CS by providing complete coordinates within every domain, i.e., $\mathbf{z} = \begin{pmatrix} \mathbf{d}_1^\top & \mathbf{d}_2^\top & \cdots & \mathbf{d}_M^\top \end{pmatrix}^\top$.

Finally, we need to define a structure corresponding to the notion of a *concept*. Gärdenfors describes a concept as a collection of regions across domains throughout the CS, as well as “correlations between the regions from different domains” [32, p. 24]. We will not consider the correlations between concepts in this work, and instead focus solely the regions. Including the correlations between these regions is an interesting direction for future work.

Definition 5. (Concept) A concept $\mathcal{C}$ is a region within the CS spanning one or more domains. If contained within a single domain, we have $\mathcal{C} \subset \mathcal{D}$, and if the concept spans across N domains, we have $\mathcal{C} \subset \times_{n=1}^N \mathcal{D}_n$.

A particular example of a concept is the Cartesian product of N properties, $\mathcal{C} = \times_{n=1}^N \mathcal{P}_n$. We now present a simple example to clarify these definitions.

Example 1. (Colors and shapes) Consider the domain of colors. Studies indicate that humans perceive three dimensions regarding color: hue ($\mathcal{Q}_1$), saturation ($\mathcal{Q}_2$), and brightness ($\mathcal{Q}_3$) [31]. These are integral *quality dimensions* for colors; a color cannot have a hue without also assigning it a saturation value. Thus, these dimensions form a *domain*, which was shown in Figure 3.1 of Chapter 3. Note that the hue dimension is circular; this will have certain implications when defining semantic distortion. Also, the dependence between the brightness and saturation dimensions that determines the spindle-like shape requires an additional condition beyond the simple Cartesian product, which would otherwise result in a cylinder. Such extra conditions can easily be incorporated into the prior definitions to extend the model.

Given the domain shown in Figure 3.1, an example property might be *light red*. This property $\mathcal{P}_1$ is the convex region in the top-half of the spindle with hue values in the range corresponding to red. Now consider the domain of *regular polygons*, having one discrete dimension quantifying the number of sides of the polygon, i.e., $\mathcal{D} = \mathcal{Q}_4$, where $q_4 = 3$ corresponds to triangles, $q_4 = 4$ corresponds to squares, and so on. Furthermore, define the singleton set $\{4\}$ as the *square* property $\mathcal{P}_2$. Then we can consider the Cartesian product of these two domains as the *color-regular polygon CS*, and the Cartesian product of $\mathcal{P}_1$ and $\mathcal{P}_2$ as the concept of a *light red square*.

Example 1 shows how a simple CS model can be designed to represent meaning. In the next subsection, we provide the details of our general framework for SemCom utilizing CSs. As we will see, this model allows for interpretable and efficient representation of knowledge is a SemCom system.

4.3.2 Toward Semantic Communication

Recall Figure 3.2 from Chapter 3, which provides the general block diagram of a framework for knowledge-driven SemCom. The proposed framework builds on existing communication paradigms with the addition of a semantic encoder and a semantic decoder. First, the semantic encoder leverages its knowledge base $\mathcal{K}$ to map some source data $\mathbf{x}$ to a representation of its meaning $\mathbf{s}$, which we denote generally as vectors. Formally, we represent the semantic encoder as a mapping $e_{\mathcal{K}} : \mathbb{R}^N \rightarrow \mathcal{S}$ such that

$$e_{\mathcal{K}}(\mathbf{x}) = \mathbf{s}, \quad (4.1)$$

where the input $\mathbf{x}$ has N dimensions and $\mathcal{S}$ denotes the semantic representation space.

We assume the existence of some CS $\mathcal{Z}$ capturing the meaning of the input data $\mathbf{x} \in \mathbb{R}^N$. Given this assumption, we generally define *knowledge* as an understanding of the relationship between the observed data and the underlying CS. Thus, the *knowledge base* can be thought of as consisting of the space $\mathcal{Z}$ as well as the mappings to and from this space. Therefore, we can rewrite (4.1) as

$$e_{\mathcal{Z}}(\mathbf{x}) = \mathbf{z}, \quad (4.2)$$

where we define $\mathbf{z} \in \mathcal{Z}$ as the semantic representation of the data $\mathbf{x}$. As shown in Figure 3.2, this semantic representation is then transmitted traditionally over a channel. We represent this process as a composition of functions

$$\hat{\mathbf{z}} = g(h(f(\mathbf{z}))), \quad (4.3)$$

where $f : \mathcal{Z} \rightarrow \{0, 1\}^K$ denotes the traditional source/channel coding functions mapping $\mathbf{z}$ to a message of K bits, $h : \{0, 1\}^K \rightarrow \{0, 1\}^K$ represents physical transmission of the signal, and $g : \{0, 1\}^K \rightarrow \mathcal{Z}$ captures the traditional decoding functions.

After the approximate semantic representation $\hat{\mathbf{z}} \in \mathcal{Z}$ is recovered, it is passed to the semantic decoder. The decoder is assumed to share the same knowledge base as the encoder¹ and is thus denoted as a function $d_{\mathcal{Z}}$. The behavior of the semantic decoder will vary depending on the communication goal. For example, if the goal of communication is to perform some task, say, classification over C classes, the semantic decoder might take the form $d_{\mathcal{Z}} : \mathcal{Z} \rightarrow \Delta^{C-1}$ with

$$d_{\mathcal{Z}}(\hat{\mathbf{z}}) = \mathbf{c}, \quad (4.4)$$

where $\mathbf{c}$ is a C -dimensional probability vector in the probability simplex Δ^{C-1} .

In this chapter, we consider a particular semantic decoder. Let the primary goal of communication be to successfully convey a *concept* to the receiver, i.e., to convey meaning. Assuming that a finite number of concepts are defined over the CS $\mathcal{Z}$, define a set $\mathcal{J}$ as the *index* set of these concepts

$$\mathcal{J} = \{1, 2, \dots, J\}, \quad (4.5)$$

where J is the total number of concepts defined over $\mathcal{Z}$, and $j \in \mathcal{J}$ corresponds to concept $\mathcal{C}_j$. Then we can consider the semantic decoder as a function $d_{\mathcal{Z}} : \mathcal{Z} \rightarrow \mathcal{J}$ defined as

$$d_{\mathcal{Z}}(\hat{\mathbf{z}}) = \hat{j}, \quad \hat{j} \in \mathcal{J}. \quad (4.6)$$

In the following section, we use these terms to define the notion of a semantic error within the context of SCCS.

The proposed framework represents a paradigmatic shift from traditional communication. It considers knowledge as an integral part of the communication process, rather than viewing it as only the semantics-agnostic process of transmitting bits. In the CS framework, a

¹SemCom with mismatched CSs at the transmitter and receiver is an interesting problem that is out of the scope of this paper, and will be considered in future work.

concept can be viewed as a sort of “semantic codeword.” In contrast to the idea of a codeword in the traditional sense, or even in the sense of black-box ML-based SemCom, these codewords correspond, by definition, to some human-interpretable concept. As will be seen in the following section, this allows for an intuitive, general definition of semantic distortion, which is a significant distinction from the notion of codeword distance in the traditional setting. Moreover, as will be seen, the notion of a “semantic error” can be well-defined mathematically within this framework while also maintaining intuitive significance as a true error with regards to some meaning.

4.4 Semantic Distortion and Error Analysis

With the formal SCCS framework in place, we now turn to the problem of quantifying semantic distortion and defining semantic errors. In this section, we will show that using CSs to model knowledge leads to a notion of semantic distortion that is natural, intuitive, and interpretable. Furthermore, we use this notion to derive bounds of the probability of semantic error. We close this section by discussing some of the practical implications of this theory.

4.4.1 Semantic Distortion in a Conceptual Space

As we have defined our model of knowledge representation in terms of geometric structures, it is natural to think of semantic distortion as a *distance measure*. Doing so gives the ability to compare the meanings of different objects and ideas. First, consider the distance between points within a single domain. For the m th domain $\mathcal{D}_m$, we define this distance as a function mapping two points in the domain to the non-negative real line, i.e. $\delta_m : \mathcal{D}_m \times \mathcal{D}_m \rightarrow \mathbb{R}_+$. While any general mapping can be considered, we are interested in functions that satisfy the three conditions of a metric, namely *non-negativity*, *symmetry*, and the *triangle inequality*.

Respectively,

$$\delta_m(\mathbf{d}_1, \mathbf{d}_2) \geq 0, \quad (4.7)$$

$$\delta_m(\mathbf{d}_1, \mathbf{d}_2) = \delta_m(\mathbf{d}_2, \mathbf{d}_1), \quad (4.8)$$

$$\delta_m(\mathbf{d}_1, \mathbf{d}_3) \leq \delta_m(\mathbf{d}_1, \mathbf{d}_2) + \delta_m(\mathbf{d}_2, \mathbf{d}_3). \quad (4.9)$$

Given δ_m for each of the M domains, we can now define semantic distortion.

Definition 6. (Semantic Distortion) Semantic distortion is a function mapping two points in the CS $\mathcal{Z}$ to the non-negative real line $\delta : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}_+$ given by

$$\delta(\mathbf{z}_1, \mathbf{z}_2) = \sum_{m=1}^M \delta_m(\mathbf{z}_1, \mathbf{z}_2) \quad (4.10)$$

where $\delta_m(\mathbf{z}_1, \mathbf{z}_2) = \delta_m(\mathbf{d}_1, \mathbf{d}_2)$ is computed using the coordinates in $\mathbf{z}_1, \mathbf{z}_2$ corresponding to domain $\mathcal{D}_m$.

Since δ is defined on the CS, it is inherently interpretable. This provides insight on *why* two elements differ semantically, as this is due to differences in the quality domains. This is a significant advantage over uninterpretable semantic distortion/similarity measures, such as the BERT-based similarity measure [120].

Example 2. (Distance in the color domain) Recall the color domain of Example 1. To determine a distance function for this domain, we must take into account its specific geometry. If we let the hue dimension take values in the interval $[0, 1]$, we cannot use absolute or squared difference to represent distance, since we need to capture the circularity. One way to represent distance on this dimension is with the expression

$$\min(|h - \hat{h}|, |1 - (h - \hat{h})|), \quad (4.11)$$

where h represents the hue value. To avoid taking a minimum, (4.11) can be approximated

as

$$\gamma(h, \hat{h}) = -\frac{1}{\rho} \ln \left(\frac{1}{2} e^{-\rho|h-\hat{h}|} + \frac{1}{2} e^{-\rho(1-|h-\hat{h}|)} \right), \quad (4.12)$$

with parameter $\rho > 0$. Since the brightness and saturation dimensions are linear, we can let them take values in the interval $[0, 1]$ and use the squared difference to represent distance. Then we define distance within the domain as a mean squared error-like function given by

$$\delta_{\text{color}}(\mathbf{d}, \hat{\mathbf{d}}) = \frac{1}{3} \left((s - \hat{s})^2 + (b - \hat{b})^2 + \gamma(h, \hat{h})^2 \right), \quad (4.13)$$

where s and b are the saturation and brightness values, respectively, and $\mathbf{d} = (h \ s \ b)^\top$.

Note that if each of the domain-specific distance functions satisfy the metric conditions, then the semantic distortion function given by (4.10) will also be a metric. Next, we'll use this observation to derive bounds on the probability of semantic error under some assumptions.

4.4.2 Bounding the Probability of a Semantic Error

In traditional information theory, an error is characterized by the reception of an incorrect symbol. As discussed toward the end of section 4.3.2, we are interested in communication when the primary goal is to convey a concept. Thus, using the notations of (4.5) and (4.6), we can define the notion of a *semantic error*.

Definition 7. (Semantic Error) A semantic error occurs when the concept decoded by the receiver does not match the concept that is to be conveyed, i.e.

$$d_Z(\hat{\mathbf{z}}) = \hat{j} \neq j^*, \quad (4.14)$$

where j^* is the index of the true concept $\mathcal{C}_{j^*}$ to be communicated.

This definition of a semantic error is rather intuitive. If we take the meaning of an idea or object to be represented by its corresponding concept in a CS, the *successful conveyance of meaning* boils down to the successful communication of these concepts. This also matches

the intuition that a syntactic error does not necessarily induce a semantic error. Perhaps the technical communication of $\mathbf{z}$ contains errors, such that $\hat{\mathbf{z}} \neq \mathbf{z}$, i.e., the CS coordinates obtained by the receiver have *shifted* to a new location within the space. Since concepts are represented as regions, it is possible that $\hat{\mathbf{z}}$ will be decoded as the correct concept. Moreover, this differs from other notions of a semantic error in that it is, by definition, grounded in an interpretable space of quality dimensions. Thus we can answer questions such as “why did an error occur?” This opens the door for human knowledge-based higher level intelligence in SemCom systems.

To derive formal bounds on the probability of semantic error, we will make a few assumptions:

- A1. Concepts are formed from the Cartesian products of properties. Thus, concepts are *convex* regions within the space, and each concept $\mathcal{C}_j$ has a *concept prototype* $\mathbf{z}_j$.
- A2. Semantic distortion $\delta : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}_+$ is a metric.
- A3. The semantic decoder is a minimum-distance decoder: $d_{\mathcal{Z}}(\hat{\mathbf{z}}) = \operatorname{argmin}_{j \in \mathcal{J}} \delta(\mathbf{z}_j, \hat{\mathbf{z}})$.

Assumption A1 is intuitive, as a concept can often be expressed as a combination of properties, e.g., the concept of a “red square” is a combination of the properties “red” and “square.” Since properties are convex regions by definition, we can meaningfully compute a prototype point $\mathbf{z}_j$ as the centroid of these convex regions.

Assumption A2 ensures that the chosen distortion function behaves like a true measure of distance. Given these first two assumptions, we can obtain a *Voronoi tessellation* of the CS [31]. A Voronoi tessellation partitions a geometric space into convex regions, where the region of the tessellation corresponding to a prototype point $\mathbf{z}_j$ is defined as all points $\mathbf{z}$ such that $\delta(\mathbf{z}, \mathbf{z}_j) \leq \delta(\mathbf{z}, \mathbf{z}_i)$ for all $i \in \mathcal{J}$ where $i \neq j$. Intuitively, it is the region of all points closer to the prototype of concept $\mathcal{C}_j$ than any other prototype. An example is provided in Figure 4.2.

These first two assumptions lead naturally to a minimum-distance decoding scheme, expressed as the final assumption A3. Under this scheme, a concept will be decoded correctly

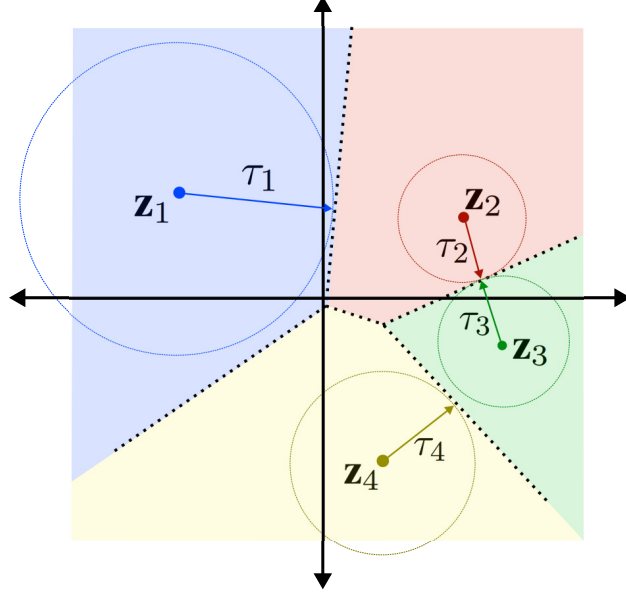


Figure 4.2: Example Voronoi tessellation with four concept prototypes on $\mathbb{R}^2$.

if the received semantic representation $\hat{\mathbf{z}}$ lies within the Voronoi region of the prototype point $\mathbf{z}_{j^*}$. Conversely, a semantic error will occur when $\hat{\mathbf{z}}$ lies *outside* this region. Let the Voronoi region corresponding to concept $\mathcal{C}_j$ (equivalently, prototype $\mathbf{z}_j$) be denoted by $\mathcal{V}_j$. Then we can express the probability of semantic error as in Definition 7 as

$$P(\hat{j} \neq j^*) = P(\hat{\mathbf{z}} \notin \mathcal{V}_{j^*}). \quad (4.15)$$

To obtain a bound on this probability, first consider the value τ_j which we define as the solution to

$$\begin{aligned} \max_{\mathbf{z} \in \mathcal{Z}} \quad & \delta(\mathbf{z}, \mathbf{z}_j) \\ \text{s.t.} \quad & \delta(\mathbf{z}, \mathbf{z}_j) \leq \delta(\mathbf{z}, \mathbf{z}_j) : j \in \mathcal{J}, j \neq j. \end{aligned} \quad (4.16)$$

Intuitively, τ_j is the radius of the largest sphere centered at $\mathbf{z}_j$ and inscribed in the Voronoi region $\mathcal{V}_j$; this notion is illustrated in Figure 4.2.

Next, consider the two potential sources of semantic distortion. Recall, the semantic encoder maps data $\mathbf{x}$ to a semantic representation $\mathbf{z}$. An imperfect encoder might introduce

some level of semantic distortion $\delta(\mathbf{z}_{j^*}, \mathbf{z})$. Moreover, syntactic errors may introduce some distortion $\delta(\mathbf{z}, \hat{\mathbf{z}})$. In general, the end-to-end distortion will be a function of these two values. Under assumption A2 that δ is a metric, the triangle inequality holds and we have

$$\delta(\mathbf{z}_{j^*}, \hat{\mathbf{z}}) \leq \delta(\mathbf{z}_{j^*}, \mathbf{z}) + \delta(\mathbf{z}, \hat{\mathbf{z}}) \quad (4.17)$$

We can now derive an upper bound on the probability of semantic error.

Lemma 1. *Let α_j denote the prior probability that concept $\mathcal{C}_j$ is chosen as the true concept $\mathcal{C}_{j^*}$. Then the probability of semantic error $\hat{j} \neq j^*$ has the upper bound*

$$P(\hat{j} \neq j^*) \leq \sum_{j \in \mathcal{J}} \alpha_j P(\delta(\mathbf{z}_j, \mathbf{z}) + \delta(\mathbf{z}, \hat{\mathbf{z}}) > \tau_j). \quad (4.18)$$

Proof. Recall that the Voronoi region corresponding to concept $\mathcal{C}_j$ is denoted by $\mathcal{V}_j$. Then

$$\begin{aligned} P(\text{sem. error}) &= P(\hat{j} \neq j^*) \\ &= \sum_{j \in \mathcal{J}} P(\hat{j} \neq j^* \mid j^* = j) P(j^* = j) \\ &= \sum_{j \in \mathcal{J}} \alpha_j P(\hat{j} \notin \mathcal{V}_j) \\ &\leq \sum_{j \in \mathcal{J}} \alpha_j P(\delta(\hat{\mathbf{z}}, \mathbf{z}_j) > \tau_j) \\ &\leq \sum_{j \in \mathcal{J}} \alpha_j P(\delta(\mathbf{z}_j, \mathbf{z}) + \delta(\mathbf{z}, \hat{\mathbf{z}}) > \tau_j) \quad \blacksquare \end{aligned}$$

Note that to compute (4.18), the probability distribution over the set of concepts is required. If this distribution is unknown, with one additional assumption a looser bound can be derived.

Lemma 2. *Define $\tau = \min_{j \in \mathcal{J}} \tau_j$. If the distribution of the distortion introduced by the semantic encoder is independent of the concept $\mathcal{C}_j$ being encoded, then the probability of*

semantic error has the upper bound

$$P(\hat{j} \neq j^*) \leq P(\delta(\mathbf{z}_j, \mathbf{z}) + \delta(\mathbf{z}, \hat{\mathbf{z}}) > \tau). \quad (4.19)$$

Proof. Beginning in the same manner as in the proof of Lemma 1, we arrive at the same result. Moreover, since $\tau \leq \tau_j$ for all $j \in \mathcal{J}$, we have

$$\begin{aligned} P(\text{semantic error}) &\leq \sum_{j \in \mathcal{J}} P(\delta(\mathbf{z}_j, \mathbf{z}) + \delta(\mathbf{z}, \hat{\mathbf{z}}) > \tau_j) P(j^* = j) \\ &\leq \sum_{j \in \mathcal{J}} P(\delta(\mathbf{z}_j, \mathbf{z}) + \delta(\mathbf{z}, \hat{\mathbf{z}}) > \tau) P(j^* = j) \end{aligned}$$

Looking at the first probability term inside the summation, we see that the only term dependent on j is $\delta(\mathbf{z}_j, \mathbf{z})$. By assumption, this random distortion is independent of the concept $\mathcal{C}_j$ being encoded, and thus the entire term does not depend on j . Therefore, we can write

$$\begin{aligned} P(\text{semantic error}) &\leq P(\delta(\mathbf{z}_j, \mathbf{z}) + \delta(\mathbf{z}, \hat{\mathbf{z}}) > \tau) \sum_{j \in \mathcal{J}} P(j^* = j) \\ &= P(\delta(\mathbf{z}_j, \mathbf{z}) + \delta(\mathbf{z}, \hat{\mathbf{z}}) > \tau) \quad \blacksquare \end{aligned}$$

4.4.3 Practical Implications

The bounds derived in Lemmas 1 and 2 can guide the design of robust SemCom systems. For example, in (4.18) we see that the bound is determined by four values, namely, the prior distribution of concepts α_j , the quality of semantic encoding $\delta(\mathbf{z}_j, \mathbf{z})$, the quality of syntactic communication $\delta(\mathbf{z}, \hat{\mathbf{z}})$, and structure of the CS τ_j . Taking the α_j 's to be fixed, this leaves three variables that can be tuned to achieve the desired performance. First, the CS should be designed such that common concepts (i.e., large α_j) have a large τ_j , decreasing the probability of a misunderstanding when communicating that concept. After this, the traditional portion of the communication system can be determined, fixing the distribution of $\delta(\mathbf{z}, \hat{\mathbf{z}})$. Finally, the semantic encoder can be designed to meet performance requirements, for instance by

training a DNN to meet the maximum distortion with high probability. Conversely, one can fix the semantic encoder and relax the requirements of the syntactic communication to achieve maximum technical efficiency while maintaining semantic performance. We end this section with an illustrative example of this process.

Example 3. (Semantic system design) Suppose we want to design a SemCom system to communicate three concepts, denoted $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$. The probabilities of these concepts are known and fixed² at $\alpha_1 = 0.5$, $\alpha_2 = 0.25$, and $\alpha_3 = 0.25$. After the CS model is designed, it is found that $\tau_1 = 3$, $\tau_2 = 2$, and $\tau_3 = 1$. For simplicity, suppose that the components of semantic distortion introduced by both the encoder and the syntactic communication are independent of the concepts, and that they can be modeled as exponential random variables, i.e., $\delta_{enc} \sim \exp(\lambda_1)$ and $\delta_{trad} \sim \exp(\lambda_2)$. Parameters λ_1 and λ_2 reflect the capabilities of the system components, e.g., a greater λ value indicates that large distortion values are less likely. After designing the semantic encoder, it is found that $\lambda_1 = 2$. We would like to design the syntactic portion of the system (channel coding, modulation, etc.) such that the probability that a semantic error occurs is less than 0.05. First, note that the sum of distortions follows a hypoexponential distribution. Letting $\delta = \delta_{enc} + \delta_{trad}$, the PDF of this random variable is given by the expression

$$p_\delta(x) = \frac{2\lambda_2}{2 - \lambda_2} (e^{-\lambda_2 x} - e^{-2x}), \quad x \geq 0. \quad (4.20)$$

Integrating this PDF, we compute the probability of this sum exceeding some value τ_j as

$$P(\delta > \tau_j) = \int_{\tau_j}^{\infty} p_\delta(x) dx = \frac{2e^{-\lambda_2 \tau_j} + \lambda_2 e^{-2\tau_j}}{2 - \lambda_2}. \quad (4.21)$$

Substituting this into the right hand side of (4.18), setting equal to 0.05, and solving we find that $\lambda_2 \approx 1.5$. Designing the syntactic portion of the system to meet this specification will ensure that the desired performance is achieved without wasted resources. For $\lambda_2 < 1.5$,

²Note that all values and distributions in this example are arbitrary, and are only meant to illustrate the process of SemCom system design.

the distortion will be too great to guarantee the desired performance. However, if $\lambda_2 > 1.5$, resources will be wasted on unnecessary technical accuracy beyond what is required for the semantic goal.

4.5 Knowledge-Driven Semantic Communication

In [118], the authors outline a vision and framework for future SemCom systems imbued with higher-level intelligent capabilities. In this section, we briefly show how the SCCS framework naturally lends itself as a knowledge-driven foundation to this kind of higher-level framework, where the notion of a “semantic language” as in [118] is elegantly realized by the CS model. We then demonstrate how semantic distortion can be generalized to handle the notion of context and conclude with a discussion of semantic ambiguity, semantic redundancy and scalability as they relate to our framework.

