# Chapter 1
# Representation Learning and NLP


**Abstract** Natural languages are typical unstructured information. Conventional Natural Language Processing (NLP) heavily relies on feature engineering, which requires careful design and considerable expertise. Representation learning aims to learn representations of raw data as useful information for further classification or prediction. This chapter presents a brief introduction to representation learning, including its motivation and basic idea, and also reviews its history and recent advances in both machine learning and NLP.

## 1.1 Motivation

Machine learning addresses the problem of automatically learning computer programs from data. A typical machine learning system consists of three components [5]:

$$\text{Machine Learning} = \text{Representation} + \text{Objective} + \text{Optimization}. \tag{1.1}$$

That is, to build an effective machine learning system, we first transform useful information on raw data into internal representations such as feature vectors. Then by designing appropriate objective functions, we can employ optimization algorithms to find the optimal parameter settings for the system.

Data representation determines how much useful information can be extracted from raw data for further classification or prediction. If there is more useful information transformed from raw data to feature representations, the performance of classification or prediction will tend to be better. Hence, data representation is a crucial component to support effective machine learning.

Conventional machine learning systems adopt careful feature engineering as preprocessing to build feature representations from raw data. Feature engineering needs careful design and considerable expertise, and a specific task usually requires customized feature engineering algorithms, which makes feature engineering labor intensive, time consuming, and inflexible.

Representation learning aims to learn informative representations of objects from raw data automatically. The learned representations can be further fed as input to



Z. Liu et al., *Representation Learning for Natural Language Processing*,
https://doi.org/10.1007/978-981-15-5573-2_1

machine learning systems for prediction or classification. In this way, machine learning algorithms will be more flexible and desirable while handling large-scale and noisy unstructured data, such as speech, images, videos, time series, and texts.

Deep learning [9] is a typical approach for representation learning, which has recently achieved great success in speech recognition, computer vision, and natural language processing. Deep learning has two distinguishing features:

- **Distributed Representation**. Deep learning algorithms typically represent each object with a low-dimensional real-valued dense vector, which is named as *distributed representation*. As compared to one-hot representation in conventional representation schemes (such as bag-of-words models), distributed representation is able to represent data in a more compact and smoothing way, as shown in Fig. 1.1, and hence is more robust to address the sparsity issue in large-scale data.
- **Deep Architecture**. Deep learning algorithms usually learn a *hierarchical deep architecture* to represent objects, known as multilayer neural networks. The deep architecture is able to extract abstractive features of objects from raw data, which is regarded as an important reason for the great success of deep learning for speech recognition and computer vision.

Currently, the improvements caused by deep learning for NLP may still not be so significant as compared to speech and vision. However, deep learning for NLP has been able to significantly reduce the work of feature engineering in NLP in the meantime of performance improvement. Hence, many researchers are devoting to developing efficient algorithms on representation learning (especially deep learning) for NLP.

In this chapter, we will first discuss why representation learning is important for NLP and introduce the basic ideas of representation learning. Afterward, we will

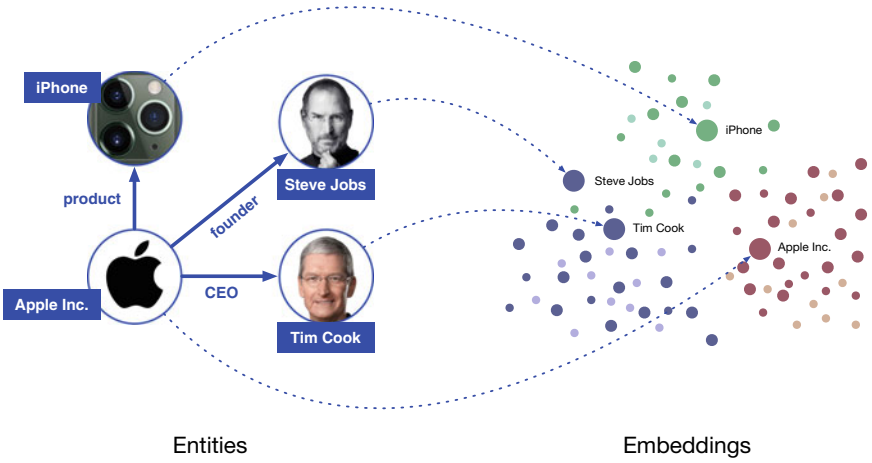


**Fig. 1.1** Distributed representation of words and entities in human languages

briefly review the development history of representation learning for NLP, introduce typical approaches of contemporary representation learning, and summarize existing and potential applications of representation learning. Finally, we will introduce the general organization of this book.

