文本分析 (三)：大语言模型及其应用

汪小圈

2025-05-19

Word2Vec 的局限性

- **一词一向量问题**：每个词只对应一个固定的向量，无法处理一词多义
- **上下文无关**：词向量无法捕捉词语在特定上下文中的含义变化
- **长距离依赖问题**：无法捕捉句子中相距较远的词之间的依赖关系
- **表达能力有限**：固定维度的向量难以编码复杂的语言知识和语法结构

上下文感知的词表示

- **动态表示**：同一个词在不同上下文中具有不同的向量表示
- **语义消歧**：能够根据上下文区分多义词的不同含义
- **句法感知**：能够捕捉词语在句子中的句法功能
- **长距离依赖**：能够建模句子中远距离词语之间的关系

一个词的含义不仅取决于它自身，更取决于它的上下文环境

语言模型：理解上下文的基础

语言模型的基本任务：预测序列中的下一个词

$$P(w_t|w_1, w_2, ..., w_{t-1})$$

不同类型的语言模型：

- **传统 n-gram 语言模型**：$P(w_t|w_{t-2}, w_{t-1})$
- **循环神经网络 (RNN) 语言模型**：通过隐藏状态递归编码全部历史
- **双向语言模型**：同时考虑左侧和右侧上下文
- **Transformer 语言模型**：通过注意力机制直接建模所有位置间的依赖关系

上下文表示的早期尝试：ELMo (2018)

ELMo (Embeddings from Language Models):

- 使用双层双向 LSTM 架构
- 融合前向和后向语言模型
- 对不同层的表示进行加权组合
- 表示公式：$ELMo_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$

特点: - 动态词表示：同一词在不同上下文中有不同表示 - 上下文敏感：捕捉多义词在不同语境中的含义 - 层次化：不同层捕捉不同类型的语言信息

预训练语言模型的发展

GPT (2018)

- 单向 Transformer 结构
- 基于 Transformer 解码器
- 仅使用前向语言模型预训练
- 首创”预训练 + 微调”范式
- 通过预测下一个词学习语言表示

BERT (2018)

- 使用双向 Transformer 编码器
- 采用掩码语言模型 (Masked LM) 预训练
- 同时使用下一句预测 (NSP) 任务
- 能够同时考虑左右上下文信息
- 极大提升了各类 NLP 任务的性能上限

Transformer 架构：BERT 的基础

自注意力机制是 Transformer 的核心组件：

1. 将输入向量 X 转换为查询 (Q)、键 (K) 和值 (V)：

$$Q = XW^Q, K = XW^K, V = XW^V$$

2. 计算注意力得分并归一化：

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

优势：- 可以捕捉任意距离的依赖关系 - 计算复杂度低，允许并行计算

多头注意力与位置编码

多头注意力:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

其中，$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

位置编码:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

Transformer 编码器结构

一个完整的 Transformer 编码器层包含：

1. 多头自注意力机制
2. 层归一化（Layer Normalization）
3. 前馈神经网络（Feed-Forward Network）
4. 残差连接（Residual Connection）

组合方式：

$$\hat{h} = LayerNorm(x + MultiHeadAttention(x))$$

$$h = LayerNorm(\hat{h} + FFN(\hat{h}))$$

BERT 的输入表示

BERT 的输入由三种嵌入的总和组成：

- **词嵌入**：WordPiece 词表中的词元对应的嵌入
- **段嵌入**：区分句子对中的第一句和第二句
- **位置嵌入**：表示词元在序列中的位置

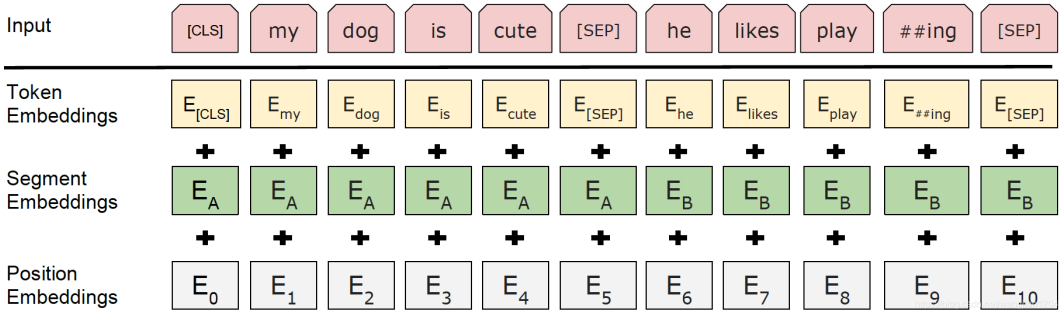

图 1: BERT 输入表示

BERT 的预训练任务

1. **掩码语言模型（Masked LM）**：
   - 随机掩盖输入中 15% 的词元
   - 80% 用 `[MASK]` 替换，10% 用随机词替换，10% 保持不变
   - 训练模型预测被掩盖的原始词元
2. **下一句预测（Next Sentence Prediction）**：
   - 预测第二句是否是第一句的真实后续
   - 训练数据中 50% 是真实连续句子，50% 是随机句子对