4.5.1 Data-Driven to Reasoning-Driven Intelligence

The authors of [118] characterize the evolution of the wireless network into four stages, namely *data-driven*, *information-driven*, *knowledge-driven*, and *reasoning-driven*. Their primary contribution is a proposed framework for the final stage of wireless network, where reasoning capabilities are built-in. One central aspect of this framework is the designation of transmitter and receiver nodes as “teacher” and “apprentice” nodes. The two agents communicate using a *semantic language*. Another key feature is the notion of *context*, which plays a large role in their causal reasoning framework consisting of interventions and counterfactuals.

Overall, this framework is proposed to meet the needs of the final stage outlined above, i.e., it is reasoning-driven. However, such reasoning-driven communication will be largely dependent on knowledge representation, as an agent must have some knowledge to reason over in the first place. In other words, *knowledge-driven* communication is a vital prerequisite to reasoning-driven communication, as is shown in the stages of evolution given above. In the previous sections, we have developed different aspects of such a knowledge-driven

approach. Now, we connect our approach to this higher-level reasoning-driven framework. First, consider the definition for a semantic language.

Definition 8. (Semantic Language [118, Defn. 3]) A semantic language $\mathcal{L} = (X_i, Z_i)$, is a dictionary (from a data structure perspective) that maps the learnable data points $X_{l,i}$ to their corresponding semantic representation Z_i , based on the identified semantic content elements Y_i .

We take the idea of a concept (as in Definition 5) to be synonymous with the idea of a “semantic content element” in Definition 8. Moreover, the geometric underpinning of the CS model inherently provides the semantic representations of such content elements, which we can take to be the coordinates of the prototype point $\mathbf{z}_j$ for concept $\mathcal{C}_j$. Then we can define a CS-based semantic language.

Definition 9. (Conceptual Space-Based Semantic Language) $\mathcal{L} = (\mathbf{x}_j, \mathbf{z}_j)$ is a dictionary that maps the learnable data points $\mathbf{x}_j$ of concept $\mathcal{C}_j$ to the corresponding concept prototype $\mathbf{z}_j$.

Therefore, our proposed model of knowledge representation is fully compatible with this higher-level framework for reasoning-driven communication. More importantly, our approach can provide a formal foundation for the qualitatively-defined components of this framework, such as semantic content elements and semantic representations. In [118], it is stated that a semantic language should exhibit three qualities. We close this subsection by highlighting how the semantic language of Definition 9 meets these three criteria:

- *Minimalism:* By representing concepts within the raw data as coordinates within a CS, the amount of data transmitted to convey the semantics can be immensely reduced. For example, in our previous work we showed that effectively transmitting image semantics with CSs can reduce the required data rate by over 99% [121].
- *Generalizability:* As the CS representation truly sits at the semantic level, domains can be reused for different tasks and settings. For example, the color domain can be used to represent concepts involving color in any given task.

- *Efficiency*: The geometric representation of semantics provided by CSs allows for efficient computation of otherwise difficult operations. Take the example of semantic similarity, which has been a particularly challenging quantity to assess. One recent approach is to use a pre-trained DNN model to assess the similarity of sentences, e.g., using BERT [120]. Even without training, a network of this size can still require billions of operations per computation. With CSs, semantic similarity is captured as a simple distance function over interpretable semantic dimensions, from which meaning can be easily compared.

4.5.2 Communicating with Context

When first describing CSs, Gärdenfors points out the importance of context. He primarily describes two methods of expressing context, which are referred to as *salience* and *sensitivity*. Salience refers to the fact that in a given situation, some domains of a CS may be more important or prominent than others. To this point, Gärdenfors states “the relative weight of the domains depends on the *context* in which the concept is used” [31, p. 103]. On sensitivity, he states that “subjects can be trained to become sensitized to certain *areas* of a dimension so that the perceived length of the area is increased” [31, p. 104].

These two aspects of context can be mathematically realized by altering the semantic distortion function in (4.10). First, note that these two notions will alter the semantic distortion function in different ways. Salience seems to work *across* domains, while sensitivity alters the space *within* a single domain. Starting with salience, as initially proposed by Gärdenfors we can capture this idea with a set of *weights* corresponding to the domains,

$$\mathcal{W} = \{w_1, w_2, \dots, w_M\}, \quad \sum_{m=1}^M w_m = 1, \quad (4.22)$$

which alters the semantic distortion function in the follow way:

$$\delta(\mathbf{z}_1, \mathbf{z}_2 \mid \mathcal{W}) = \sum_{m=1}^M w_m \delta_m(\mathbf{z}_1, \mathbf{z}_2) \quad (4.23)$$

Given the communication context, greater distortion in highly-weighted domains will have a greater effect of the overall distortion. Additionally, it is easy to show that if the semantic distortion $\delta(\cdot)$ is a metric function, the salience-weighted semantic distortion $\delta(\cdot \mid \mathcal{W})$ is also a metric, and the results in the previous section still apply.

Furthermore, we can think of sensitivity as a transformation of a domain, i.e., some dimensions may be “stretched” and some may be “shrunk” to reflect the sensitivity of the agent in the given context. We define a set of transformations corresponding to each of the domains in the space,

$$\mathcal{T} = \{T_1, T_2, \dots, T_M\}, \quad (4.24)$$

such that

$$\tilde{\mathcal{D}}_m = T_m(\mathcal{D}_m), \quad \tilde{\delta}_m = \delta_m(\tilde{\mathcal{D}}_m). \quad (4.25)$$

Transformation T_m alters the geometry of the m th domain to match the sensitivity of the agent in the context, and thus defines a new function $\tilde{\delta}_m$ which maps the new domain to the non-negative real line. Substituting these new distortion functions into (4.23) we have

$$\delta(\mathbf{z}_1, \mathbf{z}_2 \mid \mathcal{W}, \mathcal{T}) = \sum_{m=1}^M w_m \tilde{\delta}_m(\mathbf{z}_1, \mathbf{z}_2), \quad (4.26)$$

which is the new context-dependent distortion function. By incorporating these different aspects, we can obtain a more realistic representation of meaning, and begin to capture some of the more nuanced aspects of meaning that can enable more intelligent SemCom.

4.5.3 Ambiguity, Redundancy, and Scalability

Two important concepts in the context of SemCom are semantic *ambiguity* and *redundancy*. Semantic ambiguity results from the fact that some data can be associated with multiple meanings. For example, the word “tree” in the context of a nature hike brings to mind an entirely different picture than that of a decision “tree” in a computer science context. Conversely, semantic redundancy refers to the scenario in which multiple unique data possess

a similar meaning, e.g., the words “begin” and “commence”. Semantic ambiguity can often be alleviated with the inclusion of contextual information. This is what is done in (4.26), where the contextual information is used to alter how meaning is represented within the CSs framework. Regarding the semantic redundancy, this issue is implicitly handled by the CSs framework in that data with similar meaning will be mapped to similar points within the semantic space. Thus, the elimination of semantic redundancy becomes of a problem of designing the semantic encoder to identify redundant syntactic data and map them accordingly to the semantic space. Our framework, therefore, provides natural ways of handling both semantic ambiguity and redundancy.

One of the primary challenges faced when implementing our proposed approach is the design of the CS itself. This is a non-trivial task that generally requires deep knowledge of the semantics related to the task at hand. In our previous work [121] and in the experiments in the following section, we hand-craft CSs to achieve SemCom. This approach to CS design lacks scalability, and thus more efficient techniques will be required to enable practical use of this framework. One possible path toward addressing this problem is to utilize existing techniques in the field of *representation learning* [122]. In this context, the dimensions of the CS could be considered as features to be learned, where the higher-level structures (domains, concepts, etc.) are built on top of the learned space. This critical challenge is addressed in Chapters 5 and 6.

4.6 Experimental Results

To demonstrate our proposed approach, we simulate a metaverse-inspired communication problem. Lately, the idea of the metaverse has received great attention due to its potential to enhance digital experiences. Metaverse applications aim to provide immersive online experiences by utilizing technologies such as virtual reality and augmented reality [123, 124]. To achieve true immersion, these applications potentially require massive amounts of data transmission. These intensive communication requirements, coupled with the expected growth of the metaverse [125], make metaverse-inspired communication a prime candidate

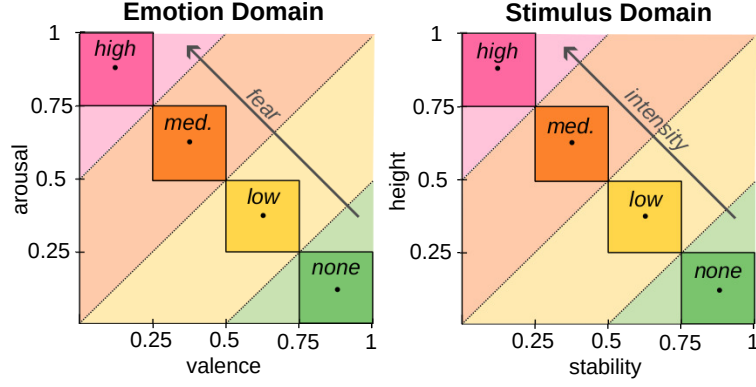


Figure 4.3: CS for using VRET to treat acrophobia. Properties, their prototype points and the Voronoi tessellation are included.

for the benefits of SemCom.

4.6.1 Problem Definition and System Description

In our experiments, we look at the problem of VRET, which is just one of the many examples in which the metaverse can be utilized in the context of healthcare [119]. VRET can be used to treat phobias by exposing the patient to fear-inducing stimuli in the virtual environment while being guided by a therapist, to gradually reduce fear of the specific stimulus [126]. One of the most common phobias is the fear of heights (acrophobia) [127]. Thus, we focus on this scenario and aim to use the tools developed in the previous sections to design and analyze a SemCom system for a VRET application within the metaverse. Specifically, we look at the end-end patient-to-therapist communication link.

The first step in our design is to build a CS model to capture the relevant semantics. These relevant semantics are often tied to the overall goal of communication, and in the patient-to-therapist scenario, the goal is to provide a clear picture of the patients current emotional state to the therapist, as well as the intensity of the current stimulus in the virtual reality environment. Thus, we define two domains, shown in Figure 4.3; namely, the emotion and stimulus domains.

We utilize valence and arousal dimensions [128] to construct the emotion domain. Valence refers to the degree to which a given emotion is positive or negative, while arousal refers to

Table 4.1: Concepts and prototype points ($\mathbf{z}_j$).

<i>Phobia Level</i>	Valence	Arousal	Height	Stability
Mild	0.375	0.625	0.875	0.125
Moderate	0.250	0.750	0.500	0.500
Extreme	0.125	0.875	0.125	0.875

the intensity or level of activation of an emotion. Fear represents a low-valence and high-arousal emotion. We also adopt the different ranges used in [128] for fear-level classification to define properties within this domain, also shown in Figure 4.3. For the stimulus domain, with acrophobia in mind, we identify the dimensions of *height* and *stability* to characterize the *intensity* of a stimulus. For example, a high-intensity stimulus would be climbing a tall ladder, while a low-intensity stimulus would be walking on a beach. For simplicity, properties are defined as degrees of intensity, analogous to the fear properties of the emotion domain.

As our CS has two domains with linear dimensions, we can define the semantic distortion as the sum of the Euclidean distances in both domains, i.e., for $\mathbf{z} = (v \ a \ h \ s)$,

$$\delta(\mathbf{z}, \hat{\mathbf{z}}) = \left\| \begin{pmatrix} v - \hat{v} \\ a - \hat{a} \end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} h - \hat{h} \\ s - \hat{s} \end{pmatrix} \right\|_2 \quad (4.27)$$

Concepts are formed by taking the Cartesian product of convex regions within the two domains, and these concepts represent *phobia levels*. For example, “mild phobia” is a concept obtained by combining “mild fear” with “extreme intensity”. Similarly, a concept of “extreme phobia” can be realized through a combination of “high fear” and “no intensity”. We define three concepts and their prototype points, which are given in Table 4.1. We limit ourselves to concepts involving medium and high fear properties, due to these being by far the most common of our defined emotional properties present in the Aff-Wild2 dataset used for training [128–130].

As described in section 4.3, our SemCom goal is the accurate communication of these concepts. In our experiments, we simulate four systems to achieve this goal.

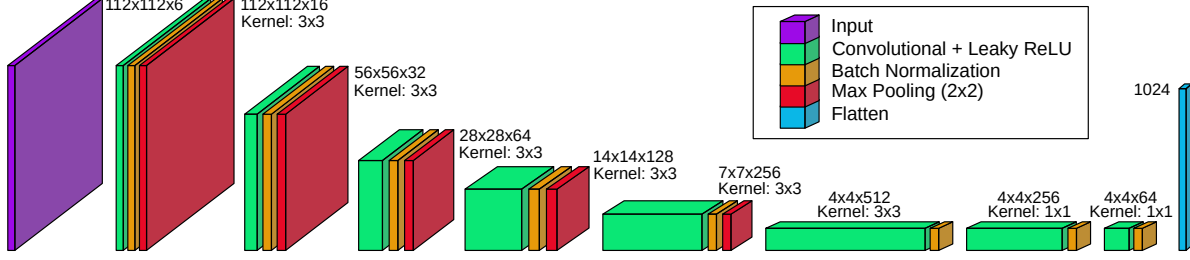


Figure 4.4: Visualization of the BaseCNN architecture used in the practical systems.

Theoretical Semantic System

First, we simulate SemCom with a theoretical semantic encoder to examine the system performance with arbitrary encoder distortion. Here we model the dimensional error introduced by the semantic encoder as a multivariate normal random vector, i.e.,

$$\mathbf{z} = \mathbf{z}_j^* + \mathbf{n}, \quad (4.28)$$

where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I})$ and $\mathbf{I}$ denotes the identity matrix. Note that we can tune the parameter σ_e to simulate varying levels of quality of the semantic encoder. The semantic representation $\mathbf{z}$ is then quantized, modulated, and transmitted over the channel. At the receiver, demodulation and channel decoding is performed, and a minimum-distance semantic decoder is employed to make a decision on the transmitted concept.

Practical Systems

We simulate three practical systems to demonstrate the performance of our proposed approach. The first represents the proposed approach and utilizes the SCCS framework. The first baseline utilizes a technique inspired by the joint source channel coding technique for SemCom [131], and the second uses traditional communication system with no SemCom. Each system utilizes a CNN architecture that is slightly modified from that of [132], which is originally based on the YOLO architecture [133]. This architecture (termed BaseCNN) is shown in Figure 4.4, and other details pertaining to the models are provided in Table 4.2.

To train the models, we utilized two datasets, each corresponding to one of the domains

Table 4.2: Practical systems: models and training details.

	SCCS	Semantic Baseline	Non-Semantic Baseline
<i>Architecture</i>	BaseCNN Dense(4), sigmoid activation	BaseCNN Dense(64), tanh activation Binarization Layer 16-QAM Layer AWGN Layer Dense (3), softmax activation	BaseCNN Dense(3), softmax activation
<i>Loss Function</i>	Equation (4.27)	Categorical Cross Entropy	
<i>Optimizer</i>	Adam		
<i>Learning Rate</i>	0.001		
<i>Minibatch Size</i>	64		
<i>Minibatches/Epoch</i>	500		
<i>Total Epochs</i>	100		
<i>Dropout Rate</i>	0.3		

of the CS. The first is the Aff-Wild2 dataset [129, 130], which consists of 564 videos of around 2.8M frames with 554 subjects (326 of which are male and 228 female), all annotated with valence/arousal labels. For the stimulus domain, we created a dataset of 43 videos of around 500k frames of various stimuli corresponding to different levels of intensity and assigned each frame a height/stability label. Data samples were constructed by sampling a frame from each dataset, resizing each to a $112 \times 112 \times 3$ RGB image, and combining the resulting images to form a $112 \times 112 \times 6$ array which serves as the input to each CNN. A total of 450k of these combined images were used for the training dataset, while 50k were used for the validation dataset.

When training the conceptual-space based model, the label for a given input is the semantic representation $\mathbf{z}$. Thus, the label for a given data sample consists of a four-element vector describing the position of the data sample in the CS. In essence, our aim is to teach the semantic encoder the geometry of the space and its relation to the raw data, thus providing it with *knowledge* of the concepts therein. For the other two models, the labels are one-hot encoded vectors specifying the present concept. For training the end-end semantic model, a uniformly random SNR ranging from -20dB to 20dB was selected to generate the added noise for each sample. For each model, dropout was applied during training after each convolutional block to mitigate overfitting. The models were created and trained using the Keras API within the Tensorflow Python package.

After the models were trained, the systems were simulated as shown in Figure 4.5. The proposed system transmits CS representations traditionally and uses a minimum-distance

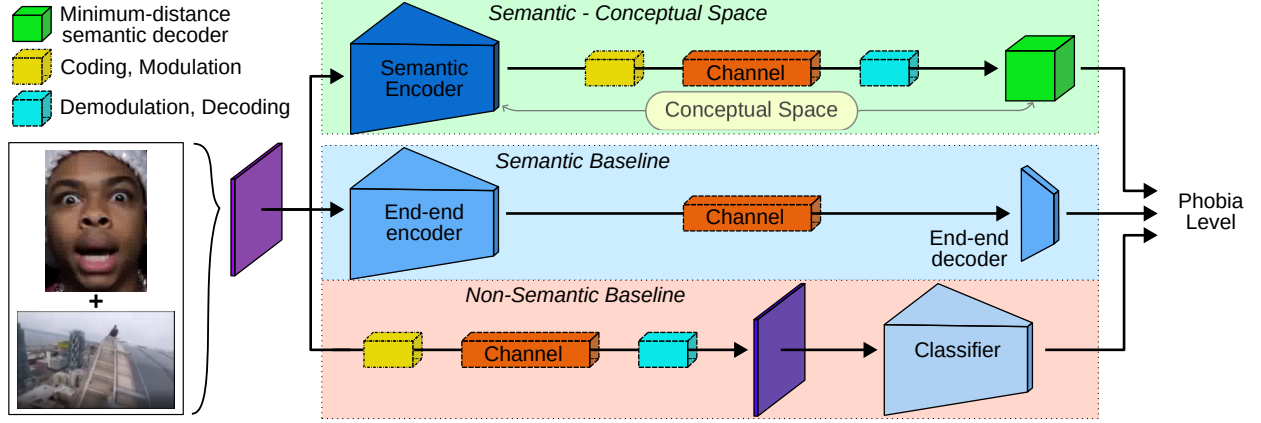


Figure 4.5: The three practical communication systems for VRET.

decoder. Given the received CS representation, this decoder uses knowledge of the geometry to get a decision on the concept. The encoder of the end-end semantic system generates complex symbols directly and a dense layer is used at the receiver to get the decision directly from the received symbols. The non-semantic system transmits both of the images over the channel and uses a classifier at the receiver to decide the phobia level.

4.6.2 Traditional Communication Components

For each of the systems, we simulate various aspects of the traditional portion of the system. Many future metaverse applications will likely utilize WiFi, and thus in our experiments we simulate the IEEE 802.11 standard [134]. We examine communication under BPSK, 16-QAM, and 256-QAM modulation. For the CS-based semantic system, we employ rate 1/2 convolutional coding at the transmitter and Viterbi decoder at the receiver. Moreover, we perform experiments for both AWGN and Rician fading channel models.

4.6.3 Results and Discussion

Monte Carlo simulations are used to demonstrate the performance of the systems described above. Here we describe these experiments and the following results, and provide some discussion as to their implications.

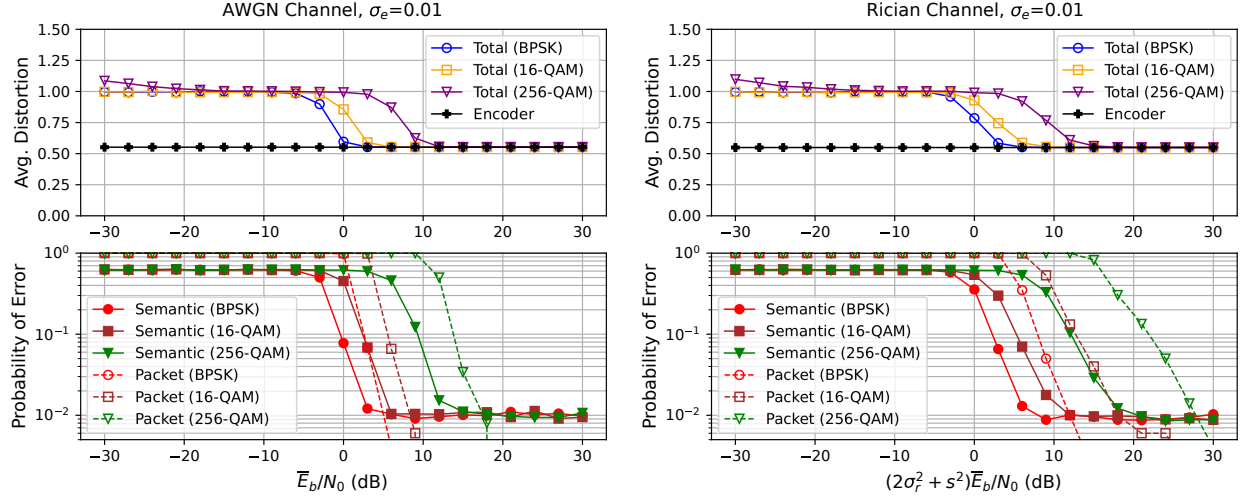


Figure 4.6: Plots of average semantic distortion vs. SNR (top) and probability of error vs. SNR (bottom) results for the proposed system with a theoretical semantic encoder under the AWGN channel (left) and Rician fading channel (right).

Theoretical Encoder

First, we perform experiments using the SCCS system with a theoretical encoder, the results of which are shown in Figure 4.6. The top-left plot shows the overall semantic distortion $\delta(\mathbf{z}_{j^*}, \hat{\mathbf{z}})$ and the encoder distortion $\delta(\mathbf{z}_{j^*}, \mathbf{z})$ as functions of the SNR for an encoder standard deviation $\sigma_e = 0.01$ under the AWGN channel. In each case, the overall semantic distortion becomes limited by the performance of the semantic encoder as the channel quality improves. Therefore, the semantic encoder is the key factor in determining the best-case performance of the overall SemCom system. Improving the performance of the semantic encoder will lower the distortion floor of the overall system.

The bottom-left plot of Figure 4.6 confirms this; there, the probability of semantic error $P(\hat{j} \neq j^*)$ is plotted against SNR, as well as the probability of packet error. Recall from Definition 7 that a semantic error is defined as an incorrectly decoded concept, analogous to the notion of a syntactic error representing an incorrectly decoded symbol. In our simulations, a packet is composed of 160 bytes corresponding to 20 semantic representations $\mathbf{z}$ each consisting of a vector of 4 8-bit floating point values, with rate 1/2 channel coding. We observe that the probability of semantic error reaches a hard limit at precisely the SNR

where the overall distortion in the top-left plot reaches its floor. As such, this error floor can be reduced by improving the quality of the semantic encoder. This plot also illustrates the intuition that a syntactic error does not necessarily induce a semantic error. For example at $E_b/N_0 = 0\text{dB}$ we see that the probability of packet error is nearly 1 for all modulation types, but for BPSK modulation the probability of semantic error is around 0.08. Thus, we are able to achieve good semantic performance despite poor technical performance, illustrating the potential robustness of SemCom systems to traditional bit errors.

The right-hand plots in Figure 4.6 provide similar results for the Rician channel model with fading parameter $K = 6\text{dB}$. We see that the semantic distortion curves are rather similar to those for the AWGN channel. Moreover, the probability of semantic error curves are also relatively similar to those for the AWGN channel, despite the greatly increased probability of packet error. These results indicate that the proposed approach can provide even greater benefits with respect to robustness in the presence of channel fading.

Practical Systems

Next we examine the performance of the proposed SCCS system compared to the baseline semantic and non-semantic systems, and the results are shown in Figure 4.7. All of the results in this subsection were obtained for the AWGN channel.

The curves marked by circles show the probability of semantic error for the proposed approach and the bound obtained from Lemma 1. We see similar behavior as the system with a theoretical semantic encoder; specifically, the performance improves with the channel conditions up to a certain point, at which it becomes limited by imperfections of the semantic encoder. We also observe that the upper bound provided by Lemma 1 indeed holds as an upper bound for the semantic performance, though it does not appear to be a very tight bound.