## 1.2 Why Representation Learning Is Important for NLP

NLP aims to build linguistic-specific programs for machines to understand languages. Natural language texts are typical unstructured data, with multiple granularities, multiple tasks, and multiple domains, which make NLP challenging to achieve satisfactory performance.

**Multiple Granularities**. NLP concerns about multiple levels of language entries, including but not limited to characters, words, phrases, sentences, paragraphs, and documents. Representation learning can help to represent the semantics of these language entries in a unified semantic space, and build complex semantic relations among these language entries.

**Multiple Tasks**. There are various NLP tasks based on the same input. For example, given a sentence, we can perform multiple tasks such as word segmentation, part-of-speech tagging, named entity recognition, relation extraction, and machine translation. In this case, it will be more efficient and robust to build a unified representation space of inputs for multiple tasks.

**Multiple Domains**. Natural language texts may be generated from multiple domains, including but not limited to news articles, scientific articles, literary works, and online user-generated content such as product reviews. Moreover, we can also regard texts in different languages as multiple domains. Conventional NLP systems have to design specific feature extraction algorithms for each domain according to its characteristics. In contrast, representation learning enables us to build representations automatically from large-scale domain data.

In summary, as shown in Fig. 1.2, representation learning can facilitate knowledge transfer across multiple language entries, multiple NLP tasks, and multiple application domains, and significantly improve the effectiveness and robustness of NLP performance.

## 1.3 Basic Ideas of Representation Learning

In this book, we focus on the distributed representation scheme (i.e., embedding), and talk about recent advances of representation learning methods for multiple language entries, including words, phrases, sentences, and documents, and their closely related objects including sememe-based linguistic knowledge, entity-based world knowledge, networks, and cross-modal entries.

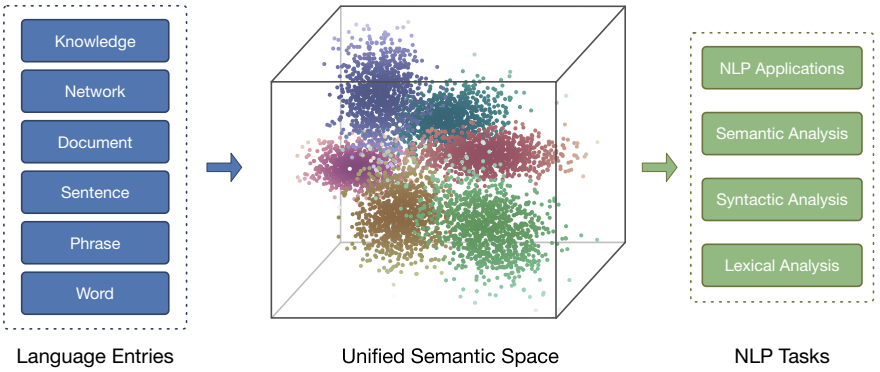


**Fig. 1.2** Distributed representation can provide unified semantic space for multi-grained language entries and for multiple NLP tasks

By distributed representation learning, all objects that we are interested in are projected into a unified low-dimensional semantic space. As demonstrated in Fig. 1.1, the geometric distance between two objects in the semantic space indicates their semantic relatedness; the semantic meaning of an object is related to which objects are close to it. In other words, it is the relative closeness with other objects that reveals an object's meaning rather than the absolute position.

