HOW TO EXTRACT MEANING FROM A PHRASE

Zhang Caigui, Yu.German
Department of Information Technologies in Automated Systems,
Belarussian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
E-mail: zhangcaigui309@gmail.com, jgerman@bsuir.by

This paper describes how the word vector model and the BERT model and its derivatives can be used to extract meaning from a phrase.

INTRODUCTION

In the field of Natural Language Processing (NLP), extracting meaning from phrases is a fundamental challenge that underpins a wide range of applications such as sentiment analysis, information retrieval and question answering systems.

Traditionally, meaning extraction has relied on lexical and syntactic approaches that focus on the components of language - words and their grammatical relationships. However, these approaches often fall short when dealing with idiomatic expressions, contextually relevant phrases, or polysemous words (which have multiple meanings depending on their usage). To address these limitations, modern techniques utilize advanced machine learning models, especially those based on deep learning architectures.

This report focuses on the application of traditional word vector models Word2Vec and GloVe as well as the BERT model and its derivatives of modern contextual embedding techniques in extracting the meaning of phrases.

I. Word2Vec MODEL

Word2Vec is an algorithm for converting words into vectors, widely used in Natural Language Processing (NLP) tasks, that employs deep learning techniques to efficiently capture the semantic relationships between words.

There are two main training models for Word2Vec: CBOW (Continuous Bag of Words) and Skip-gram, but they are trained differently.

The goal of CBOW model is to predict the target word from context words. Given a sentence, it tries to predict the center word using the surrounding words. For a given context:

$$C = \{w_{t-m},w_{t-m+1},\dots,w_{t+m}\} \tag{1}$$

the center word is w_t, the objective function of CBOW can be expressed as:

$$J = -\sum_{t=1}^{T} \log P(w_t \mid C) \tag{2}$$

where $P(w_t \mid C)$ is the probability of predicting the center word w_t given the context C.

The core idea of Skip-gram model is opposite to CBOW. Its goal is to predict the context word from the current center word. Skip-gram focuses on the relationship between the center word and its surrounding words. Given the center word w_t and the context words:

$$C = \{w_{t-m},w_{t-m+1},\dots,w_{t+m}\} \tag{3}$$

the objective function for Skip-gram can be expressed as:

$$J = -\sum_{t=1}^{T} \sum_{w_j \in C} \log P(w_j \mid w_t) \tag{4}$$

where $P(w_j \mid w_t)$ is the probability of predicting the context word w_j given the center word w_t.

II. GloVe MODEL

The core idea of GloVe is based on word co-occurrence statistics, using global statistical information to generate word vectors. GloVe constructs a co-occurrence matrix that records the co-occurrence frequency of each pair of words in the corpus.

Specific steps: first, count the number of times each pair of words co-occurs in a certain window, and build the co-occurrence matrix X, where X_{ij} is the co-occurrence frequency of word i and word j. Then, calculate the co-occurrence probability of each pair of words, defined as:

$$P_{ij} = \frac{X_{ij}}{\sum_k X_{ik}} \tag{5}$$

after that, the cost function is constructed to optimize the word vectors:

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(\mathbf{w}_i^T \mathbf{w}_j + b_i + b_j - \log(X_{ij})\right)^2 \tag{6}$$

where $f(X_{ij})$ is the weighting function and b_i and b_j are bias terms.

Finally, optimization algorithms such as gradient descent are used to minimize the cost function and learn the word vectors.

III. BERT MODELs

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model based on the Transformer architecture.

The core idea of BERT is to generate contextual embeddings for words through unsupervised learning on a large-scale text dataset. Its main training objectives are twofold:

1. **Masked Language Model (MLM)**: In the input sentence, some words are randomly masked, and the model's task is to predict these masked words. This allows the model to learn the relationships between words and their context.
2. **Next Sentence Prediction (NSP)**: Given two sentences, the model needs to determine whether the second sentence is the next sentence of the first one. This task helps the model understand the relationships between sentences.

The input to BERT consists of token embeddings, position embeddings, and segment embeddings for the input words.

BERT is composed of multiple stacked Transformer encoders. The core of each encoder is the self-attention mechanism, which is defined by the following formula:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (7)$$

where Q is the query vector. K is the key vector. V is the value vector. d_k is the dimension of the key vector.

In the task of extracting the meaning of short sentences, the BERT model can be applied through the following steps:

1. **Input Preparation**: Convert the short sentence into a format acceptable to BERT, adding special tokens (such as [CLS] and [SEP]).

2. **Feature Extraction**: Input the short sentence into the BERT model to obtain the contextual embeddings for each word.

3. **Sentence Representation**: Typically, the output corresponding to the [CLS] token is used as the representation for the entire sentence, to perform subsequent tasks (such as classification or regression).

4. **Downstream Tasks**: Utilize the obtained sentence representation for specific downstream tasks, such as sentence similarity calculation and sentiment analysis.

IV. DERIVATIVE MODELS OF BERT

The success of BERT has led to the emergence of several derivative models, such as:

1. **RoBERTa**: An improvement over BERT that utilizes larger training data, longer training times, and removes the NSP task.

2. **DistilBERT**: A model that compresses BERT using knowledge distillation techniques, reducing computational load while maintaining high performance.

3. **ALBERT**: A model that significantly reduces the number of parameters through parameter sharing and factorized embedding matrices.

V. SUMMARY

This report examines the applications of Word2Vec, GloVe, and BERT, along with its derivative models, in extracting the meaning of short sentences.

Word2Vec generates word embeddings by analyzing local context through methods like Continuous Bag of Words and Skip-Gram. GloVe focuses on global statistical information to create embeddings that capture word relationships from a co-occurrence matrix.

BERT takes a step further by using a transformer architecture and bidirectional context, trained on tasks like Masked Language Model and Next Sentence Prediction, which enhances its ability to understand nuanced meanings in sentences.

Derivative models such as RoBERTa, DistilBERT, and ALBERT improve upon BERT by optimizing training data, reducing model size, and sharing parameters, respectively, making them efficient for various NLP tasks.

Overall, these models collectively advance the capabilities of NLP in understanding and processing short sentences effectively.

1. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In Advances in Neural Information Processing Systems (pp. 3111-3119).
2. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532-1543).
3. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). 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 (pp. 4171-4186).
4. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.
5. Lan, Z., Chen, M., Goodman, S., Goot, J., & Wang, W. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In Proceedings of the 34th AAAI Conference on Artificial Intelligence (pp. 13633-13634).
6. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Ott, M., ... & Stoyanov, V. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Processing. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 7871-7880).