One reason for this loose bound lies in the fact that we are measuring semantic distortion with respect to concept prototypes. Therefore, even if the semantic encoder is perfect in mapping a given input to its true representation $\mathbf{z}$, this point may not be close to any

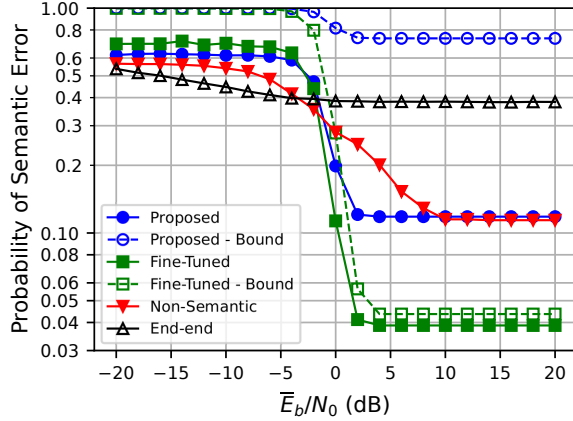


Figure 4.7: Probability of semantic error for the proposed system (filled circles) and a fine-tuned version (filled squares), baseline end-end semantic system (open triangles) and baseline non-semantic system (filled triangles), plotted against SNR.

prototypes; in other words, some data are *conceptually ambiguous* (which is different from *semantic ambiguity* that was discussed in section 4.5.3). Being relatively unrelated to any of the concepts of interest, these points induce great uncertainty at the receiver. Intuitively, we should observe improved performance and a tighter bound by eliminating this conceptual ambiguity. To test this hypothesis, we created a new dataset by randomly sampling points from the initial dataset that lie within the spheres centered at the prototype points with radius given by (4.16). This new dataset was then used to fine-tune the general CNN semantic encoder and to test the performance of the system with this fine-tuned encoder. The results are given in Figure 4.7, and are denoted by the curves marked with squares. These results confirm our intuition, as the best-case probability of error drops from around 0.11 to about 0.04. We also observe that the upper bounds on the semantic performance are indeed much tighter.

The curve marked by filled triangles in Figure 4.7 displays the results for the baseline non-semantic system, while the unfilled triangles denote the performance of the baseline end-end semantic system. First, we observe that the end-end baseline performs the best out of the three for low SNR, which agrees with previous results for this kind of approach [4]. However, the end-end baseline performs the worst at high SNR values. This is most likely due to the

inclusion of the non-differentiable binarization, 16-QAM, and AWGN layers which introduce non-smooth behavior, impairing the gradient-based updates of the backpropagation algorithm during training. Next, we see that the non-semantic baseline performs similarly to the proposed system at extreme SNRs, with better performance at SNR values under 0dB and worse performance at SNR values greater than 0dB.

Finally, it is important to note the efficiency gains of the semantic systems to the non-semantic system. Each semantic representation consists of 4 8-bit values, coded at a rate of $1/2$ for a total of 64 bits (the same number of bits is used in the end-end system). In the non-semantic system, each inference is carried out over $2 \times 112 \times 112 \times 3$, for a total of 75,264 pixels. Each pixel uses 8 bits, resulting in 602,112 bits transmitted per inference. Thus, the semantic systems reduce the overall rate by over 99.9%, with the proposed system achieving the best performance. These results demonstrate that the proposed system can achieve reliable SemCom with a massive reduction in rate.

4.7 Chapter Summary

In this chapter, we have greatly expanded both theoretical and practical aspects of a SemCom system with knowledge representation based on the theory of CSs. We provided formal definitions of key aspects, defined the notions of semantic distortion and semantic error, and derived error bounds that follow from these definitions. We showed how this theory can serve as the underlying foundation for a truly intelligent reasoning-driven system, and how important notions such as context can be easily incorporated into the framework. To illustrate some of the key benefits of our approach, we simulated a VRET application utilizing the SCCS framework, and demonstrated robust semantic performance with more than 99.9% percent reduction in rate as compared to a more traditional system. These results reveal some important insights, such as the importance of the semantic encoder and the CS design in the overall performance of the SCCS system.

Chapter 5

Learning Conceptual Spaces

One of the primary challenges by the SCCS framework is obtaining the CS model. In the previous chapters, these models were hand-crafted using domain knowledge, after which a ML model was trained in a supervised manner to map raw data to the CS. This approach has two major drawbacks. First, the hand-crafted design of the CS is not scalable and depends on the knowledge of the designer. Second, ground-truth CS coordinate labels are required to train the semantic encoder, and obtaining this data can be difficult and costly. To address these shortcomings, in this chapter we propose a method of learning the domains of a CS in a partially supervised manner using a modified neural architecture based on the concept of an AE. This approach removes the hand-crafted aspect and automates the CS design, requiring only high-level property labels for training, rather than fine-grained CS coordinate labels.

5.1 Related Work

Closely related to the problem of learning a CS model is the problem of *feature learning* or *representation learning*, where the goal is to extract meaningful patterns from data, usually in a lower-dimensional space. Some approaches to this problem include probabilistic models, manifold learning, and AEs [122]. As a simple example, the well-known principal component

analysis technique can be viewed as a particular example of all three of these methods. More information on these methods of representation learning can be found in [122]. However, these approaches are not directly applicable to the problem CS learning, as the learned dimensions often lack interpretability and can be intractable.

5.1.1 Chapter Contributions

In this chapter we introduce an AE-based technique for CS learning. The key idea is to utilize semantic regularization based on the geometric structure of the learned representations and the semantic similarity derived from that structure to engender the key properties of a CS in the learned space. We focus on learning a single domain of a CS and hope to build on this approach in future work. In short, the primary contributions of this chapter include:

- The introduction of a novel approach to learning a domain of a CS model based on the AE architecture. The modified architecture includes a maximum-similarity property classifier module to add semantic regularization.
- An iterative algorithm to learn the semantic properties and the model mapping raw data to the semantic domain.
- Experimental results using the MNIST and CelebA datasets to demonstrate that the proposed framework learns semantically consistent property regions and interpretable quality dimensions.

5.2 Problem Definition

In this chapter we focus on the problem of learning a single domain of a CS model, which we refer to as *domain learning*. To formally define the problem, we begin by recalling the definitions for quality dimension, domain, and property first presented in Chapter 4.

Definition 1. (Quality Dimension [reproduced from Section 4.3.1]) A quality dimension is a set of scalar values quantifying some quality of an idea or object. A key point here is that

these dimensions should be interpretable; typical ML-generated features in many popular approaches to SemCom do not fit this criterion. We denote a quality dimension by $\mathcal{Q}$, where specific values along a quality dimension are denoted by q , such that $q \in \mathcal{Q}$.

Definition 2. (Domain [reproduced from Section 4.3.1]) A domain $\mathcal{D}$ is a set that is constructed from the Cartesian product of integral quality dimensions, i.e., $\mathcal{D} = \times_{k=1}^K \mathcal{Q}_k$. A point within a domain is specified by a column vector of quality values $\mathbf{d} = (q_1 \ q_2 \ \cdots \ q_K)^\top$.

Definition 3. (Property [reproduced from Section 4.3.1]) A property $\mathcal{P}$ of a single domain $\mathcal{D}$ is defined as a convex subset of that domain, i.e., $\mathcal{P} \subset \mathcal{D}$ where, for $\lambda \in [0, 1]$ and any $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{P}$, $\lambda \mathbf{p}_1 + (1 - \lambda) \mathbf{p}_2 \in \mathcal{P}$.

Quality dimensions are the building blocks of CS models. Integral quality dimensions form a domain, where two qualities are said to be *integral* if a value on one dimension cannot be assigned to an object without giving it a value on the other, e.g., hue and brightness of color [31]. Properties are then defined as regions within a domain, where the convexity requirement ensures that if two points represent a given property, all points in between also represent that property.

As a geometric entity, it is natural to think of a semantic distance or distortion metric on a domain. Here we consider a particular case of the general semantic distortion presented in [135] with only a single domain.

Definition 10. (Single-domain semantic distortion) Semantic distortion is a function δ mapping two points in the domain $\mathcal{D}$ to the non-negative real line, i.e., $\delta : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_+$.

Moreover, we define the notion of semantic similarity, which acts on the semantic distortion to yield a measure of similarity between two points in the CS. In [31], it is suggested that a Gaussian function could be used to transform semantic distance to similarity; we adopt this function here.

Definition 11. (Semantic similarity) Semantic similarity maps semantic distortion to the interval $(0,1]$, with 0 representing total dissimilarity and 1 representing identity. Formally,

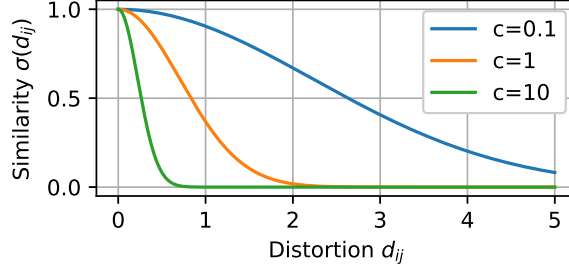


Figure 5.1: Semantic similarity vs. semantic distortion.

we have $\sigma : \mathbb{R}_+ \rightarrow (0, 1]$ such that,

$$\sigma(d_{ij}) = e^{-cd_{ij}^2}, \quad (5.1)$$

where d_{ij} is the distortion between two points $\mathbf{q}_i$ and $\mathbf{q}_j$ and c is a tunable parameter.

We see that the similarity is 1 when d_{ij} is 0 and approaches 0 as d_{ij} approaches infinity. Similarity will remain approximately 1 for small distortion values, begin to quickly decrease as distortion continues to increase, and eventually become approximately 0 for large distortion values. The parameter c can be used to “shrink” or “stretch” this function, such that the similarity function is more or less sensitive to small changes in distortion, as shown in Figure 5.1.

At the outset, we assume that we are given a dataset with two components. First, the dataset contains samples of the raw data to be communicated, i.e., the input to the semantic encoder in a typical SemCom setup. Second, it contains *semantic property* labels for each data sample. Note that these labels contain no information about the property regions as defined in Definition 3, and only specify *which* property the data possesses. Further, in this work we assume that the quality dimensions to be learned are continuous intervals that lie in Euclidean space such that $\mathcal{Q} \subset \mathbb{R}$ and $\mathcal{D} \subset \mathbb{R}^N$ where N is the number of dimensions in the domain. As such, we take the semantic distortion function to be the Euclidean distance function, i.e.,

$$\delta(\mathbf{q}_i, \mathbf{q}_j) = \|\mathbf{q}_i - \mathbf{q}_j\|_2 = \sqrt{\sum_{n=1}^N \left(q_i^{(n)} - q_j^{(n)}\right)^2}. \quad (5.2)$$

Given these data and assumptions, our objective is to employ ML to learn a CS domain describing the semantics of the data. Thus, the quality dimensions will be characterized by the dimensions that the network learns to map the data to. To validate the quality dimensions of the CS domain, we require them to be interpretable. Furthermore, we require that semantically similar properties be located closer in the space than semantically dissimilar properties. Instead of learning the property regions themselves, we can take these regions to be the corresponding Voronoi regions formed using the semantic distortion function and a set of P property *prototypes* $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^P$, where $\mathbf{p} \in \mathcal{D}$ [31, 135]. Thus, specifying the property regions is equivalent to specifying the coordinates of each prototype point in the domain, and all we need to learn are these prototype points. To summarize, we have three requirements for the result of the domain learning process:

- R1. A model mapping data to interpretable dimensions.
- R2. A set of distinct points defining the property regions.
- R3. Semantically consistent similarities between properties.

5.3 Proposed Approach

To address these challenges, we propose an AE-based approach as shown in Figure 5.2. The architecture contains three primary modules: the encoder, the decoder, and the classifier. The encoder and decoder modules are the same as in a traditional AE. The central innovation is the addition of the property classifier module, which provides a kind of *semantic regularization* to the network while learning. The details of each module are outlined below.

5.3.1 Encoder Module

The encoder takes the input and reduces the dimensionality to produce an output vector $\mathbf{q}$, which is a lower-dimensional representation of the input data $\mathbf{x}$. This module consists of traditional neural network layers, which can vary depending on the input data type. For

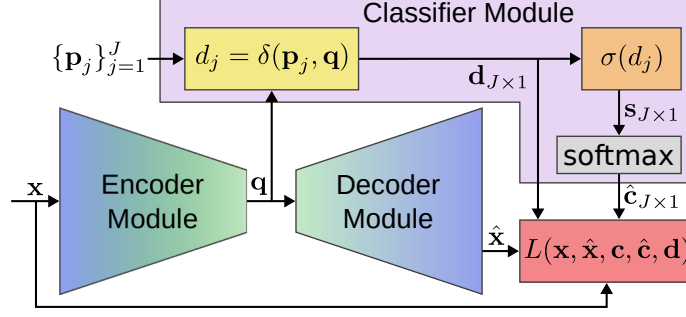


Figure 5.2: Architecture of the domain learning network.

example, the encoder can comprise many dense layers for typical vector-shaped input data, or convolution layers for three-dimensional image data. At the output of the encoder, a dense layer with no activation is included to reduce the features to the domain dimensionality.

5.3.2 Decoder Module

The task of the decoder module is to reconstruct the input data using the output of the encoder module. In the domain learning task, the decoder module is responsible for ensuring that similar data samples are mapped closer together within the CS domain. The intuition is that if two inputs $\mathbf{x}$, $\mathbf{x}'$ are similar, the decoder will require similar representations $\mathbf{q}$, $\mathbf{q}'$ in order to recover the original inputs $\mathbf{x}$ and $\mathbf{x}'$. Structurally, the decoder module is built to mirror the encoder module, with layers that expand the dimensions reduced by layers of the encoder. To measure the quality of the reconstructed output $\hat{\mathbf{x}}$, we define the reconstruction loss term ℓ_r as the mean squared difference between the input and reconstructed data,

$$\ell_r(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{M} \sum_{m=1}^M (x_m - \hat{x}_m)^2. \quad (5.3)$$

5.3.3 Classifier Module

The classifier module is the key element of the domain learning architecture. This module takes as inputs the encoded representation $\mathbf{q}$ and a list of prototypes $\{\mathbf{p}_j\}_{j=1}^J$ that describe the prototypical properties within the domain. First, the semantic distortion between $\mathbf{q}$ and each prototype is computed, yielding a $J \times 1$ vector $\mathbf{d}$ of distortion values. The semantic

similarity is then computed for each of the distortion values, the result of which is another $J \times 1$ vector $\mathbf{s}$. Finally, a softmax operation is performed on $\mathbf{s}$ to get a classification vector $\hat{\mathbf{c}}$, which is passed to the loss function. Note that the classifier module does not contain any trainable layers.

This module addresses the scalability problem by allowing us to learn a domain with only high-level property labels, rather than fine-grained coordinate labels which are more difficult and costly to obtain. It acts as a *maximum-similarity decoder* mapping domain coordinates to the property within the domain that is the most similar. As the properties basically encode the difference in meanings between the input data samples, this module ensures that the encoded representations are separated in the space according to their different meanings. Furthermore, unlike a basic classifier network, the output classification is entirely dependent on the similarity between the property prototypes and the domain representation. This ensures that notions of semantic distortion and semantic similarity are well-represented within the learned domain.

There are two loss terms associated with the classifier module. First, the classification loss seeks to train the network to produce classification vectors $\hat{\mathbf{c}}$ which are close to the true $\mathbf{c}$ representing the correct properties, which are one-hot encoded vectors. Categorical cross-entropy is used for this loss term,

$$\ell_c(\mathbf{c}, \hat{\mathbf{c}}) = - \sum_{j=1}^J c_j \log(\hat{c}_j). \quad (5.4)$$

Second, we need to ensure that the distances between representations in the space are reasonable, i.e., the similarities take on meaningful values. For example, in Figure 1, for the $c = 1$ curve, any distortions greater than approximately 2.25 will yield a similarity of essentially zero. Thus, we include a distortion-regularization term which penalizes large distortions between an encoded representation and the prototypes of the properties to which it does *not* belong. This ensures that the space does not “blow up” and that the regions

remain relatively close together. We achieve this with the function

$$\ell_d(\mathbf{d}) = \frac{1}{J-1} \sum_{j \neq j^*} \frac{1}{k_1} d_j^{k_2}, \quad (5.5)$$

where j^* denotes the index of the true property, and k_1 and k_2 are hyperparameters that can be used to adjust the range of distortion values that are penalized.

5.3.4 Putting it Together: Domain Learning Algorithm

Combining equations (5.3)-(5.5), the loss function used to train the domain learning architecture is defined as

$$L(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{c}, \hat{\mathbf{c}}, \mathbf{d}) = \alpha \ell_r(\mathbf{x}, \hat{\mathbf{x}}) + \beta \ell_c(\mathbf{c}, \hat{\mathbf{c}}) + \lambda \ell_d(\mathbf{d}) \quad (5.6)$$

where α, β and λ are hyperparameters that can be tuned to balance the terms of the loss function. This loss is used with backpropagation to tune the parameters of the encoder and decoder while learning the domain. However, recall that computing $\mathbf{d}$ requires the set of prototypes $\{\mathbf{p}_j\}_{j=1}^J$, which are one of the aspects of the domain which we are trying to learn and do not have *a priori* knowledge of at the start. Therefore, we propose an iterative algorithm to learn both the dimensions of the domain and the property prototypes in a joint manner; this algorithm is presented in Algorithm 1.

First, the number of batches M and batch size B used for training on each step of the algorithm is set. Then the encoder and decoder are trained for one step with the parameter β set to 0; this is equivalent to training a vanilla AE on MB samples of the input data with some distortion regularization. For this first step, prototypes are set to the zero vector $\mathbf{0}$ to keep the AE-learned features centered at the origin. To initialize the prototype points, M batches are then passed through the encoder to obtain MB encoded representations. The representations that correspond to property j are then averaged to obtain the initial prototype point for that property. With the initial properties set, the network is then trained over another M batches using the full loss function, after which the prototypes are updated

Algorithm 1 Domain Learning

Require: $\alpha, \beta, \lambda, \mu, M, B, J, \{\mathbf{p}_j\}_{j=1}^J = \mathbf{0}$

- 1: **for** each epoch i in $\{1, 2, \dots, E\}$ **do**
- 2: **for** each batch m in M random batches **do**
- 3: **if** i is 1 **then**
- 4: $L \leftarrow \frac{1}{B} \sum_{b=1}^B \alpha \ell_r(\mathbf{x}_b, \hat{\mathbf{x}}_b) + \lambda \ell_d(\mathbf{d}_b)$
- 5: **else**
- 6: $L \leftarrow \frac{1}{B} \sum_{b=1}^B \alpha \ell_r(\mathbf{x}_b, \hat{\mathbf{x}}_b) + \beta \ell_c(\mathbf{c}_b, \hat{\mathbf{c}}_b) + \lambda \ell_d(\mathbf{d}_b)$
- 7: **end if**
- 8: Perform batch update with backpropagation
- 9: **end for**
- 10: Encode MB random samples to obtain $\{\mathbf{q}\}$
- 11: **for** each property j in $\{1, 2, \dots, J\}$ **do**
- 12: $\bar{\mathbf{q}}_j \leftarrow \text{mean}(\{\mathbf{q} : \mathbf{q} \text{ possesses property } j\})$
- 13: $\mathbf{p}_j^{(i)} \leftarrow \bar{\mathbf{q}}_j$ if $i = 1$, else $\mu \mathbf{p}_j^{(i-1)} + (1 - \mu) \bar{\mathbf{q}}_j$
- 14: **end for**
- 15: **end for**
- 16: **return** Encoder, decoder and prototypes $\{\mathbf{p}_j\}_{j=1}^J$

in the same manner that they were initialized, with the one difference that a parameter $\mu \in [0, 1]$ is used to “mix” the old prototypes with the new ones. This process repeats for a fixed number of epochs, denoted by E .

5.4 Experimental Results

We use image data in our experiments, as images contain semantics that are easy to interpret and have well-established architectures for encoding and decoding. Note that the proposed approach is not exclusive to images, and other modalities can be used with an appropriate model architecture.

The architecture of the encoder and decoder is shown in Figure 5.3. The encoder uses four convolutional layers, each followed by batch normalization and ReLU activation. The first convolutional layer has F filters, a kernel size of 7×7 and a stride of 2, halving the input dimensions. The second layer has F filters with a 5×5 kernel and stride of one. The third and fourth layers both use $2F$ filters with 3×3 kernels, while the third layer uses a stride length of 2. The resulting tensor is reshaped to a single vector, which is then passed

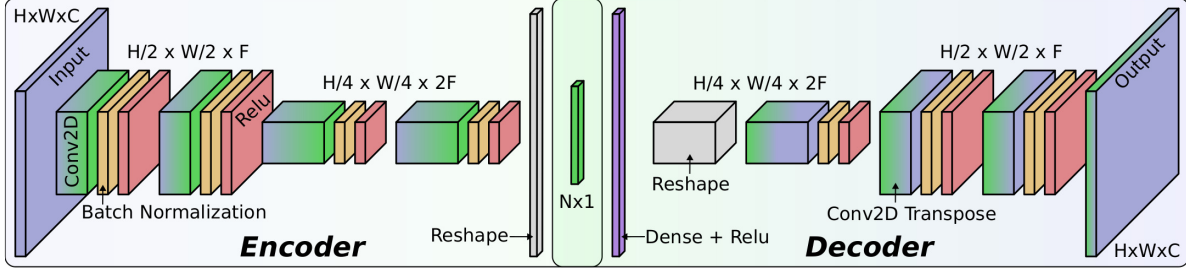


Figure 5.3: Architecture of the encoder and decoder used in the experiments.

through a dense layer with no activation to obtain the encoded domain representation. The decoder is a mirror image of the encoder, using transposed convolutional layers to “undo” the convolutional layers of the input. Experiments are performed using Tensorflow, and the Adam optimizer with a learning rate of 0.0002 is used to train the networks. For each experiment, the batch size B is set to 32, the number of batches M used to train each epoch is 250, and the number of epochs is 75.¹

5.4.1 Experiment 1: MNIST

We use the MNIST dataset as a starting point. The MNIST dataset contains 70,000 28x28x1 black-and-white images of handwritten digits from 0-9, split into 60,000 training images and 10,000 test images [136]. Thus, we have $H = 28$, $W = 28$, and $C = 1$. As the MNIST dataset is relatively simple, we choose $F = 16$ for encoder/decoder filter parameter. For the loss function hyperparameters, we choose $\alpha = 5$, $\beta = 1$ and $\lambda = 1$ so that the three loss terms take on a similar range of values during training. The prototype mixing parameter is chosen to be $\mu = 0.75$ to encourage smooth updates. For the scaling parameters in equation 5.5, we use $k_1 = 50$ and $k_2 = 5$.

We set the dimension of the encoded representations to $N = 2$. In addition to the domain learning model, we also train a basic classifier (encoder only, with a single dense classifier head layer) and a basic AE (encoder and decoder only). Figure 5.4 shows the learned feature spaces for the three approaches. In Figure 5.4a, we see that the classifier learns to map the different digits to distinct regions, but the semantic similarity is not captured, e.g., 4 and

¹Code implementing the domain learning approach is available at <https://github.com/dw-610/CSLearn>. See Appendix A for more details.

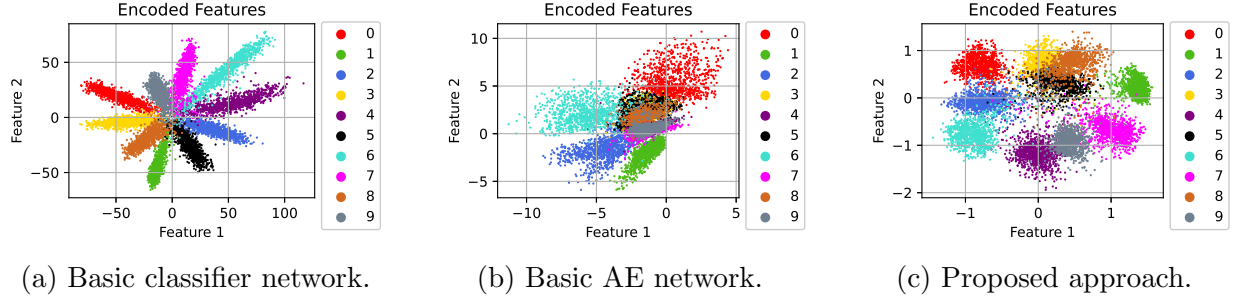


Figure 5.4: The learned representations of the validation samples of the MNIST dataset for (a) a basic classifier (b) a basic AE and (c) the proposed domain learning framework.

9 are very similar, but their regions are separated by 6 which is not similar to either. For the AE features shown in Figure 5.4b, the similarity is captured, but the properties are not distinct, e.g., 4, 7, and 9 are all mapped to approximately the same region of the space. In Figure 5.4c, we see that similar properties are indeed mapped closer to each other (e.g., 4 and 9, and 3 and 5) and that each property has a distinct region within the space.

These results show that the proposed approach is able to satisfy R2 and R3. However, the axes are not very interpretable. This is due to the nature of the input data; what numerically quantifiable quality makes a 4 similar to a 9 but different than a 3? What is “in between” a 4 and a 6? In order to learn continuous dimensions about which the semantics of the data vary, the data must share such a continuous quality.

5.4.2 Experiment 2: CelebA

Next, we utilize the CelebA dataset for domain learning. The dataset contains approximately 200,000 color images of celebrity faces [137], which we resize to 112x112x3. The labels contain 40 binary attribute annotations, such as male/female or “has blonde hair”/“does not have blonde hair”. We select two attributes to form our property labels, namely, male/female and smiling/not smiling, due to the attributes being evenly represented (in contrast to “has gray hair”, which has far more negative samples than positive). In this experiment, we increase the filter parameter to $F = 32$, set $\alpha = 1, \beta = 2, \lambda = 1$, and $\mu = 0.75$, while k_1 and k_2 are the same.