## 1.4 Development of Representation Learning for NLP

In this section, we introduce the development of representation learning for NLP, also shown in Fig. 1.3. To study representation schemes in NLP, words would be a good start, since they are the minimum units in natural languages. The easiest way to represent a word in a computer-readable way (e.g., using a vector) is **one-hot vector**, which has the dimension of the vocabulary size and assigns 1 to the word's corresponding position and 0 to others. It is apparent that one-hot vectors hardly contain any semantic information about words except simply distinguishing them from each other.

One of the earliest ideas of word representation learning can date back to ***n*-gram models** [15]. It is easy to understand: when we want to predict the next word in a sequence, we usually look at some previous words (and in the case of $n$-gram, they are the previous $n-1$ words). And if going through a large-scale corpus, we can count and get a good probability estimation of each word under the condition of all combinations of $n-1$ previous words. These probabilities are useful for predicting words in sequences, and also form vector representations for words since they reflect the meanings of words.

The idea of $n$-gram models is coherent with the **distributional hypothesis**: linguistic items with similar distributions have similar meanings [7]. In another phrase, "a word is characterized by the company it keeps" [6]. It became the fundamental idea of many NLP models, from word2vec to BERT.

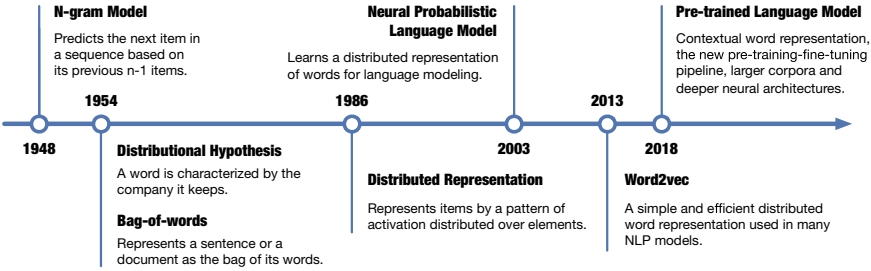


**Fig. 1.3** The timeline for the development of representation learning in NLP. With the growing computing power and large-scale text data, distributed representation trained with neural networks and large corpora has become the mainstream

Another example of the distributional hypothesis is **Bag-Of-Words (BOW) models** [7]. BOW models regard a document as a bag of its words, disregarding the orders of these words in the document. In this way, the document can be represented as a vocabulary-size vector, in which each word that has appeared in the document corresponds to a unique and nonzero dimension. Then a score can be further computed for each word (e.g., the numbers of occurrences) to indicate the weights of these words in the document. Though very simple, BOW models work great in applications like spam filtering, text classification, and information retrieval, proving that the distributions of words can serve as a good representation for text.

In the above cases, each value in the representation clearly matches one entry (e.g., word scores in BOW models). This one-to-one correspondence between concepts and representation elements is called **local representation** or **symbol-based representation**, which is natural and simple.

In **distributed representation**, on the other hand, each entity (or attribute) is represented by a pattern of activation distributed over multiple elements, and each computing element is involved in representing multiple entities [11]. Distributed representation has been proved to be more efficient because it usually has low dimensions that can prevent the sparsity issue. Useful hidden properties can be learned from large-scale data and emerged in distributed representation. The idea of distributed representation was originally inspired by the neural computation scheme of humans and other animals. It comes from neural networks (activations of neurons), and with the great success of deep learning, distributed representation has become the most commonly used approach for representation learning.

One of the pioneer practices of distributed representation in NLP is **Neural Probabilistic Language Model (NPLM)** [1]. A language model is to predict the joint probability of sequences of words ($n$-gram models are simple language models). NPLM first assigns a distributed vector for each word, then uses a neural network to predict the next word. By going through the training corpora, NPLM successfully learns how to model the joint probability of sentences, while brings **word embeddings** (i.e., low-dimensional word vectors) as learned parameters in NPLM. Though