Figure 5.5 shows the learned features of the “face” domain, where we again choose $N = 2$.

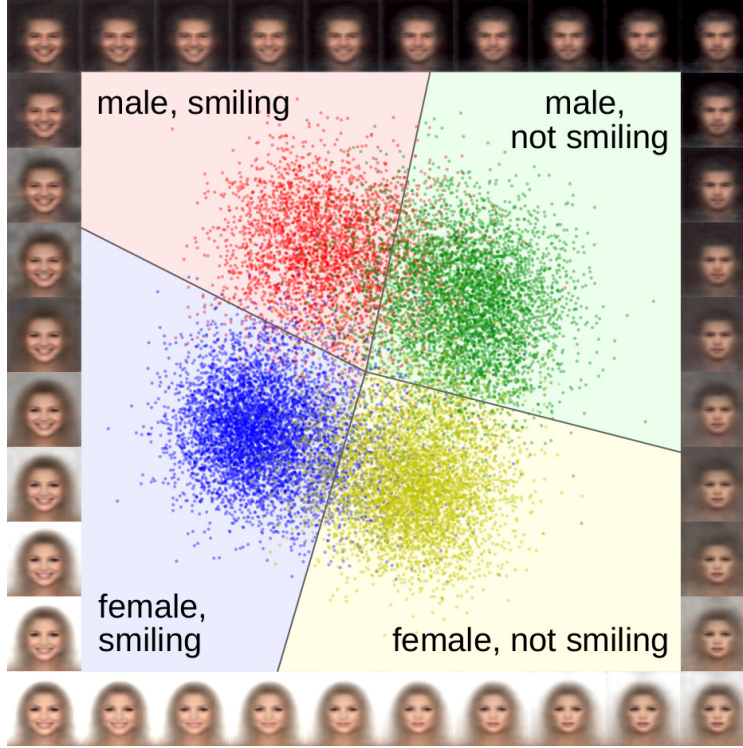


Figure 5.5: The learned representations of the validation samples of the CelebA dataset, and decoded representations at the edge of the domain. The domain encodes smiling/not smiling primarily along the x-axis and female/male primarily along the y-axis.

We see that as in the case of MNIST, the model learns property regions that are both distinct and accurately characterize the semantic similarity between properties. For example, the center of the “female, not smiling” property region is closer to the centers of the “male, not smiling” and “female, smiling” regions than that of the “male, smiling” region. This again demonstrates how the proposed framework satisfies R2 and R3.

Further, the learned dimensions of the domain are indeed interpretable, satisfying R1. It holds that moving from left to right on the x-axis corresponds to decreasing the amount of smiling, regardless of male or female. Similarly, moving from the bottom up on the y-axis corresponds to transitioning from female to male faces, regardless of whether the face is smiling. This is seen on the border of Figure 5, which shows decoded representations at the extreme points on edge of the domain. These images are generated directly from the selected domain coordinates, i.e., no real images were used to create them. This illustrates a powerful feature of the proposed AE-based framework, namely, that the decoder module

can be used to *interpret* the learned dimensions.

Finally, note that the encoded representations require far less bits as opposed to the raw images themselves. As only two floating-point values are used to encode the semantics of the images, only $2 \times 32 = 64$ bits are required to convey the relevant meaning, rather than the $112 \times 112 \times 3 \times 8 = 301,056$ bits (before compression) required to transmit the raw image. This corresponds to a rate reduction of over 99%, illustrating the gains that are possible using SemCom.

5.5 Chapter Summary

In this chapter, we proposed a method for learning a domain of a CS model, to reduce the reliance of SCCS on hand-crafted semantic models. The proposed framework is based on the AE architecture, with an added property classifier module to provide semantic regularization during training. We set out to show that the proposed method learns distinct property regions that preserve semantic similarity relations, and that the learned dimensions of the domain are interpretable. Experimental results confirmed that the framework is indeed capable of meeting these requirements of CS learning.

Chapter 6

Improved Learning and Causal Reasoning for Semantic Communication

As discussed in the previous chapter, one of the major challenges of the SCCS approach is obtaining the CS itself. As a solution, we proposed an AE-based framework for learning the domains of a CS [138]. Humans' perception of meaning is often influenced by prior knowledge/experience that is not effectively captured in [138]. In this chapter, we further innovate on this framework by leveraging the VAE architecture [139] and Wasserstein distance from optimal transport (OT) theory to improve the quality of the learned domains and allow for the inclusion of prior knowledge. Moreover, it has been suggested that *causal reasoning* will be a key functionality of intelligent future SemCom systems [140]. To achieve this higher level of intelligence, in this chapter we also integrate causal reasoning into our SCCS framework for a goal-oriented SemCom scenario by way of a goal-oriented reasoning module which learns causal relationships to transmit only the most relevant meanings to the intended receiver in bandwidth-constrained scenarios.

6.1 Related Work

In the previous chapter, we proposed a framework for learning the domains of a CS and discussed some related work. Additionally, causal reasoning has been proposed as a promising technique for achieving this greater intelligence [118]. In the current literature, there has been some recent interest in integrating reasoning into SemCom. Some of these works study contextual reasoning [141], reasoning with generative AI [142], reasoning over explicit/implicit semantics [143], and reasoning about the relevance of image objects [144] for SemCom. However, all of these ignore the causal relationships among the data, the knowledge of which is crucial for truly intelligent SemCom [140]. On the other hand, some reasoning-based SemCom approaches do address the causal aspects of reasoning [145, 146], where the causal relationships among the observed event variables (or system states) are regarded as the semantics of the information. This approach is fundamentally different from the one proposed in this chapter, where we use a CS to model the semantics of the observed data and then use the causal relationships between these semantic variables and some task variables to improve the communication process. To the best of our knowledge, this work presented here is the first to leverage the causal relationships between explicitly-defined semantic variables and downstream task variables to achieve higher-level intelligence in SemCom.

6.1.1 Chapter Contributions

In this chapter we innovate on the AE-based framework for learning the domains of a CS, as well as propose a novel *reasoning module* which is integrated into the SemCom system in order to leverage the benefits of causal reasoning for SCCS. Specifically, our contributions include:

- Introduction of the VAE for learning the domains of the CS. In contrast to a basic AE, use of the VAE results in a simpler training process through implicit regularization and higher-quality representation spaces.
- The use of the p -Wasserstein distance function in the loss of the domain learning

process. Utilizing this mapping, rather than a simple cross-entropy loss term, allows for the incorporation of intuitive prior knowledge about the similarity structure of the domain, resulting in learned semantic spaces that better reflect human knowledge.

- The addition of a causal reasoning module into the SCCS framework, termed SCCS-R. This module imbues the system with a greater level of intelligence, allowing it to reason over the ways in which its learned semantics affect the task at hand. This greater intelligence is thus used to communicate more effectively and efficiently.
- Simulation of the proposed learning and reasoning frameworks on an image dataset, and comparison to several baseline systems, including technical, effective, and end-to-end semantic baselines. The simulations show that the SCCS-R system often outperforms the baseline methods, achieving accuracy improvements of up to 40% on classification tasks and demonstrating similar performance to more traditional communication techniques with a 99.9% reduction in the communication rate.

6.2 Background

We begin by providing an overview of some important background information regarding the concepts used throughout this chapter. Specifically, we provide some details on OT theory and causal ML.

6.2.1 Optimal Transport Theory

OT theory centers around a problem with deep historical roots, and has recently reemerged to offer useful solutions for many modern problems [147]. The solution of the OT problem provides an optimal mapping between probability distributions, considering some ground metric on the support set of these distributions which defines a *cost* of moving from one point to another. OT applies to both continuous and discrete distributions; in what follows, we focus only on the discrete case. The OT problem for two discrete distributions μ, ν in

the probability simplex Δ^{N-1} , the solution of which yields the *Wasserstein p -distance* W_p^p between these distributions, is defined as [148]

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\mathbf{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \mathbf{T}, \mathbf{C} \rangle, \quad (6.1)$$

where $\mathbf{C} \in \mathbb{R}_+^{N \times N}$ is the *cost matrix* for which $\mathbf{C}[i, j] = \delta^p(x_i, x_j)$ defines the cost to move from the i th element to the j th element in the support set of the distributions. The set of valid transport plans is defined as [148]

$$\Pi(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\mathbf{T} \in \mathbb{R}_+^{N \times N} : \mathbf{T}\mathbf{1} = \boldsymbol{\mu}, \mathbf{T}^T\mathbf{1} = \boldsymbol{\nu}\}, \quad (6.2)$$

where $\mathbf{1}$ is the all-one vector. Efficient optimization of this problem is possible by way of *entropic regularization*, which reformulates the problem as [148]

$$\gamma W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\mathbf{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \mathbf{T}, \mathbf{C} \rangle - \frac{1}{\gamma} H(\mathbf{T}), \quad (6.3)$$

where

$$H(\mathbf{T}) = - \sum_{i=1}^N \sum_{j=1}^N t_{ij} \log t_{ij}. \quad (6.4)$$

Solving the OT problem (or the regularized approximation) and obtaining the Wasserstein p -distance provides a way of comparing two probability distributions. Optimizing for the Wasserstein p -distance has been shown to result in representations that provide better interpolations between classes and more interpretable spaces [149].

6.2.2 Causal Machine Learning

Causal methods are concerned with identifying the true *causal* effect of one variable on another, in contrast to the *correlation* between the two. The divergence between causality and correlation is due to the presence of *confounding variables* which produce a spurious correlation between the variables of interest [150]. As a simple example, the number of

shark attacks and ice cream sales might show a strong correlation, though it is clear that buying ice cream does not cause one to be attacked by a shark. The confounding variable in this case is the time of year, which affects both of the variables and creates a spurious correlation between the two.

Causal relationships between variables in a data-generating process are often modeled using a directed acyclic graph (DAG) $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of directed edges [151]. Obtaining this graph (i.e., identifying the presence of causal relationships) can be done either by utilizing background/expert knowledge about the data-generating process or by using data-driven techniques known as *causal discovery*.

Given $\mathcal{G}$, *causal inference* is the process of estimating the strength of the causal effect of one variable on another. One way this effect is quantified is by the *average treatment effect* (ATE). Suppose we have a binary treatment variable $T \in \{0, 1\}$. The ATE with respect to some outcome variable $Y \in \mathbb{R}$ is defined as [151]

$$\text{ATE} = \frac{1}{N} \sum_{i=1}^N Y_i^1 - Y_i^0, \quad (6.5)$$

where Y_i^j denotes the value of the i th outcome under treatment $T = j$. Note that this illustrates the fundamental problem of causal inference: we can never observe the outcome of two mutually exclusive treatments at the same time [151]. However, given that some assumptions are satisfied and that $\mathcal{G}$ possesses a certain structure, it is possible to accurately estimate causal effects provided only observational data. For more details, the reader is referred to [150, 151].

6.3 Improved Domain Learning

In this section, we focus on the problem of learning CS representations for SCCS. We describe in detail two innovations of the AE-based domain learning framework presented in Chapter 5, namely, the use of a VAE for improved and inherent regularization as well as the Wasserstein p -distance for improved quality and incorporation of prior domain knowledge. These changes

impact both the architecture (shown in Figure 5.2) and the loss function (given by (5.6)).

There are some drawbacks to this domain learning approach. First, the distance regularization term ℓ_d , defined in (5.5), is a heuristic regularization term with its own hyperparameters that must be tuned. These extra parameters further complicate the process of learning a domain. Moreover, there are often times when we have some background knowledge regarding the similarity structure of the domain to be learned. For example, say we want to learn a domain representing colors. We know intuitively that orange is more similar to red than blue, and thus should be closer to the former than the latter in the learned space. In the original framework, however, there is no way to include this valuable prior knowledge into the learning process.

6.3.1 Variational Autoencoder

The VAE combines the well-known AE architecture from ML with the theoretically-rich variational Bayesian approach to statistical inference [139]. The *encoder* model $\mathcal{E}$ of an AE maps the input data $\mathbf{x}$ to some (typically lower-dimensional) encoded representation $\mathbf{z}$. The *decoder* model $\mathcal{D}$ takes $\mathbf{z}$ as its input and outputs an approximate reconstruction of the original data $\hat{\mathbf{x}}$. The basic AE is represented mathematically as

$$\hat{\mathbf{x}} = \mathcal{D}_{\Theta}(\mathcal{E}_{\Phi}(\mathbf{x})), \quad (6.6)$$

where Θ and Φ denote the parameters of the decoder and encoder, respectively.

In a VAE, the decoder is considered as a parametric model used to approximate the distribution of the observed data conditioned on the latent variable $p_{\Theta}(\mathbf{x} \mid \mathbf{z})$. Conversely, the encoder captures the conditional distribution of the latent variables on the observed variables $q_{\Phi}(\mathbf{z} \mid \mathbf{x})$. The process of obtaining a prediction is given by

$$\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z = \mathcal{E}_{\Phi}(\mathbf{x}), \text{ sample } \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_z, \text{diag}(\boldsymbol{\sigma}_z^2)), \quad (6.7)$$

$$\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x = \mathcal{D}_{\Theta}(\mathbf{z}), \text{ sample } \hat{\mathbf{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \text{diag}(\boldsymbol{\sigma}_x^2)), \quad (6.8)$$

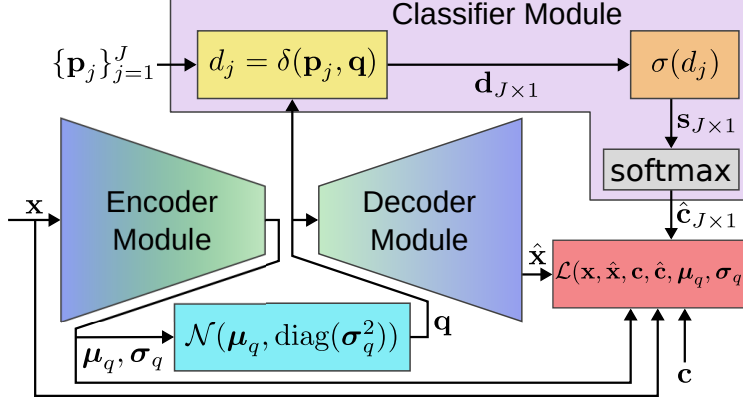


Figure 6.1: Proposed domain learning framework.

where $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$ denotes a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\text{diag}(\boldsymbol{\sigma}^2)$. For training the VAE, one can maximize the evidence lower bound (ELBO) [139]

$$ELBO(\mathbf{x}, \mathbf{z}) = E_{q_{\Phi}(\mathbf{z}|\mathbf{x})}[\log p_{\Theta}(\mathbf{x} | \mathbf{z})] - E_{q_{\Phi}(\mathbf{z}|\mathbf{x})} \left[\log \frac{q_{\Phi}(\mathbf{z} | \mathbf{x})}{p(\mathbf{z})} \right], \quad (6.9)$$

where $\log$ denotes the natural logarithm and $p(\mathbf{z})$ is the assumed prior latent distribution, typically $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

The proposed VAE-based domain learning architecture is shown in Figure 6.1. We take the semantic distance $\delta : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_+$ within a domain $\mathcal{D}$ to be the Euclidean distance

$$\delta(\mathbf{q}_i, \mathbf{q}_j) = \|\mathbf{q}_i - \mathbf{q}_j\|_2 = \sqrt{\sum_{n=1}^N \left(q_i^{(n)} - q_j^{(n)} \right)^2}, \quad (6.10)$$

and we take the semantic similarity function $\sigma : \mathbb{R}_+ \rightarrow (0, 1]$ to be defined as

$$\sigma(d_{ij}) = e^{-cd_{ij}^2}, \quad (6.11)$$

where d_{ij} is the semantic distance between two points $\mathbf{q}_i$ and $\mathbf{q}_j$ and $c \in \mathbb{R}_+$ is a tunable hyperparameter. The goal of the encoder with parameters Φ in the VAE is to learn the conditional distribution $f_{\Phi}(\mathbf{q} | \mathbf{x})$. We assume that this distribution takes the form of a

multivariate Gaussian with independent elements $\mathcal{N}(\boldsymbol{\mu}_q, \text{diag}(\boldsymbol{\sigma}_q^2))$, where $\boldsymbol{\mu}_q$ and $\boldsymbol{\sigma}_q^2$ are the mean and variance vectors, respectively. As such, instead of mapping directly to the domain, the semantic encoder produces two vectors $\boldsymbol{\mu}_q \in \mathcal{D}$ and $\boldsymbol{\sigma}_q \in \mathbb{R}_+^N$ to characterize the distribution¹. With these parameters in hand, a random sample $\mathbf{q} \sim \mathcal{N}(\boldsymbol{\mu}_q, \text{diag}(\boldsymbol{\sigma}_q^2))$ can be drawn to obtain the domain representation of the input $\mathbf{x}$.

Similarly, the goal of the decoder is to learn the conditional distribution $f_{\Theta}(\mathbf{x} \mid \mathbf{q})$. Again, we assume that this distribution is multivariate Gaussian, i.e., $f_{\Theta}(\mathbf{x} \mid \mathbf{q}) = \mathcal{N}(\boldsymbol{\mu}_x, \text{diag}(\boldsymbol{\sigma}_x^2))$. Simplifying assumptions are typically made on the covariance values of this distribution [152], and we will do the same when deriving the loss terms. Finally, to derive the loss function, we need to assume a prior distribution $f(\mathbf{q})$ on the domain representations $\mathbf{q}$. Following the standard approach [139], we take this prior to be the standard normal distribution $f(\mathbf{q}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ where $\mathbf{0}$ is the all-zeros vector and $\mathbf{I}$ is the identity matrix.

We optimize the VAE by maximizing the ELBO (6.9) for the assumed distributions stated above. Let $\boldsymbol{\Sigma}_x$ denote $\text{diag}(\boldsymbol{\sigma}_x^2)$. Looking at the first term of (6.9), we have

$$\begin{aligned} E_{f_{\Phi}(\mathbf{q}|\mathbf{x})}[\log f_{\Theta}(\mathbf{x} \mid \mathbf{q})] &= E_{f_{\Phi}(\mathbf{q}|\mathbf{x})} \left[\log \left(\frac{1}{(2\pi)^{M/2} |\boldsymbol{\Sigma}_x|^{1/2}} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_x)^T \boldsymbol{\Sigma}_x^{-1} (\mathbf{x} - \boldsymbol{\mu}_x)} \right) \right] \quad (6.12) \\ &= E_{f_{\Phi}(\mathbf{q}|\mathbf{x})} \left[-\frac{M}{2} \log 2\pi - \frac{1}{2} \sum_{i=1}^M \log \sigma_{x_i}^2 - \frac{1}{2} \sum_{i=1}^M \frac{(x_i - \mu_{x_i})^2}{\sigma_{x_i}^2} \right] \\ &= -\frac{1}{2} E_{f_{\Phi}(\mathbf{q}|\mathbf{x})} \left[\sum_{i=1}^M (\log 2\pi + \log \sigma_{x_i}^2 + (x_i - \mu_{x_i})^2 / \sigma_{x_i}^2) \right]. \end{aligned}$$

Given a batch of B samples of $\mathbf{x}$ and the $\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x$ obtained by propagating these $\mathbf{x}$ through the VAE, we can use a Monte Carlo approximation to estimate the expectation

$$E_{f_{\Phi}(\mathbf{q}|\mathbf{x})}[\log f_{\Theta}(\mathbf{x} \mid \mathbf{q})] \approx -\frac{1}{2B} \sum_{j=1}^B \sum_{i=1}^M \left(\log 2\pi + \log \sigma_{x_{i,j}}^2 + (x_{i,j} - \mu_{x_{i,j}})^2 / \sigma_{x_{i,j}}^2 \right), \quad (6.13)$$

where $\hat{x}_{i,j}$ denotes the i th element of the j th sample of $\hat{\mathbf{x}}$ obtained from the decoder.

Turning to the second term of (6.9), this is nothing more than the Kullback-Leibler

¹Strictly speaking, the encoder is usually designed to output $\log \boldsymbol{\sigma}_q$ to improve stability and simplify computations.

(KL) divergence between the conditional distribution $f_{\Phi}(\mathbf{q} \mid \mathbf{x})$ and the prior $f(\mathbf{q})$. First separating the log term in the expectation and then proceeding in a similar way as above, we arrive at

$$E_{f_{\Phi}(\mathbf{z}|\mathbf{x})} \left[\log \frac{f_{\Phi}(\mathbf{q} \mid \mathbf{x})}{f(\mathbf{q})} \right] = \frac{1}{2} \sum_{k=1}^N (\mu_{q_k}^2 + \sigma_{q_k}^2 - \log \sigma_{q_k}^2 - 1) \quad (6.14)$$

We can now combine (6.13) and (6.14) to obtain an approximation of the ELBO defined in (6.9)

$$\begin{aligned} ELBO(\mathbf{x}, \mathbf{q}) \approx & -\frac{1}{2} \sum_{k=1}^N (\mu_{q_k}^2 + \sigma_{q_k}^2 - \log \sigma_{q_k}^2 - 1) \\ & - \frac{1}{2B} \sum_{j=1}^B \sum_{i=1}^M \left[\log 2\pi + \log \sigma_{x_{i,j}}^2 + (x_{i,j} - \mu_{x_{i,j}})^2 / \sigma_{x_{i,j}}^2 \right]. \end{aligned} \quad (6.15)$$

As we want to maximize this expression using backpropagation, we first multiply by -1 to transform it to a minimization problem. We can then drop the factor of 1/2 and the constant terms, as they do not affect optimization. Finally, assuming that the loss is averaged over a batch as is typically done when training ML models, we drop this explicit average from the expression. We are then left with the loss function $\mathcal{L}_{VAE}$ where

$$\mathcal{L}_{VAE}(\mathbf{x}, \boldsymbol{\mu}_x, \boldsymbol{\sigma}_x, \boldsymbol{\mu}_q, \boldsymbol{\sigma}_q) = \sum_{i=1}^M (\log \sigma_{x_i}^2 + (x_i - \mu_{x_i})^2 / \sigma_{x_i}^2) + \sum_{k=1}^N (\mu_{q_k}^2 + \sigma_{q_k}^2 - \log \sigma_{q_k}^2). \quad (6.16)$$

As a final step, we make the simplifying assumption that $\forall i \in \{1, 2, \dots, M\}$ we have $\sigma_{x_i} = \sigma_x$ and let $\hat{\mathbf{x}} = \boldsymbol{\mu}_x$. Denote $\lambda = \sigma_x^2$ as a tunable hyperparameter. Dropping the constant term and multiplying the whole expression by the constant λ , we have

$$\begin{aligned} \mathcal{L}_{VAE}(\mathbf{x}, \hat{\mathbf{x}}, \boldsymbol{\mu}_q, \boldsymbol{\sigma}_q) &= \sum_{i=1}^M (x_i - \hat{x}_i)^2 + \lambda \sum_{k=1}^N (\mu_{q_k}^2 + \sigma_{q_k}^2 - \log \sigma_{q_k}^2) \\ &= \ell_r(\mathbf{x}, \hat{\mathbf{x}}) + \lambda \ell_{KL}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q). \end{aligned} \quad (6.17)$$

This loss function provides the new reconstruction and regularization loss terms ℓ_r and

ℓ_{KL} , respectively, for domain learning with the VAE. First, note that the reconstruction loss is nothing more than a squared-error loss which is equivalent to the original reconstruction loss term given by (5.3). Moreover, the regularization term ℓ_{KL} provides multiple advantages over ℓ_d in (5.5). First, it is theoretically-grounded as representing the KL divergence between the assumed prior for $\mathbf{q}$ and the conditional given $\mathbf{x}$, whereas ℓ_d is an entirely heuristic regularization with no theoretical justification. Second, recall that the term ℓ_d contains two hyperparameters of its own, which need to be chosen and tuned based on the problem at hand. This further complicates the training process, whereas ℓ_{KL} does not suffer from this drawback. Overall, the improved loss function is more straightforward to implement and provides greater control over the “shape” of the resulting domain through selection of the prior.

6.3.2 Learning With Wasserstein Distance

To further innovate on the domain learning framework, we propose the use of the entropic-regularized Wasserstein p -distance ${}^\gamma W_p^p$ defined in (6.3) as the loss term corresponding to the classifier module. As shown in Figure 6.1, the classifier module outputs a vector $\hat{\mathbf{c}} \in \Delta^{J-1}$, obtained by taking the softmax operation on the similarity vector $\mathbf{s} \in [0, 1]^J$ where $[0, 1]^J$ denotes the unit hypercube in J dimensions. Thus, we can interpret $\hat{\mathbf{c}}$ as a probability vector over the J properties, where a higher similarity s_j relative to the other $s_{i, i \neq j}$ will yield a higher probability $\hat{c}_j$. Given that the true labels $\mathbf{c} \in \Delta^{J-1}$ are one-hot encoded vectors, both $\hat{\mathbf{c}}$ and $\mathbf{c}$ represent discrete probability distributions, and we can solve (6.3) to obtain a distance between the two².