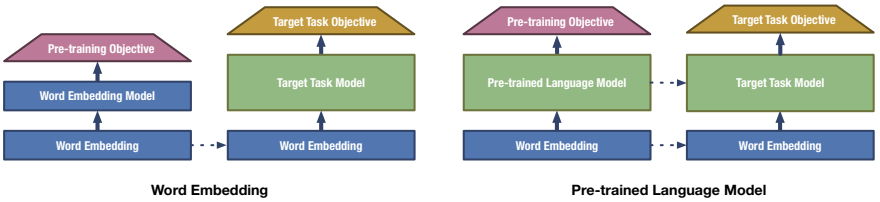


**Fig. 1.4** This figure shows how word embeddings and pre-trained language models work in NLP pipelines. They both learn distributed representations for language entries (e.g., words) through pretraining objectives and transfer them to target tasks. Furthermore, pre-trained language models can also transfer model parameters

it is hard to tell what each element of a word embedding actually means, the vectors indeed encode semantic meanings about the words, verified by the performance of NPLM.

Inspired by NPLM, there came many methods that embed words into distributed representations and use the language modeling objective to optimize them as model parameters. Famous examples include **word2vec** [12], **GloVe** [13], and **fastText** [3]. Though differing in detail, these methods are all very efficient to train, utilize large-scale corpora, and have been widely adopted as word embeddings in many NLP models. Word embeddings in the NLP pipeline map discrete words into informative low-dimensional vectors, and help to shine a light on neural networks in computing and understanding languages. It makes representation learning a critical part of natural language processing.

The research on representation learning in NLP took a big leap when **ELMo** [14] and **BERT** [4] came out. Besides using larger corpora, more parameters, and more computing resources as compared to word2vec, they also take complicated context in text into consideration. It means that instead of assigning each word with a fixed vector, ELMo and BERT use multilayer neural networks to calculate dynamic representations for the words based on their context, which is especially useful for the words with multiple meanings. Moreover, BERT starts a new fashion (though not originated from it) of the pretrained fine-tuning pipeline. Previously, word embeddings are simply adopted as input representation. But after BERT, it becomes a common practice to keep using the same neural network structure such as BERT in both pretraining and fine-tuning, which is taking the parameters of BERT for initialization and fine-tuning the model on downstream tasks (Fig. 1.4).

Though not a big theoretical breakthrough, BERT-like models (also known as **Pre-trained Language Models (PLM)**, for they are pretrained through language modeling objective on large corpora) have attracted wide attention in the NLP and machine learning community, for they have been so successful and achieved state-of-the-art on almost every NLP benchmarks. These models show what large-scale data and computing power can lead to, and new research works on the topic of Pre-Trained language Models (PLMs) emerge rapidly. Probing experiments demonstrate that PLMs implicitly encode a variety of linguistic knowledge and patterns inside

their multilayer network parameters [8, 10]. All these significant performances and interesting analyses suggest that there are still a lot of open problems to explore in PLMs, as the future of representation learning for NLP.

Based on the distributional hypothesis, representation learning for NLP has evolved from symbol-based representation to distributed representation. Starting from word2vec, word embeddings trained from large corpora have shown significant power in most NLP tasks. Recently, emerged PLMs (like BERT) take complicated context into word representation and start a new trend of the pretraining fine-tuning pipeline, bringing NLP to a new level. What will be the next big change in representation learning for NLP? We hope the contents of this book can give you some inspiration.

## 1.5 Learning Approaches to Representation Learning for NLP

People have developed various effective and efficient approaches to learn semantic representations for NLP. Here we list some typical approaches.

**Statistical Features**: As introduced before, semantic representations for NLP in the early stage often come from statistics, instead of emerging from the optimization process. For example, in $n$-gram or bag-of-words models, elements in the representation are usually frequencies or numbers of occurrences of the corresponding entries counted in large-scale corpora.

**Hand-craft Features**: In certain NLP tasks, syntactic and semantic features are useful for solving the problem. For example, types of words and entities, semantic roles and parse trees, etc. These linguistic features may be provided with the tasks or can be extracted by specific NLP systems. In a long period before the wide use of distributed representation, researchers used to devote lots of effort into designing useful features and combining them as the inputs for NLP models.

**Supervised Learning**: Distributed representations emerge from the optimization process of neural networks under supervised learning. In the hidden layers of neural networks, the different activation patterns of neurons represent different entities or attributes. With a training objective (usually a loss function for the target task) and supervised signals (usually the gold-standard labels for training instances of the target tasks), the networks can learn better parameters via optimization (e.g., gradient descent). With proper training, the hidden states will become informative and generalized as good semantic representations of natural languages.

For example, to train a neural network for a sentiment classification task, the loss function is usually set as the cross-entropy of the model predictions with respect to the gold-standard sentiment labels as supervision. While optimizing the objective, the loss gets smaller, and the model performance gets better. In the meantime, the hidden states of the model gradually form good sentence representations by encoding the necessary information for sentiment classification inside the continuous hidden space.

**Self-supervised Learning**: In some cases, we simply want to get good representations for certain elements, so that these representations can be transferred to other tasks. For example, in most neural NLP models, words in sentences are first mapped to their corresponding word embeddings (maybe from word2vec or GloVe) before sent to the networks. However, there are no human-annotated "labels" for learning word embeddings. To acquire the training objective necessary for neural networks, we need to generate "labels" intrinsically from existing data. This is called self-supervised learning (one way for unsupervised learning).

For example, language modeling is a typical "self-supervised" objective, for it does not require any human annotations. Based on the distributional hypothesis, using the language modeling objective can lead to hidden representations that encode the semantics of words. You may have heard of a famous equation: $\mathbf{w}(\texttt{king}) - \mathbf{w}(\texttt{man}) + \mathbf{w}(\texttt{woman}) = \mathbf{w}(\texttt{queen})$, which demonstrates the analogical properties that the word embeddings have possessed through self-supervised learning.

We can see another angle of self-supervised learning in autoencoders. It is also a way to learn representations for a set of data. Typical autoencoders have a reduction (encoding) phase and a reconstruction (decoding) phase. In the reduction phase, an item from the data is encoded into a low-dimensional representation, and in the reconstruction phase, the model tries to reconstruct the item from the intermediate representation. Here, the training objective is the reconstruction loss, derived from the data itself. During the training process, meaningful information is encoded and kept in the latent representation, while noise signals are discarded.

Self-supervised learning has made a great success in NLP, for the plain text itself contains abundant knowledge and patterns about languages, and self-supervised learning can fully utilize the existing large-scale corpora. Nowadays, it is still the most exciting research area of representation learning for natural languages, and researchers continue to put their efforts into this direction.

Besides, many other machine learning approaches have also been explored in representation learning for NLP, such as adversarial training, contrastive learning, few-shot learning, meta-learning, continual learning, reinforcement learning, et al. How to develop more effective and efficient approaches of representation learning for NLP and to better take advantage of large-scale and complicated corpora and computing power, is still an important research topic.

## 1.6 Applications of Representation Learning for NLP

In general, there are two kinds of applications of representation learning for NLP. In one case, the semantic representation is trained in a pretraining task (or designed by human experts) and is transferred to the model for the target task. Word embedding is an example of the application. It is trained by using language modeling objective and is taken as inputs for other down-stream NLP models. In this book, we will

also introduce sememe knowledge representation and world knowledge representation, which can also be integrated into some NLP systems as additional knowledge augmentation to enhance their performance in certain aspects.

In other cases, the semantic representation lies within the hidden states of the neural model and directly aims for better performance of target tasks as an end-to-end fashion. For example, many NLP tasks want to semantically compose sentence or document representation: tasks like sentiment classification, natural language inference, and relation extraction require sentence representation and the tasks like question answering need document representation. As shown in the latter part of the book, many representation learning methods have been developed for sentences and documents and benefit these NLP tasks.

## 1.7 The Organization of This Book

We start the book from word representation. By giving a thorough introduction to word representation, we hope the readers can grasp the basic ideas for representation learning for NLP. Based on that, we further talk about how to compositionally acquire the representation for higher level language components, from sentences to documents.