Solving (6.3) requires us to find a transport map $\mathbf{T}$ from $\hat{\mathbf{c}}$ to $\mathbf{c}$ (or equivalently, from $\mathbf{c}$ to $\hat{\mathbf{c}}$) given a cost matrix $\mathbf{C}$, where $\mathbf{C}[i, j] = \delta^p(x_i, x_j)$ and x_i, x_j are elements of the support set for $\hat{\mathbf{c}}, \mathbf{c}$. The element $\mathbf{C}[i, j]$ captures the cost of “moving” probability mass from the i th element in the support set to the j th element. As $\hat{\mathbf{c}}$ and $\mathbf{c}$ are defined over the set of properties

²Note that we require $p \geq 1$ for $W_p = \sqrt[p]{W_p^p}$ to be a valid metric function [147], while ${}^\gamma W_p^p$ represents a regularized approximation of the p th power of this distance function. The p th power is used to simplify computations as it is monotonic on $\mathbb{R}_+$ for $p > 0$ and does not affect optimization.

in the domain, x_i and x_j thus correspond to the properties indexed by i and j . Moreover, as we have already defined the notion of a prototype point $\mathbf{p}_j \in \mathcal{D}$ corresponding to prototype j and the semantic distance function δ , in our framework the costs can be naturally defined as the *semantic distances between property prototypes*, i.e., $\mathbf{C}[i, j] = \delta^p(\mathbf{p}_i, \mathbf{p}_j)$. The greater the semantic distance between properties i and j , the higher the penalty will be for mapping from i to j . Using an example of the color domain, given the correct cost structure, mistaking red for blue in this domain will be penalized greater than mistaking red for orange. As a key takeaway, leveraging the geometry of the CS model of meaning while optimizing the Wasserstein p -distance allows us to assign interpretable costs between domain properties when learning semantic representations.

As can be seen in (6.3), computing the regularized distance ${}^\gamma W_p^p$ involves solving an optimization problem, and is thus not always straightforward. However, in some cases this optimization can be performed efficiently and incorporated into the standard supervised training algorithm of a DNN. One example from the continuous domain is when both distributions are Gaussian and the ground metric is the Euclidean distance, for which a closed-form expression of the Wasserstein distance exists [153]. In our case, the solution to the regularized discrete OT problem can be found efficiently using the Sinkhorn-Knopp algorithm [148]. Moreover, we can use Sinkhorn iterations to compute the gradients $\nabla_{\hat{\mathbf{c}}}({}^\gamma W_p^p)$ necessary for backpropagation [148]. This algorithm is given below in Algorithm 2, where $\mathbf{K} = e^{-\gamma \mathbf{C}^{-1}}$ and the exponential is performed element-wise.

Algorithm 2 Gradient of the Wasserstein loss [148, Alg. 1].

```

1: Given  $\hat{\mathbf{c}}, \mathbf{c}, \gamma, \mathbf{K}$ 
2:  $\mathbf{u} \leftarrow \mathbf{1}$ 
3: while  $\mathbf{u}$  has not converged do
4:    $\mathbf{u} \leftarrow \hat{\mathbf{c}} \oslash (\mathbf{K}(\mathbf{c} \oslash \mathbf{K}^T \mathbf{u}))$ 
5: end while
6:  $\nabla_{\hat{\mathbf{c}}}({}^\gamma W_p^p) \leftarrow \frac{\log \mathbf{u}}{\gamma} - \frac{\log \mathbf{u}^T \mathbf{1}}{\gamma J} \mathbf{1}$ 

```

6.3.3 Putting It Together

Letting ${}^\gamma W_p^p$ replace the classification loss ℓ_c in (5.6), the new loss function for the domain learning framework is given by

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{c}, \hat{\mathbf{c}}, \boldsymbol{\mu}_q, \boldsymbol{\sigma}_q) = \alpha \ell_r(\mathbf{x}, \hat{\mathbf{x}}) + \beta {}^\gamma W_p^p(\mathbf{c}, \hat{\mathbf{c}}) + \lambda \ell_{KL}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q), \quad (6.18)$$

where ℓ_r and ℓ_{KL} are defined in (6.17), ${}^\gamma W_p^p$ is defined in (6.3), and α, β, λ are tunable hyperparameters for balancing the loss function. As we are using Algorithm 2 to directly compute the gradient $\nabla_{\hat{\mathbf{c}}} \mathcal{L} = \beta \nabla_{\hat{\mathbf{c}}} {}^\gamma W_p^p$ rather than the loss itself, we must do the same with the other loss terms. For the reconstruction loss, we have

$$\nabla_{\hat{\mathbf{x}}} \mathcal{L} = \alpha \nabla_{\hat{\mathbf{x}}} \ell_r = 2\alpha(\hat{\mathbf{x}} - \mathbf{x}), \quad (6.19)$$

and

$$\nabla_{\boldsymbol{\mu}_q} \mathcal{L} = \lambda \nabla_{\boldsymbol{\mu}_q} \ell_{KL} = 2\lambda \boldsymbol{\mu}_q, \quad (6.20)$$

$$\nabla_{\log \boldsymbol{\sigma}_q} \mathcal{L} = \lambda \nabla_{\log \boldsymbol{\sigma}_q} \ell_{KL} = 2\lambda(e^{2\log \boldsymbol{\sigma}_q} - 1), \quad (6.21)$$

where the encoder of the VAE is typically configured to output $\log \boldsymbol{\sigma}$, and thus we require the gradient with respect to these outputs rather than the variable $\boldsymbol{\sigma}_q$. With all gradients in hand, we can optimize the model using backpropagation.

The full domain learning algorithm is given in Algorithm 3. The key idea is similar to that of Algorithm 1, which is to perform iterative updates to learn the semantic encoder model mapping raw data to the CS domain as well as the prototype points of the property regions within the domain.

Algorithm 3 Domain Learning with VAE and Wasserstein p -Distance

Require: $\alpha, \beta, \lambda, \eta, K, B, \{\mathbf{p}_j\}_{j=1}^J = \mathbf{0}$

- 1: **for** each epoch i in $\{1, 2, \dots, E\}$ **do**
- 2: **for** each batch K random batches **do**
- 3: **if** i is 1 **then**
- 4: $\nabla_{\hat{\mathbf{c}}} \mathcal{L} \leftarrow \mathbf{0}$
- 5: $\nabla_{\hat{\mathbf{x}}} \mathcal{L}, \nabla_{\mu_q} \mathcal{L}, \nabla_{\log \sigma_q} \mathcal{L} \leftarrow$ from (6.19), (6.20), (6.21)
- 6: **else**
- 7: $\nabla_{\hat{\mathbf{c}}} \mathcal{L} = \beta \nabla_{\hat{\mathbf{c}}} \gamma W_p^p \leftarrow$ from Algorithm 2
- 8: $\nabla_{\hat{\mathbf{x}}} \mathcal{L}, \nabla_{\mu_q} \mathcal{L}, \nabla_{\log \sigma_q} \mathcal{L} \leftarrow$ from (6.19), (6.20), (6.21)
- 9: **end if**
- 10: Perform batch update with backpropagation
- 11: **end for**
- 12: Encode KB random samples to obtain $\{\mathbf{q}\}$
- 13: **for** each property j in $\{1, 2, \dots, J\}$ **do**
- 14: $\bar{\mathbf{q}}_j \leftarrow \text{mean}(\{\mathbf{q} : \mathbf{q} \text{ possesses property } j\})$
- 15: $\mathbf{p}_j^{(i)} \leftarrow \bar{\mathbf{q}}_j$ if $i = 1$, else $\eta \mathbf{p}_j^{(i-1)} + (1 - \eta) \bar{\mathbf{q}}_j$
- 16: **end for**
- 17: **end for**
- 18: **return** Encoder, decoder and prototypes $\{\mathbf{p}_j\}_{j=1}^J$

6.4 Conceptual Semantic Communication with Causal Reasoning

Reasoning can be defined as “to think in a connected or logical manner” [154]. Thus, reasoning is envisioned as a key capability of future intelligent systems [140]. With regard to our goal of improved SemCom, causal reasoning can enable a system to learn causal relations in the data and perform intelligent decision-making given this causal knowledge. In this paper, we aim to employ causal reasoning to answer the goal-oriented question: “Which semantic qualities in $\mathbf{z}$ have the most causal influence on the current task?” Suppose a setup in which the ultimate output of the SemCom system is some task variable(s) denoted by the vector $\mathbf{u} \in \mathbb{R}^U$, where U is the number of elements in the vector.

Given the SCCS framework for SemCom, we can immediately identify two examples of valuable causal information. The first is information about the causal effects of the

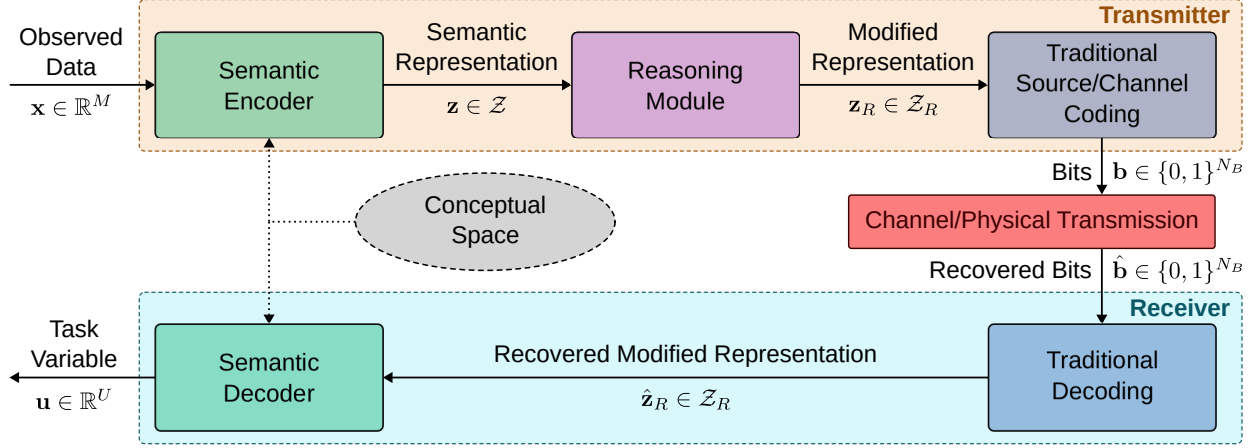


Figure 6.2: Block diagram of the proposed goal-oriented SCCS framework with reasoning

observed data $\mathbf{x}$ on the full CS semantic representations³ $\mathbf{z}$, which would allow us to answer questions such as “which variables in $\mathbf{x}$ caused the semantic representation $\mathbf{z}$ to take on a certain property?” Second, suppose a goal-oriented setup in which the ultimate output of the SemCom system is some task variable(s) denoted by the vector $\mathbf{u} \in \mathbb{R}^U$, where U is the number of elements in the vector. Then we can use causal knowledge between the semantic and task variables to answer questions such as “which semantic qualities in $\mathbf{z}$ have the most causal influence on the current task?” This knowledge can then be leveraged to make intelligent decisions within the SCCS system. In this chapter, we focus on the goal-oriented scenario.

An overview of our proposed approach to SCCS with reasoning (SCCS-R) is illustrated in Figure 6.2. The key innovation in the SCCS-R system is the addition of the *reasoning module*. This module leverages causal knowledge about the semantic and task variables to improve task performance under communication constraints, such as bandwidth limitations. This causal knowledge is obtained by performing causal *discovery* and *inference* on a dataset to uncover the causal relationships between the semantic and task variables. In the rest of this section, we describe in detail the reasoning module, the four-step process of obtaining causal knowledge, and the semantic decoder module.

³Note that $\mathbf{z}$ denotes the representation over the entire CS, which is composed of multiple domains in general. The vector $\mathbf{z}$ is formed by concatenating representations $\mathbf{q}$ from the individual domains of the CS.

6.4.1 Reasoning Module

As shown in Figure 6.2, the reasoning module takes the output $\mathbf{z}$ of the semantic encoder(s) and transforms it before it is sent over the channel to the receiver. Therefore, we can generally represent the reasoning module as a function $\mathcal{R}_{\mathbf{E}} : \mathcal{Z} \rightarrow \mathcal{Z}_R$ mapping $\mathbf{z} \in \mathcal{Z}$ to some set $\mathcal{Z}_R$, i.e.,

$$\mathcal{R}_{\mathbf{E}}(\mathbf{z}) = \mathbf{z}_R, \quad (6.22)$$

where $\mathbf{z}_R$ is some transformed version of the original semantic representation $\mathbf{z}$. This transformation is performed using what we term the *effect matrix*, denoted by $\mathbf{E}$, where the rows correspond to the elements of $\mathbf{z}$ and the columns correspond to the elements of $\mathbf{u}$, and the element $\mathbf{E}[i, j]$ represents the causal effect of semantic variable z_i on task variable u_j .

This transformation is dependent on two factors, namely, the communication constraints and the causal knowledge possessed by the reasoner. The communication constraints determine *what* transformation is performed. For example, given a maximum power constraint, the reasoner might allocate more power to some elements of $\mathbf{z}$ than others. Likewise, given a bandwidth constraint, the reasoner might choose to drop certain elements of $\mathbf{z}$ to reduce the number of bits transmitted in a given message. The causal knowledge then determines *how* this transformation is applied. This is done using what we term the *effect matrix*, denoted by $\mathbf{E}$, where the rows correspond to the elements of $\mathbf{z}$ and the columns correspond to the elements of $\mathbf{u}$, and the element $\mathbf{E}[i, j]$ represents the causal effect of semantic variable z_i on task variable u_j . As the effect matrix represents the knowledge on which the reasoner module acts, we include this matrix in the notation for the reasoner as

In what follows, we assume that the system is operating under a bandwidth constraint, such that the reasoner must discard some values of $\mathbf{z}$ so that $\mathbf{z}_R$ contains a maximum of L values. Thus we have $\mathcal{Z}_R \subset \mathcal{Z}$ where $\mathcal{Z}_R$ is formed by a subset of the quality dimensions in $\mathcal{Z}$. To transmit the most *useful* semantic information, we would like to convey the dimensions that have the greatest causal effect on the task variables. We can determine these dimensions by summing the absolute values of the rows of $\mathbf{E}$ to obtain the vector $\mathbf{e} \in \mathbb{R}_+^Z$, whose elements

are given by

$$e_i = \sum_{j=1}^U |\mathbf{E}[i, j]|, \quad i = \{1, 2, \dots, Z\}, \quad (6.23)$$

where Z denotes the total number of quality dimensions in the CS, and U denotes the number of elements in $\mathbf{u}$, i.e., the number of task variables. Moreover, denote the ordered elements of $\mathbf{e}$ as $\{e^{(1)}, e^{(2)}, \dots, e^{(Z)}\}$ where $e^{(1)}$ denotes the maximum element of $\mathbf{e}$ and $e^{(Z)}$ denotes the minimum, and define $\mathcal{J}(e^{(i)})$ as the function that maps the element $e^{(i)}$ to its original index in $\mathbf{e}$. Then the function $\mathcal{R}_{\mathbf{E}}$ will output a vector $\mathbf{z}_R$ with elements

$$z_{Ri} = z_{\mathcal{J}(e^{(i)})}, \quad i = 1, 2, \dots, L. \quad (6.24)$$

Now that the reasoning module has been defined, we proceed to the problem of learning the causal effects, that is, determining the matrix $\mathbf{E}$ that is utilized by the reasoning module.

6.4.2 The Four Steps of Causal Inference

The process of estimating causal effects can be broken down into four primary steps [151]. In this subsection, we describe in detail the process for determining the causal effects that constitute the effect matrix $\mathbf{E}$. As our primary contribution is the application of this general framework rather than any specific implementation, we will focus more on the general process and provide only brief details for our chosen implementation that we use to illustrate and experiment with this process. We emphasize that a number of techniques can be chosen for each of the four steps based on the design requirements and available data.

Causal Discovery

As mentioned in section 6.2.2, causal relationships are often modeled using a DAG, denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of directed edges between them [151]. In this model, the nodes represent variables of the data-generating process and an edge from node i to node j represents the fact that the i th variable has some causal influence on the j th variable. The first step of estimating the strength of these influences is to identify

which edges are present in the graph. This is the problem of causal discovery: given $\mathcal{V}$ and some samples of the variables represented by $\mathcal{V}$, estimate $\mathcal{E}$. Many techniques for causal discovery have been proposed in the literature, and for a recent survey we refer the reader to [155].

There are two primary techniques for accomplishing this task. The first assumes that we have access to some expert knowledge about the underlying data-generating process, in which case we can leverage this knowledge to manually assign edges between nodes. Alternatively, given a dataset $\{\mathbf{v}\}$, where $\mathbf{v}$ is a vector denoting the values of the nodes in $\mathcal{V}$, we can leverage data-driven techniques to identify causal connections in a more automated fashion. Though this is a rather difficult problem, many techniques have been proposed in the literature, including constraint-based [156], score-based [157], functional-based [158] and gradient-based [159] methods. For a recent survey on causal discovery methods, we refer the reader to [155].

In the context of SCCS-R, we need to determine the effect matrix $\mathbf{E}$ capturing the causal effects of the semantic representation variables $\mathbf{z}$ on the task variables represented by $\mathbf{u}$. To begin, we assume that we have access to a dataset of samples of observed data and the correspond task-related labels $\{\mathbf{x}, \mathbf{u}\}$. We train some semantic encoder(s) as described in section 6.3 and obtain the semantic representations corresponding to the $\{\mathbf{x}\}$ to form the new dataset $\{\mathbf{z}, \mathbf{u}\}$. We assume no “backward” causal connections, i.e., no elements $\mathbf{u}$ cause any elements of $\mathbf{z}$, formally, $\forall u_i \in \mathbf{u}, z_j \in \mathbf{z}$ we have $(u_i, z_j) \notin \mathcal{E}$. Moreover, we assume that no causal connections exist between quality dimensions belonging to the same domain, i.e., if $z_i \in \mathcal{Q}_i, z_j \in \mathcal{Q}_j$ and $\mathcal{Q}_i, \mathcal{Q}_j \in \mathcal{D}_k$ for some $\mathcal{D}_k \in \mathcal{Z}$, then $(z_i, z_j) \notin \mathcal{E}$. To enforce these assumptions, we require a causal discovery method that allows for the incorporation of prior knowledge. In this work, we use the DirectLiNGAM algorithm [158]. This approach assumes that the data are linearly related with some non-Gaussian additive noise. DirectLiNGAM requires no algorithmic parameters and is guaranteed to converge to the true solution, given that the data follows the model [158].

Estimand Identification

With an estimated DAG in hand, we can proceed to identify the causal estimand, which describes the quantity that we will estimate to obtain the causal effect. Identifying the estimand involves examining the structure of the causal graph to determine which variables need to be controlled for when estimating the causal effect of one variable on another. We do this by utilizing the framework of *do*-calculus, which uncovers all causal effects that can be identified from a given graph [160]. Basically, *do*-calculus provides us with a way of converting probabilities using the *do* operator to ordinary conditional probabilities that can be estimated using any traditional techniques. For example, the *causal effect rule* is given as

$$P(Y = y \mid do(X = x)) = \sum_z P(Y = y \mid X = x, Pa = z)P(Pa = z) \quad (6.25)$$

where X, Y, Z are variables used only for illustration in this example and Pa denotes the parents of X in $\mathcal{G}$. Using *do*-calculus, we can identify certain criteria that allow us to estimate true causal effects. The back-door criterion aims to block spurious paths between the treatment and outcome nodes [151, 160]. The front-door criterion allows us to obtain an estimand in some cases where the back-door criterion is not met [151, 160]. If neither of these criteria are met, the method of instrumental variables can sometimes be used to estimate the causal effect [151, 161]. A detailed introduction to *do*-calculus is beyond the scope of this work, and we refer the reader to [150, 160] for more details.

Effect Estimation

Once we have identified the estimand, we require a method to carry out the estimation of the causal effect. This estimator can take several forms depending on the problem at hand and the chosen method, from simple linear regression to complex ML approaches. To determine the effect matrix $\mathbf{E}$, we are interested in estimating the conditional average treatment effect (CATE). For an outcome variable Y , binary treatment T and a set of related variables $\{X\}$,

the CATE is expressed as

$$CATE(X) = E[Y \mid T = 1, \{X\}] - E[Y \mid T = 0, \{X\}]. \quad (6.26)$$

In (6.26), $CATE(X)$ captures the effect that the treatment T has on the outcome Y . Notice that this formulation only makes sense for binary treatments. However, in our problem of interest, the “treatments” $\mathbf{z}$ are continuous variables. Therefore, we use the *double machine learning* (DML) method [162], which is valid for continuous treatments and makes the following structural assumptions on the data generating process:

$$Y = \theta(X) \cdot T + g(X, W) + \epsilon \quad E[\epsilon \mid X, W] = 0 \quad (6.27)$$

$$T = f(X, W) + \eta \quad E[\eta \mid X, W] = 0 \quad (6.28)$$

$$E[\eta \cdot \epsilon \mid X, W] = 0, \quad (6.29)$$

where W are the controls/confounders, X are the observable characteristics of the treated samples, ϵ, η are noise variables, f, g are arbitrary functions and $\theta(X)$ denotes the CATE. Let $\mathbf{z}_w$ denote the controls/confounders identified in the previous step and $\mathbf{z}_x$ be the remaining CS variables. Then for the problem of estimating $\mathbf{E}[i, j]$, we have

$$u_j = \theta_{ij}(\mathbf{z}_x) \cdot z_i + g(\mathbf{z}_x, \mathbf{z}_w) + \epsilon \quad E[\epsilon \mid \mathbf{z}_x, \mathbf{z}_w] = 0 \quad (6.30)$$

$$z_i = f(\mathbf{z}_x, \mathbf{z}_w) + \eta \quad E[\eta \mid \mathbf{z}_x, \mathbf{z}_w] = 0 \quad (6.31)$$

$$E[\eta \cdot \epsilon \mid \mathbf{z}_x, \mathbf{z}_w] = 0. \quad (6.32)$$

We can now fit two models to estimate the conditional expectations

$$h(\mathbf{z}_x, \mathbf{z}_w) = E[u_j \mid \mathbf{z}_x, \mathbf{z}_w], \quad f(\mathbf{z}_x, \mathbf{z}_w) = E[z_i \mid \mathbf{z}_x, \mathbf{z}_w], \quad (6.33)$$

and obtain the residuals

$$u'_j = u_j - h(\mathbf{z}_x, \mathbf{z}_w), \quad z'_i = z_i - f(\mathbf{z}_x, \mathbf{z}_w), \quad (6.34)$$

which are related by

$$u'_j = \theta_{ij}(\mathbf{z}_x) \cdot z'_i + \epsilon. \quad (6.35)$$

Finally, we need to solve

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} E \left[(u'_j - \theta(\mathbf{z}_x) \cdot z'_i)^2 \right], \quad (6.36)$$

and we can then average to obtain the effect

$$\mathbf{E}[i, j] = \frac{1}{|\{\mathbf{z}_x\}|} \sum_{\mathbf{z}_x \in \{\mathbf{z}_x\}} \theta(\mathbf{z}_x), \quad (6.37)$$

where $|\{\mathbf{z}_x\}|$ is the number of samples in the dataset. The DML approach is versatile, as we can fit any arbitrary model to estimate the functions f, h , and θ [162].

Refutation

The final step in the causal inference process is to validate the obtained estimates with *refutation tests*. These tests alter either the causal model/graph or the data to achieve this goal. Some examples of altering the model include adding a placebo treatment, a dummy outcome, or a simulated outcome [163]. Alternatively, one could add “unobserved” confounders for which we know the overall effect should not change the estimated effect, such as a random common cause [163]. When altering the data, one option is to re-estimate the effect on many subsets of the original dataset, as this should also produce the same effect [163].

After obtaining the effect matrix $\mathbf{E}$, we perform random subset and random common cause refutation tests [163] on the estimated effects. For the random subset test, each effect $\mathbf{E}[i, j]$ is re-estimated using the same DML technique on a subset of the data samples. The

average of the newly estimated effects is then taken and compared to the original estimate to check for significant deviation. For the random common cause test, an independent variable is added to the model as a common cause of both z_i and u_j . This variable is included as a confounder in the identified estimand, and the estimate is re-computed and compared to check for significant deviation.

6.4.3 The Semantic Decoder

With the effect matrix $\mathbf{E}$ and the reasoning module $\mathcal{R}_{\mathbf{E}}$ in place, the final piece of the communication chain is the semantic decoder, shown in Figure 6.2. After the semantic representation $\mathbf{z}$ has been transformed to $\mathbf{z}_R$, it is transmitted over the communication channel, after which $\hat{\mathbf{z}}_R$ is recovered at the receiver. The job of the semantic decoder is to map the recovered representation $\hat{\mathbf{z}}_R$ to an estimate $\hat{\mathbf{u}}$ of the true task variable $\mathbf{u}$.

We propose the use of a DNN model to handle the task of the semantic decoder. The proposed architecture is shown in Figure 6.3. This network consists of multiple fully-connected layers that are optimized to learn the mapping from $\hat{\mathbf{z}}_R$ to $\mathbf{u}$. The size and activation function of the output layer are both dependent of the type of task, e.g., softmax activation could be used for a multiclass-classification task. One notable aspect of this model is the requirement of handling missing values. The transmitted $\hat{\mathbf{z}}_R$ is generally a subset of the values in $\hat{\mathbf{z}}$. We assume that information about *which* elements are included in $\mathbf{z}_R$ is reliably transmitted, e.g., the dropped values are transmitted as null values or a packet header is included with this information. As we would like the decoder to be able to handle representations with any number of missing elements, we add a custom layer to the front of the semantic decoder which maps missing values to a scalar parameter whose value is learned throughout training.

To train the semantic decoder, data is generated using the original dataset $\{\mathbf{z}, \mathbf{u}\}$. First, elements of $\mathbf{z}$ are *randomly* dropped, i.e., without the use of $\mathcal{R}_{\mathbf{E}}$, to obtain a new dataset $\{\mathbf{z}_R, \mathbf{u}\}$. Training with randomly-dropped elements is done for two reasons. First, this allows the decoder to recognize dropped values and learn to ignore them. Second, this ensures that the decoder is learning to perform the task solely based on the semantic representation,

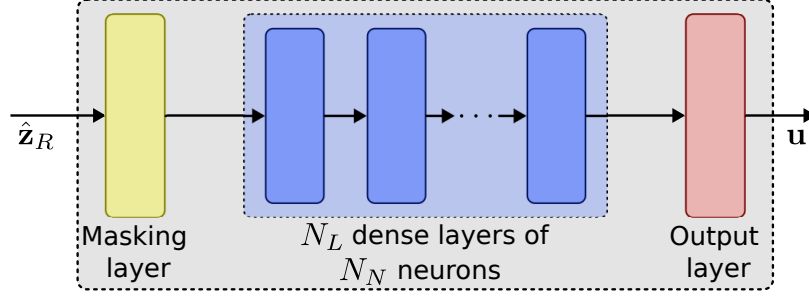


Figure 6.3: Semantic decoder architecture.

i.e., it does not learn any task-related information from the decisions of the reasoner. The final dataset for training the decoder is then obtained by simulating the transmission of $\mathbf{z}_R$ through a communication channel, resulting in data-label pairs $\{\hat{\mathbf{z}}_R, \mathbf{u}\}$. The decoder is then trained in a supervised fashion using backpropagation.

6.5 Experimental Results

In this section we perform experiments to evaluate the merit of the proposed methods. Specifically, we take the case of goal-oriented SemCom of images. We carry out the full workflow of learning the domains for SCCS and building out the SCCS-R system. Finally, we simulate SCCS-R system in a wireless communication scenario and compare the performance to that of some baseline methods⁴.

6.5.1 Setup

In these experiments we use the German Traffic Sign Recognition Benchmark [164], which is a dataset containing over 50,000 real-world color images of 43 different German traffic signs. Our overall objective is to utilize the proposed SCCS-R system for goal-oriented SemCom of this image data. To this end, we define several simple classification tasks to be performed at the receiver: shape classification (with classes triangle, square, octagon, and circle), color classification (with classes red, blue, yellow, and black), and a binary classification task to

⁴Full project code and instructions for running the experiments, from downloading the data to generating the final results, is open and available at <https://github.com/dw-610/concept-learning-and-causal-reasoning>. See Appendix A for more details.

determine whether the image depicts a speed limit sign or not. The first two tasks correspond directly to two domains in the learned CS, and thus provide instances in which we have a reasonable idea of the true causal structure of the data with which we can compare. The third task does not possess this simple correspondence to the learned domains and represents a more realistic scenario. We resize each of the images in the dataset to 64x64x3 and divide the data into three sets. We use the original training set of 39,209 images for learning the domains, and we randomly split the original test set into a set of 10,000 images for performing the causal learning tasks and leave 2,630 images for testing.

Additionally, we simulate three baselines with which we compare our proposed approach:

- *Technical baseline*: We simulate a traditional communication system operating at the “technical” level, in which the full image is transmitted to the receiver. At the receiver, a classifier model is implemented to perform the corresponding task.
- *Effective baseline*: We also simulate a purely goal-oriented communication system, thus operating at the “effective” level of communication. This system implements a classifier model at the transmitter and sends the decision directly to the receiver.
- *End-to-end semantic baseline*: To compare to another popular approach to SemCom, we implement a system in which models at both the transmitter and receiver are jointly trained to carry out the tasks described above. The encoder model at the transmitter is trained to map the image directly to symbols which are transmitted over the channel, and the decoder model at the receiver is trained to map the received symbols to the predicted class.

For the proposed system and each of the baselines, we simulate a wireless communication scenario. The physical operations of the baseband digital communication system are simulated according to the IEEE 802.11 standard [134]. Moreover, we make use of both the AWGN and Rayleigh fading channel models. Simulation of the digital communication system is performed in Python⁵.

⁵These simulation tools are open-source can be accessed at <https://github.com/dw-610/digital-comm-tools>.

6.5.2 Domain Learning on Traffic Signs

We first implement the proposed domain learning technique to learn the CS for the traffic sign dataset. We aim to learn three domains for this data: the *shape* domain, the *color* domain, and the *symbol* domain. We define the following properties for the shape domain: triangle, square, octagon, circle. Similarly, we define properties for the color domain: red, blue, yellow, black. Each sign corresponds to a unique property in the symbol domain (except for two signs that are blank with no symbol). Shape and color are two key features that distinguish different kinds of traffic signs, but sometimes different signs will share the same shape and color. Hence, we include the symbol domain to capture these differences in meaning.

We utilize Algorithm 3 to learn the domains of the CS. For each domain, the encoder model of the VAE takes the form of a CNN with five convolutional layers having 16, 32, 64, 128, and 256 filters, respectively. Each layer uses a 3x3 kernel size and a stride length of 2. The decoder model of the VAE is built as a mirror image to the encoder. For the hyperparameters in Algorithm 3, we use $\alpha = 0.5/(64 \cdot 64 \cdot 3)$, $\beta = 2.0$, $\lambda = 0.1/N$, and $\eta = 0.75$. We train over 200 epochs, learning over 200 batches of size 64 each epoch. We apply a cosine decay learning rate schedule to the Adam optimizer, with an initial rate of 0.0001 increasing to 0.001 after 40 epochs, and decaying to 0 over the next 160 epochs. The parameters for the p -Wasserstein loss were chosen to be $p = 1.0$ and $\gamma = 2.0$. We choose the dimensionality N of the shape, color, and symbol domains as 2, 3, and 15, respectively.

As discussed in section 6.3.2, choosing the cost matrix $\mathbf{C}$ is a crucial step in setting up the domain learning process. For the shape and color domains, it is easy to design this matrix based on relative similarities (e.g., a triangle is closer to a square than a circle). However, for the symbol domain this is not as straightforward. To address this challenge, we define the cost matrix by first creating text descriptions of each sign’s symbol and using SBERT [165] to produce embeddings for these descriptions, and then use the pairwise Euclidean distances between these embeddings to define $\mathbf{C}$. These descriptions can be found in the accompanying code repository. We omit the cost matrix corresponding to the symbol domain for the sake

of brevity, and list these matrices for the shape and color domains in (6.38) and (6.39).

$$\mathbf{C}_{\text{shape}} = \begin{array}{c} \begin{array}{c} \text{triangle} \\ \text{square} \\ \text{octagon} \\ \text{circle} \end{array} \begin{array}{c} \begin{array}{c} \text{triangle} \quad \text{square} \quad \text{octagon} \quad \text{circle} \end{array} \\ \left[\begin{array}{cccc} 0.0 & 0.8 & 1.6 & 2.0 \\ 0.8 & 0.0 & 0.8 & 1.2 \\ 1.6 & 0.8 & 0.0 & 0.4 \\ 2.0 & 1.2 & 0.4 & 0.0 \end{array} \right] \end{array} \quad (6.38)$$

$$\mathbf{C}_{\text{color}} = \begin{array}{c} \begin{array}{c} \text{red} \\ \text{blue} \\ \text{yellow} \\ \text{black} \end{array} \begin{array}{c} \begin{array}{c} \text{red} \quad \text{blue} \quad \text{yellow} \quad \text{black} \end{array} \\ \left[\begin{array}{cccc} 0.0 & 1.0 & 0.5 & 0.75 \\ 1.0 & 0.0 & 1.5 & 0.75 \\ 0.5 & 1.5 & 0.0 & 0.75 \\ 0.75 & 0.75 & 0.75 & 0.0 \end{array} \right] \end{array} \quad (6.39)$$

We provide here some results of learning the shape domain, as the 2D representations allow for intuitive visualization. The scatter plot of the encoded representations for the test set is shown in Figure 6.4. We can see that the learned domain partitions the space into distinct regions corresponding to each property and that the inherent regularization of the VAE loss keeps the features relatively centered about the origin. Moreover, the specified similarity structure is seen in the learned domain. This is even more clear when looking at the heatmap for the similarities between the learned property prototype points, shown in Figure 6.5, which clearly shows that the specified cost structure has enforced our prior understanding of the semantic similarities between these properties. We observed similar results for the color and symbol domains.

To begin to understand what the quality dimensions of this domain represent, we use the decoder to generate images from representations taken along the axes of the domain; these results are shown in Figure 6.6. We can see that the first dimension captures the different shapes nicely; moving from left to right (negative to positive values), we observe the images changing from a circle, to an octagon, to a square, and finally to a triangle. The second

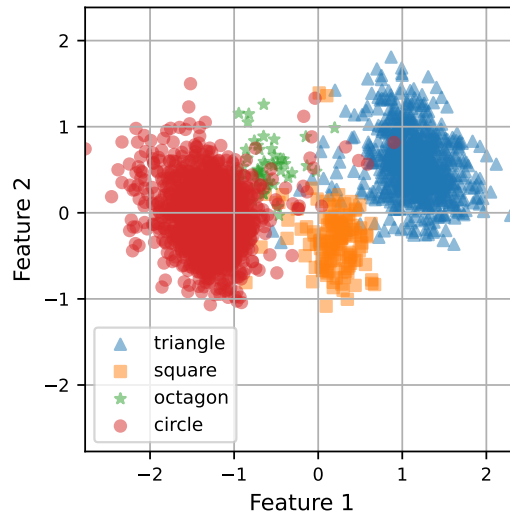


Figure 6.4: Scattered 2D learned representations of the shape domain.

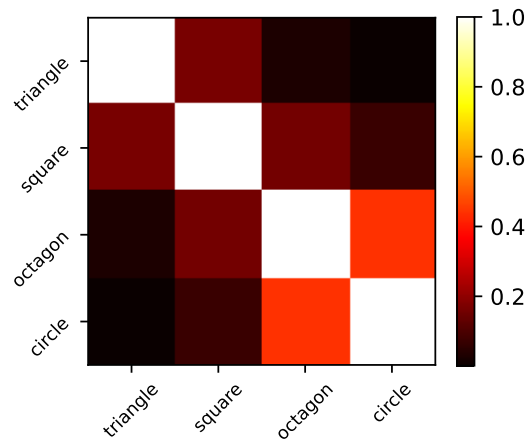


Figure 6.5: Semantic similarity heatmap between the learned property prototypes of the shape domain.



Figure 6.6: Visualized dimensions of the shape domain, obtained by decoding semantic representations along the axes. The top row of images corresponds to the first quality dimension and the bottom row to the second dimension.

dimension is not quite as interpretable, as this axis seems to lie “between” the properties (see Figure 6.4). Further insights into the meaning of this dimension could be found by decoding additional “slices” along this axis in the domain space. Overall, we see that the proposed domain learning technique learns meaningful representations of the input data and successfully preserves the specified similarity structure of the properties.

6.5.3 Reasoning for Improved Communication

The first step in building out the SCCS-R system is to estimate the causal effect matrix $\mathbf{E}$. This is done following the methods outlined in Section 6.4.2 using open-source implementations, namely the LiNGAM [158] (causal discovery), DoWhy [166] (estimand identification, refutation), and EconML [167] (effect estimation) Python packages. As an example, we first perform these steps using only the representations from the shape and color domains for the shape prediction task. Doing so, we end up with the causal effect matrix and absolute-summed vector

$$\mathbf{E} = \begin{array}{c} \text{shape}_1 \\ \text{shape}_2 \\ \text{color}_1 \\ \text{color}_2 \\ \text{color}_3 \end{array} \begin{array}{c} \text{triangle} \text{ square} \text{ octagon} \text{ circle} \\ \begin{bmatrix} 0.33 & - & -0.99 & -0.34 \\ 0.17 & -0.02 & 0.01 & - \\ 0.05 & -0.01 & - & -0.05 \\ 0.18 & -0.11 & - & 0.04 \\ -0.01 & -0.03 & -0.001 & - \end{bmatrix} \end{array}, \mathbf{e} = \begin{bmatrix} 1.664 \\ 0.204 \\ 0.107 \\ 0.329 \\ 0.036 \end{bmatrix}, \quad (6.40)$$

where a dash indicates that no causal relationship was identified in the causal discovery step. We see that the largest causal effect is found for the first dimension of the shape domain, which matches our observations from the previous section. Moreover, note that the dimension with the second-largest influence is found to be the second dimension of the color domain. Notice that it was found to have the greatest effect on the second class (square). In addition to being the only square sign in the dataset, this class is also the only *yellow* sign in the dataset, and thus it is reasonable that a significant causal effect on this variable might be found with regard to a color dimension.

Next, we train the semantic decoder model and the models used in the baseline systems. For the semantic decoder, we use $N_L = 4$ layers of $N_N = 32$ neurons. This model is trained for 100 epochs with a constant learning rate of 0.001 and batch size of 64 on a dataset generated according to the process described in section 6.4.3. For the baseline models, a basic image classifier was trained for each task using an identical architecture and training schedule as the semantic encoder described in section 6.5.2. The only difference in this model is that in every convolutional layer, it uses three times the filters as the semantic encoder model, to balance the fact that three semantic encoders are used in the SCCS-R system. Likewise, an encoder with triple the parameters is jointly trained with decoder models corresponding to each task for the end-to-end semantic baseline, again with an identical training schedule and parameters.

We first compare the SCCS-R system to a similar system that performs random modifications to the semantic representations (i.e., does not employ reasoning to make these decisions) to show the effect of the reasoning module in the proposed system. These results are shown Figure 6.7 for the AWGN channel with BPSK modulation, when the transmitted message is constrained to different lengths. We immediately see that the system with reasoning outperforms the system with random modifications at all message lengths for mid and high SNR values, confirming the benefit of including reasoning. In Figure 6.7a, we see that at message length 1, the system with reasoning nearly matches the system without reasoning using a message length of 4. This again matches previous results, as the first dimension of the shape domain contained nearly all the information needed to accurately classify the

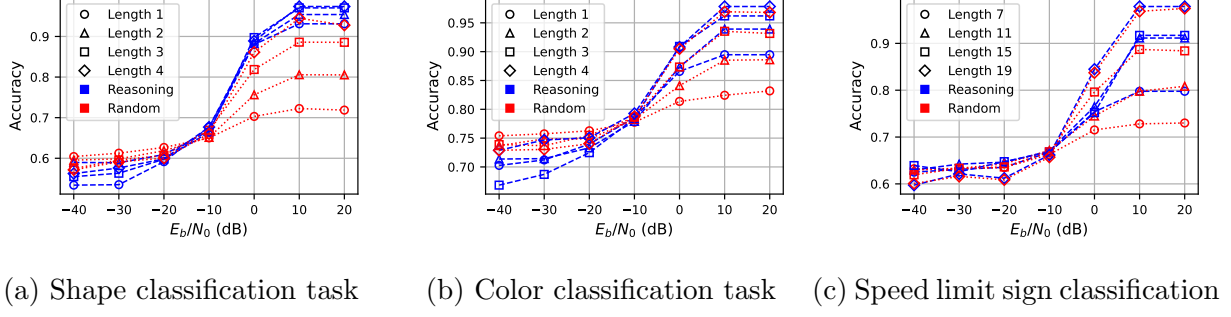
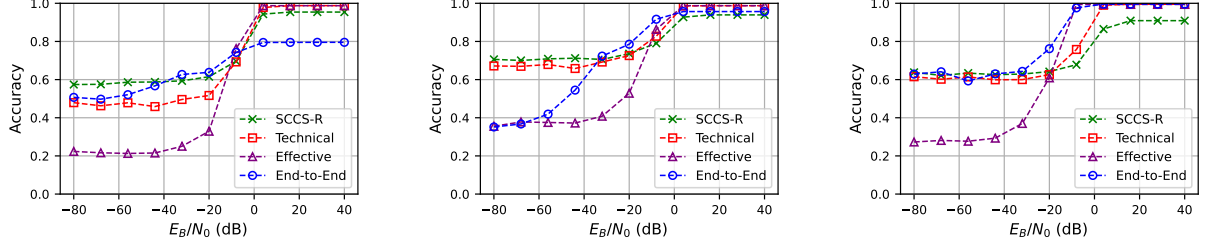


Figure 6.7: Task accuracy vs. SNR with and without reasoning.

shapes. We also see that at low SNR values, the random system seems to outperform the system with reasoning. This is likely due to the fact that, when SNR is low, the reasoning system sends badly corrupted versions of the “important” information, thus causing errors. The random system, on the other hand, is likely to send unimportant information, for which the corruption caused by the channel is not a significant factor. In this case, the decoder may make a random guess which could still be correct.

Next we compare the proposed approach to the baseline systems for the AWGN channel with BPSK modulation, where the messages are constrained to a length of two semantic values for the shape and color tasks, and a length of 10 for the speed limit sign task. These results are shown in Figure 6.8. We see that the SCCS-R system consistently performs well, both in low and high SNR scenarios, and for all tasks. The performance of the end-to-end system seems to be limited for the shape task for high SNR values compared to the other systems, likely due to the joint training process prioritizing the other tasks. The SCCS-R system does not share this limitation, as the decoders are trained separately from the encoder, allowing for individual training and more focused performance on each task. The technical baseline shows similar performance to the proposed system, however, this system sends over 6000 times as many bits for each inference by transmitting the entire 64x64x3 image made of up of 8-bit pixels, while the SCCS-R system sends only two 8-bit semantic values. We also see that the performance of the effective baseline drops of significantly in the low SNR domain, as compared to the proposed approach.

Finally, we provide similar results for the Rayleigh fading channel with 16-QAM modu-



(a) Shape classification task (b) Color classification task (c) Speed limit sign classification

Figure 6.8: Task accuracy vs. SNR for the proposed and baseline systems for the AWGN channel with BPSK modulation.

lation on the shape classification task, where the message length is again constrained to two semantic values (16 bits) for the SCCS-R system with the effective and end-to-end baselines also sending 16 bits per transmission. The results are shown in Figure 6.9. We observe largely the same results as in the AWGN scenario with the exception of the end-to-end baseline, which seems to exhibit improved performance in the low SNR domain with further limitations to the performance in the high SNR domain. As entirely new models must be trained for the end-to-end system when simulating the new channel, the random nature of the training process is likely the reason for these differences. This retraining is not necessary for our proposed approach, as it can operate on top of any traditional communication components without modifying the existing infrastructure. The performance limitation of the end-to-end system in favorable conditions is again a drawback to this approach, while the SCCS-R system does not suffer this same effect.

6.6 Chapter Summary

In this chapter we presented two main innovations for SCCS. The first was the use of the VAE and the p -Wasserstein distance for learning the domains of the CS. The use of the VAE as opposed to the basic AE allows for a probability theory-based implicit regularization of the learned domain, while incorporating the Wasserstein distance into the loss function allows the designer to specify prior knowledge about the similarity structure of the CS. The second innovation is the introduction of causal reasoning into the SCCS system, dubbed

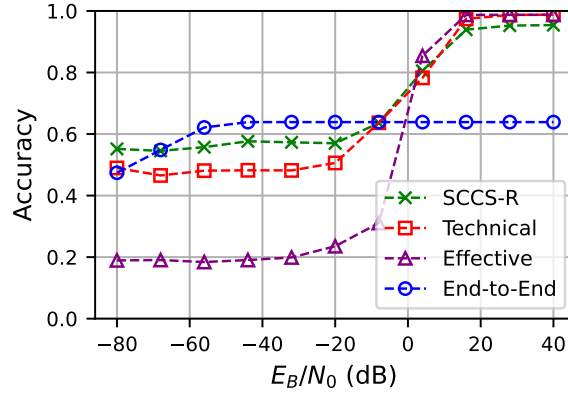


Figure 6.9: Shape classification accuracy vs SNR for the proposed (message length = 2) and baseline systems (Rayleigh fading, 16-QAM modulation).

SCCS-R, in which the estimated causal relationships between semantic and task variables are utilized to communicate more intelligently and efficiently. The problems of learning and reasoning over the semantics of data are great challenges to overcome in the development of intelligent SemCom systems, and the methods expounded here represent significant steps toward realizing these systems.

Chapter 7

Semantic Rate Distortion Theory

Despite the strong enthusiasm and research activity surrounding SemCom, a major challenge remains: the lack of a general theory for SemCom, similar to classical information theory for traditional communication [168]. In this chapter, we aim to partly fill this gap by presenting a theoretical upper bound on the rate distortion function for a general model of a goal-oriented SemCom system, as well as numerical results for a simulated practical system inspired by this theory.

7.1 Related Work

Most studies related to SemCom have focused on developing ML models for achieving SemCom, while fewer have studied the theoretical aspects [168]. Similar to the work in this chapter, Liu *et al.* study the rate distortion problem for SemCom [169]. Their model includes an unobservable semantic source and corresponding distortion measure for which they derive the achievable region. Stavrou and Kountouris assume a similar model of SemCom and derive new bounds on the rate distortion function in a goal-oriented setting [170], as well as study the effect of multiple goal-oriented distortion criteria [171]. Achievability in each of these prior works is defined as satisfying both semantic and syntactic distortion criteria, whereas we only focus on goal-oriented achievability in this letter. Most other works

concerning semantic rate distortion assume a significantly different model than the one assumed in this work, or take a more empirical approach of optimizing a neural network with a rate distortion-inspired loss function. More details on recent developments in semantic rate distortion can be found in a recent survey [168].

7.2 Chapter Contributions

In this chapter, we study the problem of goal-oriented rate distortion for SemCom. Differing from related work, we assume a model in which the encoder maps observed data to some semantic/conceptual space which is then encoded in the classical sense and used at the receiver to perform some task. In summary, the primary contributions include the following:

- Introduction of the novel notion of *distortion discrepancy* as a measure by which to characterize a goal-oriented SemCom system.
- Derivation of an upper bound on the rate distortion function for the defined semantic-functional system, which hinges on the semantic distortion threshold and asymptotic distortion discrepancy.
- Numerical experiments highlighting the performance of a practical “minimum discrepancy” code, achieving optimal performance with respect to the expected distortion in certain scenarios and demonstrating up to 54.9% performance gain as compared to a naive scheme.

7.3 Problem Definition

The system model is illustrated in Figure 7.1. We consider a source generating sequences of independent and identically distributed (i.i.d.) symbols $\{X_i\}_{i=1}^n$ where $X_i \sim p(x)$ for x in some discrete alphabet $\mathcal{X}$. Define the goal (or task) of communication as a function $f : \mathcal{X} \rightarrow \mathcal{U}$ where $\mathcal{U}$ is a discrete alphabet of task symbols $u \in \mathcal{U}$ with corresponding bounded distortion function $d(u, \hat{u}) : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}_+$ mapping task variable pairs to the set

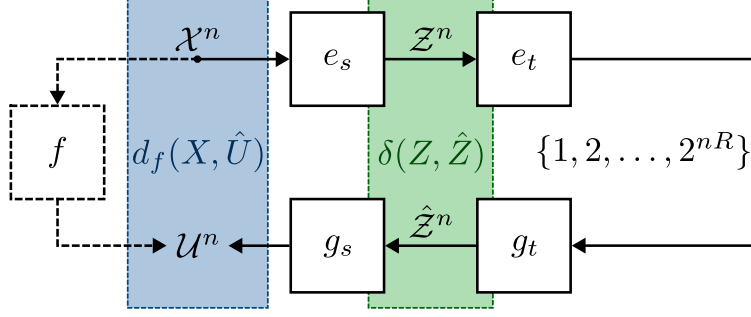


Figure 7.1: Flowchart of a $(2^{nR}, n)$ semantic-functional code.

of non-negative real numbers. For n -sequences of symbols $X^n = \{X_i\}_{i=1}^n$, the task function operates on a symbol-by-symbol basis such that $U^n = \{f(X_i)\}_{i=1}^n$. Moreover, define the task-specific distortion $d_f : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}_+$ induced by f , where $d_f(X, \hat{U}) = d(f(X), \hat{U})$. For sequences, we define

$$d_f(X^n, \hat{U}^n) = \frac{1}{n} \sum_{i=1}^n d_f(X_i, \hat{U}_i). \quad (7.1)$$

We define a semantic-functional code (SFC) with rate R as four encoding/decoding functions. First, the semantic encoder $e_s : \mathcal{X}^n \rightarrow \mathcal{Z}^n$ maps a sequence of source symbols to a sequence of semantic representations from a discrete set $\mathcal{Z}$. The technical encoder $e_t : \mathcal{Z}^n \rightarrow \{1, 2, \dots, 2^{nR}\}$ maps the sequence Z^n to one of 2^{nR} distinct codewords, and the technical decoder $g_t : \{1, 2, \dots, 2^{nR}\} \rightarrow \hat{\mathcal{Z}}^n$ takes this codeword to a sequence of approximate semantic representations $\hat{Z}^n$. Finally, the semantic decoder $g_s : \hat{\mathcal{Z}}^n \rightarrow \mathcal{U}^n$ maps the approximate semantic representation to a sequence of task variables $\hat{U}^n$.