As shown in Fig. 1.5, representation learning will be able to incorporate various types of structural knowledge to support a deep understanding of natural languages, named as knowledge-guided NLP. Hence, we next introduce two forms of knowledge representation that are closely related to NLP. On the one hand, sememe representation tries to encode linguistic and commonsense knowledge in natural languages. Sememe is defined as the minimum indivisible unit of semantic meaning [2]. With the help of sememe representation learning, we can get more interpretable and more robust NLP models. On the other hand, world knowledge representation studies how to encode world facts into continuous semantic space. It can not only help with knowledge graph tasks but also benefit knowledge-guided NLP applications.

Besides, the network is also a natural way to represent objects and their relationships. In the network representation section, we study how to embed vertices and edges in a network and how these elements interact with each other. Through the applications, we further show how network representations can help NLP tasks.

Another interesting topic related to NLP is the cross-modal representation, which studies how to model unified semantic representations across different modalities (e.g., text, audios, images, videos, etc.). Through this section, we review several cross-modal problems along with representative models.

At the end of the book, we introduce some useful resources to the readers, including deep learning frameworks and open-source codes. We also share some views about the next big topics in representation learning for NLP. We hope that the resources and the outlook can help our readers have a better understanding of the content of the book, and inspire our readers about how representation learning in NLP would further develop.

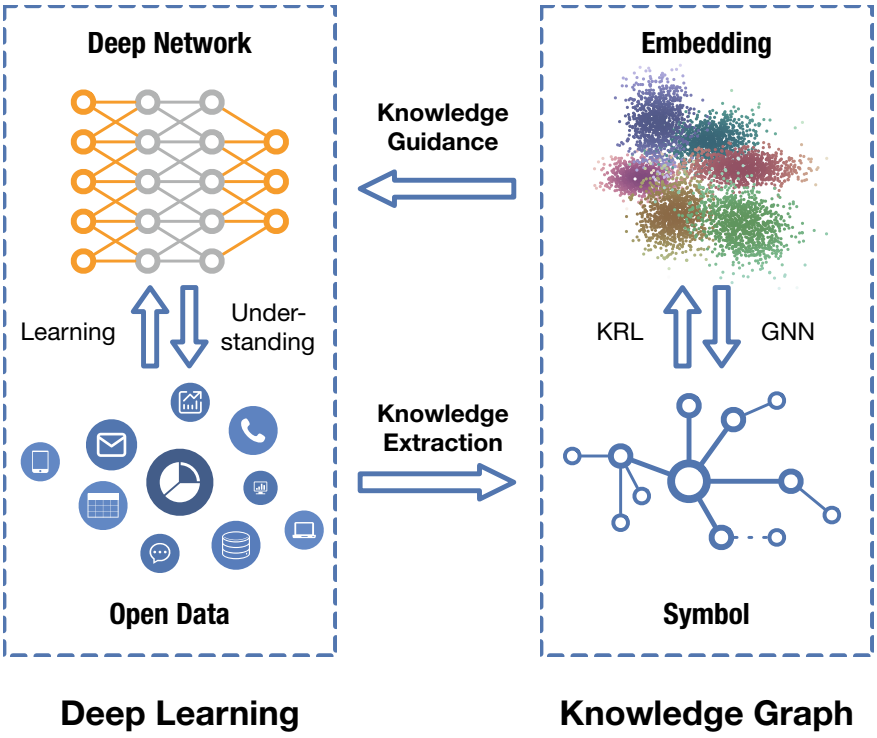


**Fig. 1.5** The architecture of knowledge-guided NLP

## References

1. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.
2. Leonard Bloomfield. A set of postulates for the science of language. *Language*, 2(3):153–164, 1926.
3. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, 2019.
5. Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.
6. John R Firth. A synopsis of linguistic theory, 1930–1955. 1957.
7. Zellig S Harris. Distributional structure. *Word*, 10(2–3):146–162, 1954.
8. John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of NAACL-HLT*, 2019.
9. Goodfellow Ian, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.
10. Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of NAACL-HLT*, 2019.

11. James L McClelland, David E Rumelhart, PDP Research Group, et al. Parallel distributed processing. *Explorations in the Microstructure of Cognition*, 2:216–271, 1986.
12. T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Proceedings of NeurIPS*, 2013.
13. Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, 2014.
14. Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.
15. Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.