In addition to the functional distortion d_f , we can also consider a semantic distortion function $\delta : \mathcal{Z} \times \hat{\mathcal{Z}} \rightarrow \mathbb{R}_+$ mapping pairs of true and recovered semantic representations to some (bounded) positive value. Similar to d_f , for sequences we define the semantic distortion as

$$\delta(Z^n, \hat{Z}^n) = \frac{1}{n} \sum_{i=1}^n \delta(Z_i, \hat{Z}_i). \quad (7.2)$$

We assume that each semantic representation Z is a M -dimensional vector lying in some semantic space $\mathcal{Z} \subset \mathbb{R}^M$. This space is presumed to represent the meaning of the data, as in a CS model presented in [31]. This space is continuous in general, while the sets $\mathcal{Z}, \hat{\mathcal{Z}} \subset \mathcal{Z}$

are discrete, and thus we can interpret these sets as discretized versions of the CS, each containing $|\mathcal{X}|$ and 2^{nR} points, respectively.

As our primary objective is to perform some task defined as the function f , we define the distortion of a $(2^{nR}, n)$ SFC as the expected value of the functional distortion, defined as

$$D_F = E \left[d_f(X^n, \hat{U}^n) \right], \quad (7.3)$$

where the expectation is with respect to the source sequence X^n . A rate distortion pair (R, D_F) is said to be achievable if there exists a sequence of $(2^{nR}, n)$ SFCs with

$$\lim_{n \rightarrow \infty} E[d_f(X^n, \hat{U}^n)] \leq D_F. \quad (7.4)$$

The rate distortion region is then defined as the closure of the set of achievable rate distortion pairs (R, D_F) , and the rate distortion function $R_F(D_F)$ is defined as the infimum of rates R such that (R, D_F) is in the rate distortion region.

7.4 Supporting Results

In this section, we provide some results necessary for the proof of our main theorem. First, we briefly restate the achievability theorem from classical rate distortion theory; for more details see [7]. Consider an i.i.d. source X as defined above, as well as a bounded distortion function $d : \mathcal{X} \times \hat{\mathcal{X}} \rightarrow \mathbb{R}_+$ with sequence distortion defined as the mean distortion, similar to the definitions above. Define a $(2^{nR}, n)$ rate distortion code as consisting of the encoding function $e : \mathcal{X}^n \rightarrow \{1, 2, \dots, 2^{nR}\}$ and the decoding function $g : \{1, 2, \dots, 2^{nR}\} \rightarrow \hat{\mathcal{X}}^n$. A rate distortion pair (R, D) is said to be achievable if there exists a sequence of $(2^{nR}, n)$ codes with

$$\lim_{n \rightarrow \infty} E \left[d_X(X^n, \hat{X}^n) \right] \leq D, \quad (7.5)$$

and define the rate distortion function $R(D)$ as the infimum of rates R such that (R, D) is achievable.

Theorem 4 (Classical rate distortion [7]). *The rate distortion function for an i.i.d. source X with distribution $p(X)$ and bounded distortion function $d(x, \hat{x})$ is given by*

$$R(D) = \min_{p(\hat{x}|x): \sum_{(x, \hat{x})} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \quad (7.6)$$

Proof. See chapter 10 of [7]. ■

In relation to the notion of SemCom, this result states that the minimum rate at which one can communicate given an acceptable level of *technical* distortion is equal to the information rate distortion function for a given source. Next, we provide an asymptotic result regarding the average sequence distortion.

Lemma 3 (Asymptotic sequence distortion). *For an i.i.d. source X with a bounded distortion function $d(X, \hat{X})$ where $\hat{X} = h(X)$ for some deterministic function h , we have*

$$\lim_{n \rightarrow \infty} E \left[d(X^n, \hat{X}^n) \right] = E \left[d(X, \hat{X}) \right]. \quad (7.7)$$

Proof. Since $\hat{X}$ is a deterministic function of X , we can simplify the notation and call $d(X, \hat{X}) = d(X)$. Then,

$$\lim_{n \rightarrow \infty} E_{X^n} [d(X^n)] = \lim_{n \rightarrow \infty} \frac{1}{n} E_{X^n} \left[\sum_{i=1}^n d(X_i) \right]. \quad (7.8)$$

Now let $\mathbf{d}$ be a random vector whose elements are the i.i.d. random variables $\{d(X_i)\}_{i=1}^n$. Then we have

$$\lim_{n \rightarrow \infty} \frac{1}{n} E_{X^n} \left[\sum_{i=1}^n d(X_i) \right] = \lim_{n \rightarrow \infty} \frac{1}{n} E_{X^n} [\mathbf{1}^\top \mathbf{d}] \quad (7.9)$$

$$= \lim_{n \rightarrow \infty} \frac{1}{n} \mathbf{1}^\top E_{X^n} [\mathbf{d}] \quad (7.10)$$

$$= \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E_X [d(X)] \quad (7.11)$$

$$= E_X [d(X)]. \quad (7.12)$$

■

We now define the notion of *distortion discrepancy* Δ as

$$\Delta = E \left[\left(d_f(X^n, \hat{U}^n) - \delta(Z^n, \hat{Z}^n) \right)^2 \right], \quad (7.13)$$

where d_f , δ are as defined in the previous section and the expectation is over X^n . The discrepancy quantifies how much the semantic and functional distortion measures differ on average for a sequence length n . Intuitively, this can be thought of as how well differences in meaning (semantic distortion) capture differences in the current task (functional distortion) and vice versa. The last result of this section looks at asymptotic discrepancy.

Lemma 4 (Asymptotic distortion discrepancy). *For Δ defined in (7.13), the limit as $n \rightarrow \infty$ is given by*

$$\lim_{n \rightarrow \infty} \Delta = \left(E \left[d_f(X, \hat{U}) - \delta(Z, \hat{Z}) \right] \right)^2. \quad (7.14)$$

Proof. By definition, we have

$$\lim_{n \rightarrow \infty} \Delta = \lim_{n \rightarrow \infty} E_{X^n} \left[\left(d_f(X^n, \hat{U}^n) - \delta(Z^n, \hat{Z}^n) \right)^2 \right] \quad (7.15)$$

$$= \lim_{n \rightarrow \infty} E_{X^n} \left[\left(\frac{1}{n} \sum_{i=1}^n d_f(X_i, \hat{U}_i) - \delta(Z_i, \hat{Z}_i) \right)^2 \right]. \quad (7.16)$$

We make use of the Dominated Convergence Theorem (DCT) [172] to show that we can take the limit inside the expectation. First, define the sequence of random variables $\{Y_n\}_{n=1}^{\infty}$ with

$$Y_n = \left(\frac{1}{n} \sum_{i=1}^n d_f(X_i, \hat{U}_i) - \delta(Z_i, \hat{Z}_i) \right)^2. \quad (7.17)$$

We need to show that 1) there exists a function that bounds each term Y_n in the sequence, and 2) the sequence converges.

Dominated. By assumption, d_f and δ are both bounded. Call the greater of these two bounds C , which is finite. Since both terms are positive, the difference of the two is also

bounded by C , and the average of any $\{d_f - \delta\}$ terms is also bounded by C . The square will then be bounded by C^2 , which is a constant universal bound for every term in the sequence.

Convergent. As $n \rightarrow \infty$, the average inside the square converges by the Weak Law of Large Numbers (WLLN) to

$$E_X[d_f(X, \hat{U}) - \delta(Z, \hat{Z})], \quad (7.18)$$

which is guaranteed to exist as d_f and δ are both bounded. As the square function is a continuous function for all real numbers, this implies that the sequence $\{Y_n\}$ converges to

$$\left(E_X[d_f(X, \hat{U}) - \delta(Z, \hat{Z})]\right)^2. \quad (7.19)$$

Thus, by the DCT, we can take the limit inside of the expectation and (7.16) becomes

$$E_{X^n} \left[\lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{i=1}^n d_f(X_i, \hat{U}_i) - \delta(Z_i, \hat{Z}_i) \right)^2 \right]. \quad (7.20)$$

Applying the WLLN once more, we have

$$\lim_{n \rightarrow \infty} \Delta = E_{X^n} \left[\left(E_X[d_f(X, \hat{U}) - \delta(Z, \hat{Z})] \right)^2 \right], \quad (7.21)$$

and as the inner expectation is non-random, we have the result:

$$\lim_{n \rightarrow \infty} \Delta = \left(E \left[d_f(X, \hat{U}) - \delta(Z, \hat{Z}) \right] \right)^2. \quad (7.22)$$

■

7.5 Semantic-Functional Rate Distortion

We now present our main result as an upper bound on the semantic-functional rate distortion function.

Theorem 5 (Semantic-Functional Rate Distortion). *Given any semantic encoder e_s , choose*

R for a sequence of $(2^{nR}, n)$ semantic-functional codes according to (7.6) in Theorem 4 for the semantic “source” $Z = e_s(X)$ and semantic distortion threshold D_S . Then for Δ defined in (7.13), the rate distortion pair $(R, D_S + \sqrt{\lim_{n \rightarrow \infty} \Delta})$ is achievable. Thus, choosing functional distortion threshold $D_F = D_S + \sqrt{\lim_{n \rightarrow \infty} \Delta}$, we have an upper bound on the semantic rate distortion function

$$R_F(D_F) \leq R. \quad (7.23)$$

Proof. From the definition of discrepancy in (7.13):

$$\Delta = E_{X^n} \left[\left(d_f(X^n, \hat{U}^n) - \delta(Z^n, \hat{Z}^n) \right)^2 \right]. \quad (7.24)$$

Taking the limit and applying Lemma 4, we have

$$\lim_{n \rightarrow \infty} \Delta = \left(E \left[d_f(X, \hat{U}) - \delta(Z, \hat{Z}) \right] \right)^2. \quad (7.25)$$

Rearranging, we get

$$E[d_f(X, \hat{U})] = E[\delta(Z, \hat{Z})] + \sqrt{\lim_{n \rightarrow \infty} \Delta} \quad (7.26)$$

Now applying Lemma 3, we can write

$$\lim_{n \rightarrow \infty} E[d_f(X^n, \hat{U}^n)] = \lim_{n \rightarrow \infty} E[\delta(Z^n, \hat{Z}^n)] + \sqrt{\lim_{n \rightarrow \infty} \Delta}. \quad (7.27)$$

By Theorem 4, we know that there exists a sequence of technical codes $\{e_t, g_t\}$ with minimal rate $R = R(D_S)$ for which the asymptotic semantic distortion $\lim_{n \rightarrow \infty} E[\delta(Z, \hat{Z})]$ is upper bounded by D_S . Fixing these codes, (7.27) becomes

$$\lim_{n \rightarrow \infty} E \left[d_f(X^n, \hat{U}^n) \right] \leq D_S + \sqrt{\lim_{n \rightarrow \infty} \Delta}. \quad (7.28)$$

This implies that the rate pair (R, D_F) with

$$D_F = D_S + \sqrt{\lim_{n \rightarrow \infty} \Delta} \quad (7.29)$$

is achievable, and thus R is an upper bound for $R_F(D_F)$. ■

7.6 Numerical Results

In this section we perform numerical experiments to illustrate the practical implications of the theoretical results¹. We simulate an SFC by first fixing the semantic encoder e_s . For the purposes of simulation, we sample an M -dimensional vector Z_i with i.i.d. $\mathcal{N}(0, 1)$ elements and assign this semantic representation to the source symbol X_i .

For the technical code $\{e_t, g_t\}$ with rate R , we implement Lloyd’s algorithm to find a set $\hat{\mathcal{Z}}$ of 2^{nR} semantic codewords (also M -dimensional vectors) that minimize the expected semantic distortion (Euclidean distance in $\mathbb{R}^M$) for the representation set $\mathcal{Z}$ with corresponding distribution $p(z)$. We then implement three different semantic decoders:

- *Minimum- Δ ($\Delta_{\min}$) decoder*: chooses the mapping g_s which minimizes the distortion discrepancy Δ for the source $X \sim p(x)$ and coding functions $\{e_s, e_t, g_t\}$. Only requires indirect knowledge the of task and distortion in the form of the discrepancy function.
- *Naive decoder*: randomly selects a mapping g_s from semantic codewords $\hat{\mathcal{Z}}$ to task variables $\hat{U}$.
- *Oracle decoder*: assumes full knowledge of d_f and chooses the g_s which minimizes this distortion for the source $X \sim p(x)$ and coding functions $\{e_s, e_t, g_t\}$.

Monte Carlo simulations were performed to estimate $E[\delta]$, $E[\Delta]$ and $E[d_f]$ for different parameter values. The results are shown in Table 7.1. Functional distortion values d_f are chosen randomly for each $(x, \hat{u})$ pair and are fixed across all experiments. The first data row illustrates the “base” case, and we see that the $\Delta_{\min}$ code performs close to the oracle code, with only a 3% increase in $E[d_f]$. Meanwhile, the $\Delta_{\min}$ code shows a 51.8% decrease in $E[d_f]$ relative to the naive code.

¹The numerical simulations were implemented using Python, and all project code can be found at <https://github.com/dw-610/semantic-rate-dist-code>. See Appendix A for more details.

Table 7.1: Simulation results for various parameter combinations. Dashes indicate that the value is the same as for the base case.

Parameters						Minimum- Δ g_s			Naive g_s		Oracle g_s	
$ \mathcal{X} $	$p(x)$	M	$ \mathcal{U} $	n	R	$E[\delta]$	$E[\Delta]$	$E[d_f(X^n, \hat{U}^n)]$	$E[\Delta]$	$E[d_f(X^n, \hat{U}^n)]$	$E[\Delta]$	$E[d_f(X^n, \hat{U}^n)]$
12	u	2	4	1	3	0.081	0.049	0.239	0.258	0.496	0.054	0.232
24	-	-	8	-	-	0.379	0.032	0.395	0.141	0.510	0.135	0.220
-	-	4	-	-	-	0.463	0.101	0.432	0.247	0.492	0.405	0.232
-	-	4	-	-	4	0.000	0.043	0.169	0.193	0.375	0.043	0.169
-	-	4	-	-	1	1.408	1.151	0.540	1.368	0.501	1.384	0.380
-	-	-	-	3	-	0.081	0.033	0.236	0.201	0.496	0.033	0.232
-	r	-	-	-	-	0.080	0.090	0.305	0.232	0.489	0.091	0.307

We draw out additional insights by altering the base case. By increasing $|\mathcal{X}|$ to 24 and increasing $|\mathcal{U}|$ to 8, we see that the $\Delta_{\min}$ code now performs considerably worse than the oracle code, with a 79.5% increase in $E[d_f]$. This is due to the fact that $E[\delta]$ is significantly higher in this case (over 300% greater than the base setup), due to the increased compression of the source. Since δ values are larger, to minimize Δ , the $\Delta_{\min}$ decoder must map to $\hat{U}$ symbols with large d_f values, leading to an increase in expected distortion. However, it still outperforms the naive code, with a 22.5% decrease in $E[d_f]$.

This behavior is also highlighted in the cases where $M = 4$, as a higher-dimensional CS leads to increased semantic distances. When $R = 1$ (single-bit compression), semantic distortion increases over 1600% compared to the base scenario, resulting in a 42% greater expected distortion for the $\Delta_{\min}$ code compared to the oracle. However, when $R = 4$, the codebook can perfectly describe the original semantic representations and $E[\delta]$ goes to 0. This results in identical performance for the $\Delta_{\min}$ and oracle codes, with the naive code still performing significantly worse. This illustrates that, generally, as the average semantic distortion decreases, the minimum-discrepancy code will converge to the optimal code.

Two additional scenarios are shown in the last two rows of Table 7.1. First, when $n = 3$, we see that the performance is similar to that of the base case, and thus the $\Delta_{\min}$ code also performs well with increasing block length. For all of the previous results, a uniform distribution $p(x) = 1/|\mathcal{X}|$ was assumed. The final row of Table 7.1 shows the results for a distribution chosen at random. We see that the performance remains comparable to that of the oracle system.

7.7 Chapter Summary

In this chapter, we provided a new perspective on the problem of goal-oriented semantic rate distortion, and derived an upper bound on the rate distortion function based on the novel notion of distortion discrepancy. Numerical experiments with a practical scheme based on the theoretical results demonstrated performance close to the optimal coding scheme given certain configurations.

Chapter 8

Conclusion and Directions for Future Work

In this dissertation, we have presented, for the first time, a method of SemCom based on the theory of CSs, termed SCCS. The proposed method aims to overcome some of the limitations of current approaches by modeling meaning in a general, expressive, and interpretable manner. In this concluding chapter, we summarize the work presented in the previous chapters and discuss some promising directions for future work.

8.1 Summary

Chapter 1 motivated the need for SemCom and outlined the primary contributions of this dissertation. In Chapter 2, we provided a survey of literature related to SemCom and discussed the primary existing approaches to modeling meaning. This survey exposed some of the key limitations of the existing approaches. In Chapter 3, we presented a preliminary investigation into an alternative way of modeling meaning and the corresponding SemCom system leveraging this method, termed *semantic communication with conceptual spaces*. Simulation results from this initial investigation demonstrated the potential of the SCCS framework, and thus in Chapter 4 we studied this approach in far more detail. We formally defined

components of the system, derived semantic performance bounds given the SCCS framework, and conducted more extensive simulations which further confirmed the potential of the proposed framework and revealed additional insights. In Chapter 5, we addressed one of the primary challenges facing the SCCS approach by presenting an AE-based learning architecture for obtaining the domains of a CS. This architecture was improved in Chapter 6, in which we additionally proposed a causal reasoning-based mechanism for equipping the SCCS system with improved intelligence regarding the task at hand. Finally, in Chapter 7, we investigated the theoretical implications of a goal-oriented SCCS framework and derived an upper bound on the goal-oriented semantic rate distortion function. Numerical simulations of a practical coding scheme based on this theory indicated nearly optimal performance under certain parameter configurations.

8.2 Directions for Future Work

There are several interesting ways in which this work can be extended. Some promising directions include:

- *Practical implementations.* The developments presented throughout this dissertation remain purely theoretical, with practical results being demonstrated via simulations. An interesting and important engineering problem is to begin to translate these theoretical developments into real-world practical systems which can be implemented in future communications systems to truly reap the benefits of SemCom.
- *Studying the effect of mismatched knowledge bases.* There are multiple ways in which one CS can differ from another. The concept regions could be slightly different from one another, or perhaps the entire conceptual space might be rotated or translated. In this work we focused on the case where the transmitter and receiver share an identical knowledge base, which in general may not be true. A potential solution for this challenge is to employ context-based transformations as described in Section 4.5.2.
- *Improved CS learning.* The domain learning techniques presented in Chapters 5 and 6

only learn a single domain of the space at a time, where it would be beneficial to be able to learn the entire CS with one model/framework. Additionally, improving this conceptual learning framework could have a far-reaching impact beyond the field of SemCom, with the potential to translate into a method for training higher-intelligence artificial models that truly understand semantics.

- *Alternative causal reasoning formulations.* With regard to the causal reasoning approach presented in Chapter 6, in addition to reasoning over the causal relationships between semantic and task variables, another interesting approach is to reason over the causal relationships between the raw, observed data and the semantic variables to extract even greater insights. This could lead to a causally-aware “semantic pre-coding” where the unimportant raw data is discarded before being passed to the semantic encoder for improved efficiency.
- *Further theoretical investigations.* In Chapter 7, only an upper bound was presented, where it would clearly be beneficial to fully characterize the rate distortion function. Moreover, it would be interesting to further explore the relationship between the semantic distortion, functional distortion, and distortion discrepancy, with the potential to uncover efficient goal-oriented semantic coding schemes for future communications.

Bibliography

- [1] Ericsson, “Ericsson Mobility Report,” 6 2024. [Online] <https://www.ericsson.com/en/mobility-report>.
- [2] J. Li, “Automatic Semantic Analysis Framework of Dickinson’s Portfolio based on Character Recognition and Artificial Intelligence,” in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1660–1664, 2020.
- [3] Y. Hua and J. Du, “Deep Semantic Correlation with Adversarial Learning for Cross-Modal Retrieval,” in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 256–259, 2019.
- [4] 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.
- [5] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, T. Soleymani, B. Soret, and K. H. Johansson, “Semantic Communications in Networked Systems,” 2021.
- [6] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [7] T. M. Cover and J. A. Thomas, *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 7 2006.
- [8] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press, 1949.

- [9] 3GPP, “Release 16 Description; Summary of Rel-16 Work Items,” 06 2021. Version 16.0.0.
- [10] C. Chaccour, M. N. Soorki, W. Saad, M. Bennis, P. Popovski, and M. Debbah, “Seven Defining Features of Terahertz (THz) Wireless Systems: A Fellowship of Communication and Sensing,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 967–993, 2022.
- [11] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [12] R. Carnap and Y. Bar-Hillel, “An Outline of a Theory of Semantic Information,” *Journal of Symbolic Logic*, vol. 19, no. 3, pp. 230–232, 1954.
- [13] L. Floridi, “Outline of a Theory of Strongly Semantic Information,” *Minds and Machines*, vol. 14, 05 2004.
- [14] S. D’Alfonso, “On Quantifying Semantic Information,” *Information*, vol. 2, no. 1, pp. 61–101, 2011.
- [15] 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.
- [16] Y. Xiao, X. Zhang, Y. Li, G. Shi, and T. Başar, “Rate-distortion theory for strategic semantic communication,” in *2022 IEEE Information Theory Workshop (ITW)*, pp. 279–284, 2022.
- [17] J. Choi, S. W. Loke, and J. Park, “A unified approach to semantic information and communication based on probabilistic logic,” *IEEE Access*, vol. 10, pp. 129806–129822, 2022.
- [18] W. Yang, Z. Q. Liew, W. Y. B. Lim, Z. Xiong, D. Niyato, X. Chi, X. Cao, and

- K. B. Letaief, “Semantic Communication Meets Edge Intelligence,” *IEEE Wireless Communications*, vol. 29, no. 5, pp. 28–35, 2022.
- [19] J. Kang, H. Du, Z. Li, Z. Xiong, S. Ma, D. Niyato, and Y. Li, “Personalized Saliency in Task-Oriented Semantic Communications: Image Transmission and Performance Analysis,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 186–201, 2023.
- [20] A. Li, X. Wei, D. Wu, and L. Zhou, “Cross-modal semantic communications,” *IEEE Wireless Communications*, vol. 29, no. 6, pp. 144–151, 2022.
- [21] R. Rada, H. Mili, E. Bicknell, and M. Blettner, “Development and application of a metric on semantic nets,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.
- [22] P. Resnik, “Using Information Content to Evaluate Semantic Similarity in a Taxonomy,” in *International Joint Conference for Artificial Intelligence (IJCAI-95)*, pp. 448–453, 1995.
- [23] J. J. Jiang and D. W. Conrath, “Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy,” *CoRR*, vol. cmp-lg/9709008, 1997.
- [24] W. Tong, F. Liu, Z. Sun, Y. Yang, and C. Guo, “Image semantic communications: An extended rate-distortion theory based scheme,” in *2022 IEEE Globecom Workshops (GC Wkshps)*, pp. 1723–1728, 2022.
- [25] Z. Li, J. Zhou, G. Nan, Z. Li, Q. Cui, and X. Tao, “Sembat: Physical layer black-box adversarial attacks for deep learning-based semantic communication systems,” in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pp. 1–5, 2022.
- [26] Y. Shao and D. Gündüz, “Semantic communications with discrete-time analog transmission: A papr perspective,” *IEEE Wireless Communications Letters*, pp. 1–1, 2022.

- [27] 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.
- [28] Y. Wang, Z. Gao, D. Zheng, S. Chen, D. Gunduz, and H. V. Poor, “Transformer-Empowered 6G Intelligent Networks: From Massive MIMO Processing to Semantic Communication,” *IEEE Wireless Communications*, pp. 1–9, 2022.
- [29] M. K. Farshbafan, W. Saad, and M. Debbah, “Common language for goal-oriented semantic communications: A curriculum learning framework,” in *ICC 2022 - IEEE International Conference on Communications*, pp. 1710–1715, 2022.
- [30] M. Kountouris and N. Pappas, “Semantics-Empowered Communication for Networked Intelligent Systems,” *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.
- [31] P. Gärdenfors, *Conceptual Spaces: The Geometry of Thought*. Massachusetts Institute of Technology, 2000.
- [32] P. Gärdenfors, *The Geometry of Meaning: Semantics Based on Conceptual Spaces*. Massachusetts Institute of Technology, 2014.
- [33] M. Ouksel and C. Naiman, “Toward the design of a semantic communication protocol in heterogeneous database systems,” in *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1271–1276 vol.2, 1992.
- [34] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Scientific American*, pp. 34–43, 5 2001.
- [35] V. Rodoplu and S. S. Vadvalkar, “Challenges and Directions for Semantic Communication,” in *International Conference on Semantic Computing (ICSC 2007)*, pp. 290–294, 2007.

- [36] B. Juba, “Compatibility among diversity Foundations, lessons, and directions of semantic communication,” in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 458–463, 2013.
- [37] B. H. Juang, “Quantification and Transmission of Information and Intelligence—History and Outlook [DSP History],” *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 90–101, 2011.
- [38] W. Tong and G. Y. Li, “Nine Challenges in Artificial Intelligence and Wireless Communications for 6G,” 2021.
- [39] E. C. Strinati and S. Barbarossa, “6G Networks: Beyond Shannon Towards Semantic and Goal-Oriented Communications,” 2021.
- [40] X. Luo, H.-H. Chen, and Q. Guo, “Semantic Communications: Overview, Open Issues, and Future Research Directions,” *IEEE Wireless Communications*, pp. 1–10, 2022.
- [41] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, “What Is Semantic Communication? A View on Conveying Meaning in the Era of Machine Intelligence,” *Journal of Communications and Information Networks*, vol. 6, 12 2021.
- [42] B. Juba, *Universal Semantic Communication*. Springer Berlin Heidelberg, 2011.
- [43] J. Dewey, *Experience and Nature*. Lectures upon the Paul Carus foundation, Norton, 1929.
- [44] L. Wittgenstein, *Philosophical Investigations*. Oxford: Basil Blackwell, 1953.
- [45] J. Delgado-Frias and W. Moore, “A semantic network architecture for artificial intelligence processing,” in *[Proceedings 1989] IEEE International Workshop on Tools for Artificial Intelligence*, pp. 162–167, 1989.
- [46] J. Mertoguno and W. Lin, “Distributed knowledge-base: adaptive multi-agents approach,” in *Proceedings IEEE International Joint Symposia on Intelligence and Systems*, pp. 76–83, 1996.

- [47] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, “A Survey on Knowledge Graphs: Representation, Acquisition, and Applications,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2022.
- [48] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.
- [49] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, “Model-Based Deep Learning,” 2020.
- [50] 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.
- [51] 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.
- [52] 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.
- [53] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, “Age of Information: An Introduction and Survey,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.
- [54] 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.
- [55] P. Basu, J. Bao, M. Dean, and J. Hendler, “Preserving quality of information by using semantic relationships,” in *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 58–63, 2012.

- [56] A. d. Garcez and L. C. Lamb, “Neurosymbolic AI: The 3rd Wave,” *Artificial Intelligence Review*, 3 2023.
- [57] B. Swartout, R. Patil, K. Knight, and T. Russ, “Toward Distributed Use of Large-Scale Ontologies,” in *Ontological Engineering, AAAI-97 Spring Symposium Series*, pp. 138–148, 1997.
- [58] D. Sathya and K. Uthayan, “Proposal for semantic metric to assess the quality of ontologies,” in *2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies*, pp. 754–756, 2011.
- [59] A. Sheth, C. Henson, and S. S. Sahoo, “Semantic Sensor Web,” *IEEE Internet Computing*, vol. 12, no. 4, pp. 78–83, 2008.
- [60] A. Gyrard, C. Bonnet, and K. Boudaoud, “Enrich machine-to-machine data with semantic web technologies for cross-domain applications,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 559–564, 2014.
- [61] S. Chun, S. Seo, B. Oh, and K.-H. Lee, “Semantic description, discovery and integration for the Internet of Things,” in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 272–275, 2015.
- [62] L. B. Bhajantri and R. Pundalik, “Data processing in semantic sensor web: A survey,” in *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 166–170, 2017.
- [63] D. Schachinger and W. Kastner, “Semantic interface for machine-to-machine communication in building automation,” in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pp. 1–9, 2017.
- [64] E. Lakka, N. E. Petroulakis, G. Hatzivasilis, O. Soultatos, M. Michalodimitrakis, U. Rak, K. Waledzik, D. Anicic, and V. Kulkarni, “End-to-End Semantic Interoperability Mechanisms for IoT,” in *2019 IEEE 24th International Workshop on Com-*

- puter Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–6, 2019.
- [65] M. Jeong, B. Kim, and G. Lee, “Semantic-oriented error correction for spoken query processing,” in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721)*, pp. 156–161, 2003.
 - [66] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, “ERNIE: Enhanced Language Representation with Informative Entities,” 2019.
 - [67] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, “Explainable Reasoning over Knowledge Graphs for Recommendation,” 2018.
 - [68] E. Aumayr, M. Wang, and A.-M. Bosneag, “Probabilistic Knowledge-Graph based Workflow Recommender for Network Management Automation,” in *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1–7, 2019.
 - [69] H. He, A. Balakrishnan, M. Eric, and P. Liang, “Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings,” 2017.
 - [70] 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.
 - [71] Y. Wei, J. Luo, and H. Xie, “KGRL: An OWL2 RL Reasoning System for Large Scale Knowledge Graph,” in *2016 12th International Conference on Semantics, Knowledge and Grids (SKG)*, pp. 83–89, 2016.
 - [72] D. Zheng, X. Song, C. Ma, Z. Tan, Z. Ye, J. Dong, H. Xiong, Z. Zhang, and G. Karypis, “DGL-KE: Training Knowledge Graph Embeddings at Scale,” 2020.
 - [73] H. Zhu, X. Wang, Y. Jiang, H. Fan, B. Du, and Q. Liu, “FTRLIM: Distributed Instance

- Matching Framework for Large-Scale Knowledge Graph Fusion,” in *Entropy*, MDPI, 2021.
- [74] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneijder, and L. A. Stein, “OWL Web Ontology Language Reference,” Recommendation, World Wide Web Consortium (W3C), 2 2004. See <http://www.w3.org/TR/owl-ref/>.
- [75] World Wide Web Consortium, “RDFa Core 1.1 - Third Edition,” 3 2015. [Online] <https://www.w3.org/TR/rdfa-core/>.
- [76] 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)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.
- [77] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating Embeddings for Modeling Multi-relational Data,” in *Advances in Neural Information Processing Systems* (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.
- [78] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox, “Natural Language Processing Advancements By Deep Learning: A Survey,” 2020.
- [79] M. Guo, T. Xu, J. Liu, Z. Liu, P. Jiang, T. Mu, S. Zhang, R. R. Martin, M. Cheng, and S. Hu, “Attention Mechanisms in Computer Vision: A Survey,” *CoRR*, vol. abs/2111.07624, 2021.
- [80] Y. Lu, Y. Shi, G. Jia, and J. Yang, “A new method for semantic consistency verification of aviation radiotelephony communication based on LSTM-RNN,” in *2016 IEEE International Conference on Digital Signal Processing (DSP)*, pp. 422–426, 2016.

- [81] 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.
- [82] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, “SEED: Semantics Enhanced Encoder-Decoder Framework for Scene Text Recognition,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13525–13534, 2020.
- [83] H. Tong, Z. Yang, S. Wang, Y. Hu, W. Saad, and C. Yin, “Federated Learning based Audio Semantic Communication over Wireless Networks,” in *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2021.
- [84] 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.
- [85] M. Sana and E. C. Strinati, “Learning Semantics: An Opportunity for Effective 6G Communications,” in *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, pp. 631–636, 2022.
- [86] K. Lu, R. Li, X. Chen, Z. Zhao, and H. Zhang, “Reinforcement Learning-powered Semantic Communication via Semantic Similarity,” 2021.
- [87] Y. Wang, M. Chen, W. Saad, T. Luo, S. Cui, and H. V. Poor, “Performance Optimization for Semantic Communications: An Attention-based Learning Approach,” in *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2021.
- [88] F. Lotfi, O. Semiari, and W. Saad, “Semantic-Aware Collaborative Deep Reinforcement Learning Over Wireless Cellular Networks,” *CoRR*, vol. abs/2111.12064, 2021.
- [89] W. J. Yun, B. Lim, S. Jung, Y.-C. Ko, J. Park, J. Kim, and M. Bennis, “Attention-based Reinforcement Learning for Real-Time UAV Semantic Communication,” in *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–6, 2021.

- [90] S. Dong, P. Wang, and K. Abbas, “A survey on deep learning and its applications,” *Computer Science Review*, vol. 40, p. 100379, 2021.
- [91] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.
- [92] Z. Li, W. Yang, S. Peng, and F. Liu, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” *CoRR*, vol. abs/2004.02806, 2020.
- [93] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
- [94] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, 2017.
- [95] A. Halevy, P. Norvig, and F. Pereira, “The Unreasonable Effectiveness of Data,” *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.
- [96] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, “Aster: An attentional scene text recognizer with flexible rectification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2035–2048, 2019.
- [97] P. Koehn, “Europarl: A Parallel Corpus for Statistical Machine Translation,” in *Proceedings of Machine Translation Summit X: Papers*, (Phuket, Thailand), pp. 79–86, 9 2005.
- [98] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *CoRR*, vol. abs/1512.03385, 2015.
- [99] M. Schuster and K. Paliwal, “Bidirectional recurrent neural networks,” *Signal Processing, IEEE Transactions on*, vol. 45, pp. 2673 – 2681, 12 1997.

- [100] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, “CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997, 2017.
- [101] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “A Brief Survey of Deep Reinforcement Learning,” *CoRR*, vol. abs/1708.05866, 2017.
- [102] M. L. Littman, I. Ajunwa, G. Gerger, C. Boutilier, M. Currie, F. Doshi-Velez, G. Hadfield, M. C. Horowitz, C. Isbell, H. Kitano, K. Levy, T. Lyons, M. Mitchell, J. Shah, S. Sloman, S. Vallor, and T. Walsh, “Gathering Strength, Gathering Storms: The One Hundred Year Study on Artificial Intelligence (AI100) 2021 Study Panel Report,” 2021.
- [103] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-Critical Sequence Training for Image Captioning,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1179–1195, 2017.
- [104] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. N. Foerster, and S. Whiteson, “QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning,” *CoRR*, vol. abs/1803.11485, 2018.
- [105] E. Uysal, O. Kaya, S. Baghaee, and H. B. Beytur, “Age of Information in Practice,” 2021.
- [106] H. B. Beytur, S. Baghaee, and E. Uysal, “Towards AoI-aware Smart IoT Systems,” in *2020 International Conference on Computing, Networking and Communications (ICNC)*, pp. 353–357, 2020.
- [107] O. Ayan, M. Vilgelm, M. Klügel, S. Hirche, and W. Kellerer, “Age-of-Information vs. Value-of-Information Scheduling for Cellular Networked Control Systems,” *CoRR*, vol. abs/1903.05356, 2019.

- [108] A. Molin, H. Esen, and K. H. Johansson, “Scheduling networked state estimators based on Value of Information,” *Automatica*, vol. 110, p. 108578, 2019.
- [109] B. T. Bacinoglu, Y. Sun, and E. Uysal, “On the Trackability of Stochastic Processes Based on Causal Information,” in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2228–2233, 2020.
- [110] J. Dommel, D. Wieruch, Z. Utkovski, and S. Stańczak, “A Semantics-Aware Communication Scheme to Estimate the Empirical Measure of A Quantity of Interest Via Multiple Access Fading Channels,” in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 521–525, 2021.
- [111] V. Doshi, D. Shah, M. Medard, and M. Effro, “Functional Compression Through Graph Coloring,” *IEEE*, 8 2010.
- [112] J. T. Rickard, J. Aisbett, and G. Gibbon, “Knowledge Representation and Reasoning in Conceptual Spaces,” in *2007 IEEE Symposium on Foundations of Computational Intelligence*, pp. 583–590, 4 2007.
- [113] J. T. Rickard, J. Aisbett, and G. Gibbon, “Type-2 fuzzy conceptual spaces,” in *International Conference on Fuzzy Systems*, pp. 1–8, 7 2010. ISSN: 1098-7584.
- [114] R. Cubek, W. Ertel, and G. Palm, “High-level learning from demonstration with conceptual spaces and subspace clustering,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2592–2597, 5 2015. ISSN: 1050-4729.
- [115] J. R. Chapman, D. Kasmier, D. Limbaugh, S. R. Gagnon, J. L. Crassidis, J. Llinas, B. Smith, and A. P. Cox, “Conceptual Space Modeling for Space Event Characterization,” in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–8, 7 2020.
- [116] X. Li, T. Wei, Y. P. Chen, Y.-W. Tai, and C.-K. Tang, “FSS-1000: A 1000-Class Dataset for Few-Shot Segmentation,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2866–2875, 2020.

- [117] C. K. Thomas, E. C. Strinati, and W. Saad, “Reasoning with the Theory of Mind for Pragmatic Semantic Communication,” in *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*, pp. 662–668, 2024.
- [118] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. V. Poor, “Less Data, More Knowledge: Building Next Generation Semantic Communication Networks,” *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.
- [119] G. Bansal, K. Rajgopal, V. Chamola, Z. Xiong, and D. Niyato, “Healthcare in Metaverse: A Survey on Current Metaverse Applications in Healthcare,” *IEEE Access*, vol. 10, pp. 119914–119946, 2022.
- [120] S. Guo, Y. Wang, and P. Zhang, “Signal Shaping for Semantic Communication Systems with A Few Message Candidates,” in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pp. 1–5, 2022.
- [121] D. Wheeler, E. E. Tripp, and B. Natarajan, “Semantic Communication With Conceptual Spaces,” *IEEE Communications Letters*, vol. 27, no. 2, pp. 532–535, 2023.
- [122] Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A Review and New Perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [123] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, “A Survey on Metaverse: The State-of-the-art, Technologies, Applications, and Challenges,” *CoRR*, vol. abs/2111.09673, 2021.
- [124] J. Sun, W. Gan, H.-C. Chao, and P. S. Yu, “Metaverse: Survey, Applications, Security, and Opportunities,” 2022.
- [125] N. Kshetri, “The Economics of the Industrial Metaverse,” *IT Professional*, vol. 25, no. 1, pp. 84–88, 2023.

- [126] G. Albakri, R. Bouaziz, W. Alharthi, S. Kammoun, M. Al-Sarem, F. Saeed, and M. Hadwan, “Phobia Exposure Therapy Using Virtual and Augmented Reality: A Systematic Review,” *Applied Sciences*, vol. 12, no. 3, 2022.
- [127] W. W. Eaton, O. J. Bienvenu, and B. Miloyan, “Specific Phobias,” *The Lancet. Psychiatry*, vol. 5, pp. 678–686, 8 2018.
- [128] O. Bălan, G. Moise, A. Moldoveanu, M. Leordeanu, and F. Moldoveanu, “Fear Level Classification Based on Emotional Dimensions and Machine Learning Techniques,” *Sensors*, vol. 19, no. 7, 2019.
- [129] D. Kollias, “ABAW: Learning from Synthetic Data & Multi-Task Learning Challenges,” in *European Conference on Computer Vision, 2022*, 2022.
- [130] D. Kollias, P. Tzirakis, M. A. Nicolaou, A. Papaioannou, G. Zhao, B. Schuller, I. Kotisia, and S. Zafeiriou, “Deep Affect Prediction in-the-Wild: Aff-Wild Database and Challenge, Deep Architectures, and Beyond,” *International Journal of Computer Vision*, vol. 127, pp. 907–929, 6 2019.
- [131] 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.
- [132] S. Handrich, L. Dinges, F. Saxen, A. Al-Hamadi, and S. Wachmuth, “Simultaneous Prediction of Valence / Arousal and Emotion Categories in Real-Time,” in *2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 176–180, 2019.
- [133] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” *CoRR*, vol. abs/1612.08242, 2016.
- [134] IEEE, “IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific

- Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, 2021.
- [135] D. Wheeler and B. Natarajan, “Knowledge-Driven Semantic Communication Enabled by the Geometry of Meaning,” 2023. arXiv:2306.02917 [eess].
 - [136] L. Deng, “The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web],” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
 - [137] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep Learning Face Attributes in the Wild,” in *Proceedings of International Conference on Computer Vision (ICCV)*, 12 2015.
 - [138] D. Wheeler and B. Natarajan, “Autoencoder-Based Domain Learning for Semantic Communication with Conceptual Spaces,” in *2024 Wireless Telecommunications Symposium (WTS)*, pp. 1–6, 2024.
 - [139] D. P. Kingma and M. Welling, “An Introduction to Variational Autoencoders,” *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.
 - [140] C. K. Thomas, C. Chaccour, W. Saad, M. Debbah, and C. S. Hong, “Causal Reasoning: Charting a Revolutionary Course for Next-Generation AI-Native Wireless Networks,” *IEEE Vehicular Technology Magazine*, vol. 19, no. 1, pp. 16–31, 2024.
 - [141] H. Seo, J. Park, M. Bennis, and M. Debbah, “Semantics-Native Communication via Contextual Reasoning,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 3, pp. 604–617, 2023.
 - [142] F. Zhao, Y. Sun, L. Feng, L. Zhang, and D. Zhao, “Enhancing Reasoning Ability in Semantic Communication through Generative AI-Assisted Knowledge Construction,” *IEEE Communications Letters*, pp. 1–1, 2024. Conference Name: IEEE Communications Letters.

- [143] Y. Xiao, Y. Liao, Y. Li, G. Shi, H. V. Poor, W. Saad, M. Debbah, and M. Bennis, “Reasoning Over the Air: A Reasoning- Based Implicit Semantic-Aware Communication Framework,” *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 3839–3855, 2024.
- [144] C. Xing, J. Lv, T. Luo, and Z. Zhang, “Representation and Fusion Based on Knowledge Graph in Multi-Modal Semantic Communication,” *IEEE Wireless Communications Letters*, pp. 1–1, 2024. Conference Name: IEEE Wireless Communications Letters.
- [145] C. Kurisummoottil Thomas, W. Saad, and Y. Xiao, “Causal Semantic Communication for Digital Twins: A Generalizable Imitation Learning Approach,” *IEEE Journal on Selected Areas in Information Theory*, vol. 4, pp. 698–717, 2023.
- [146] C. Kurisummoottil Thomas and W. Saad, “Neuro-Symbolic Causal Reasoning Meets Signaling Game for Emergent Semantic Communications,” *IEEE Transactions on Wireless Communications*, vol. 23, no. 5, pp. 4546–4563, 2024.
- [147] G. Peyré and M. Cuturi, “Computational Optimal Transport: With Applications to Data Science,” *Foundations and Trends in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.
- [148] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio, “Learning with a Wasserstein Loss,” in *Advances in Neural Information Processing Systems* (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), Curran Associates, Inc., 2015.
- [149] J. Shen, Y. Qu, W. Zhang, and Y. Yu, “Wasserstein Distance Guided Representation Learning for Domain Adaptation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, April 2018. Number: 1.
- [150] J. Pearl, *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd ed., 2009.

- [151] A. Molak, *Causal Inference and Discovery in Python: Unlock the secrets of modern causal machine learning with DoWhy, EconML, PyTorch and more*. Birmingham: Packt Publishing, 1. ed., 2023. <https://amzn.to/3RebWzn>.
- [152] D. Zheng, Y. Zou, X. Zhang, and C. Bao, “SeNM-VAE: Semi-Supervised Noise Modeling with Hierarchical Variational Autoencoder,” March 2024. arXiv:2403.17502 [cs].
- [153] H. Janati, B. Muzellec, G. Peyré, and M. Cuturi, “Entropic Optimal Transport between Unbalanced Gaussian Measures has a Closed Form,” December 2020. arXiv:2006.02572 [math, stat].
- [154] Oxford English Dictionary, “reason, v.,” March 2024.
- [155] A. Zanga and F. Stella, “A Survey on Causal Discovery: Theory and Practice,” May 2023. arXiv:2305.10032 [cs].
- [156] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, vol. 81 of *Lecture Notes in Statistics*. New York, NY: Springer New York, 1993.
- [157] M. Chickering, “Statistically Efficient Greedy Equivalence Search,” in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 241–249, PMLR, August 2020. ISSN: 2640-3498.
- [158] S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvarinen, Y. Kawahara, T. Washio, P. O. Hoyer, and K. Bollen, “DirectLiNGAM: A Direct Method for Learning a Linear Non-Gaussian Structural Equation Model,” *Journal of Machine Learning Research*, 2011.
- [159] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, “DAGs with NO TEARS: Continuous Optimization for Structure Learning,” in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018.
- [160] J. Pearl, M. Glymour, and N. P. Jewell, *Causal inference in statistics: a primer*. Chichester, West Sussex: Wiley, 2016.

- [161] R. J. Bowden and D. A. Turkington, “Instrumental variables,” *Cambridge University Press*, 1984.
- [162] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins, “Double/debiased machine learning for treatment and structural parameters,” *The Econometrics Journal*, vol. 21, pp. C1–C68, February 2018.
- [163] A. Sharma, V. Syrgkanis, C. Zhang, and E. Kıcıman, “DoWhy: Addressing Challenges in Expressing and Validating Causal Assumptions,” August 2021. arXiv:2108.13518 [cs].
- [164] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “The German Traffic Sign Recognition Benchmark: A multi-class classification competition,” in *The 2011 International Joint Conference on Neural Networks*, pp. 1453–1460, 2011.
- [165] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019.
- [166] P. Blöbaum, P. Götz, K. Budhathoki, A. A. Mastakouri, and D. Janzing, “DoWhy-GCM: An extension of DoWhy for causal inference in graphical causal models,” *arXiv preprint arXiv:2206.06821*, 2022.
- [167] K. Battocchi, E. Dillon, M. Hei, G. Lewis, P. Oka, M. Oprescu, and V. Syrgkanis, “EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation.” <https://github.com/py-why/EconML>, 2019. Version 0.x.
- [168] G. Xin, P. Fan, and K. B. Letaief, “Semantic Communication: A Survey of Its Theoretical Development,” *Entropy*, vol. 26, Feb. 2024.
- [169] J. Liu, W. Zhang, and H. V. Poor, “A Rate-Distortion Framework for Characterizing Semantic Information,” in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2894–2899, July 2021.

- [170] P. A. Stavrou and M. Kountouris, “A Rate Distortion Approach to Goal-Oriented Communication,” in *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 590–595, June 2022.
- [171] P. A. Stavrou and M. Kountouris, “The Role of Fidelity in Goal-Oriented Semantic Communication: A Rate Distortion Approach,” *IEEE Transactions on Communications*, pp. 1–1, 2023.
- [172] W. Rudin, *Principles of Mathematical Analysis*. International series in pure and applied mathematics, New York: McGraw-Hill, 3. ed., [nachdr.] ed., 2008.

Appendix A

Code Repositories

This appendix contains information regarding code repositories that were developed for experiments discussed throughout this dissertation.

A.1 CSLearn

A Python package was developed to carry out the AE-based domain learning framework described in Chapters 5 and 6. This package is called CSLearn, and can be found in this GitHub repository: <https://github.com/dw-610/CSLearn>. Key features of the CSLearn package include:

- A simple high-level interface for training domain learning models on image data, as well as basic classifier and AE models for comparison.
- Various evaluation routines to examine the result of the domain learning algorithm.
- Options for training with the advanced VAE architecture with Wasserstein loss.
- Thorough in-code documentation for all modules in the CSLearn package.

A.2 Chapter 6 Experiment Code

The experiments in Chapter 6 testing the advanced domain learning framework and causal reasoning mechanism were conducted using Python code, including the CSLearn package mentioned above. All code as well as step-by-step instructions for reproducing the experiments can be found in this GitHub repository: <https://github.com/dw-610/concept-learning-and-causal-reasoning>. The repository includes scripts which do the following:

- Download and prepare the raw data used in the experiments.
- Implement the CSLearn package to learn the domains for the acquired data.
- Create the datasets and carry out the methods for causal learning.
- Train the semantic decoder and baseline models.
- Perform wireless simulations comparing SCCS-R to the baseline systems.

A.3 Chapter 7 Experiment Code

The numerical simulations detailed in Chapter 7 were also implemented in Python. The code can be found in the following GitHub repository: <https://github.com/dw-610/semantic-rate-dist-code>. The simulations are performed by running the script `simulate.py`. This script does the following:

- Creates the encoder/decoder functions for the naive, oracle, and minimum-discrepancy codes.
- Samples input symbols from the source alphabet and repeats a specified number of Monte Carlo trials for each coding scheme.
- Reports approximate expected values for the functional distortion, semantic distortion, and distortion discrepancy for the three schemes.