BERT 的模型变体

BERT-Base:

- 12 层 Transformer 编码器
- 12 个注意力头
- 768 维隐藏层
- 1.1 亿参数

BERT-Large:

- 24 层 Transformer 编码器
- 16 个注意力头
- 1024 维隐藏层
- 3.4 亿参数

BERT 的微调方式

- **序列级任务**（如分类）：使用 `[CLS]` 标记的最终隐藏状态
- **词元级任务**（如 NER）：使用每个词元的最终隐藏状态
- **句子对任务**（如问答）：同时输入问题和段落，识别答案跨度

微调过程通常只需要少量标注数据和训练轮次

BERT 的内部工作机制

层次化语言知识：

- **底层**（1-4 层）：表面语法特征、词性、局部依赖
- **中层**（5-8 层）：短语级语义和共指关系
- **高层**（9-12 层）：长距离依赖和抽象语义关系

注意力头的专业化：

- **语法头**：关注句法依赖关系
- **语义头**：关注语义相关的词
- **共指头**：关注指代同一实体的表达

BERT 的后续演进

预训练任务优化：

- **RoBERTa**：移除 NSP 任务，更大批量和更多数据
- **ALBERT**：参数共享和分解嵌入
- **ELECTRA**：判别式替换检测训练

知识增强：

- **KnowBERT**：集成知识库信息
- **ERNIE**：加入实体和短语级掩码
- **FinBERT**：针对金融领域的专业知识

模型架构改进：

- **SpanBERT**：掩盖连续文本片段
- **XLNet**：排列语言模型
- **DeBERTa**：解耦注意力机制

Transformer 架构的扩展

编码器-解码器结构：

- **编码器**：处理输入序列，生成上下文表示
- **解码器**：基于编码器输出生成目标序列
- **交叉注意力**：解码器通过注意力机制访问编码器输出
- 代表模型：T5, BART

仅解码器架构：

- **单向自注意力**：每个位置只能看到其前面的位置
- **自回归生成**：逐词生成输出序列
- 代表模型：GPT 系列, LLaMA

大型语言模型的关键创新

规模扩展：

- 从亿级参数到千亿参数
- 训练数据从 GB 级到 TB 级
- 计算资源大幅增长

涌现能力（Emergent Abilities）：

- **指令跟随**：理解并执行自然语言指令
- **思维链推理**：通过分步骤推理解决复杂问题
- **上下文学习**：从少量示例中学习新任务

提示工程与思维链推理

提示工程（Prompt Engineering）：

- 通过精心设计的提示引导模型行为
- 不同于传统的微调范式
- 灵活调整模型输出

思维链推理（Chain-of-Thought）：

- 让模型先生成推理过程，再给出结论
- 显著提高解决复杂问题的能力
- Prompt + 思考过程 $\rightarrow$ 更准确的结果

上下文学习（In-context Learning）：

- 在提示中包含示例，引导模型学习模式
- 无需参数更新，即可适应新任务

代表性大型语言模型

GPT 系列：

- **GPT-1** (2018)：1.17 亿参数
- **GPT-2** (2019)：15 亿参数
- **GPT-3** (2020)：1750 亿参数
- **GPT-4** (2023)：>1 万亿参数（估计）

开源大模型：

- **LLaMA 系列**：7B-65B 参数
- **ChatGLM**：双语模型
- **DeepSeek**：专注长序列处理
- **Qwen**：阿里云开发

多模态大语言模型

跨模态理解与生成：

- **CLIP**：连接图像和文本的表示学习，通过对比学习实现
- **GPT-4V**：分析图像内容并生成相关文本解释
- **Gemini**：处理文本、图像、音频和视频的多模态系统
- **Claude 3**：具有强大视觉理解和推理能力

应用场景：- 金融图表理解与分析 - 多媒体内容自动总结 - 文档理解（包含表格、图片和文本）- 跨模态检索（通过文本查找图像或反之）

大语言模型的金融应用

信息提取与分析：

- 报告解析（财报关键指标提取、研报摘要）
- 市场情感分析（新闻情绪、投资者情绪）
- 事件提取（财经新闻事件识别、因果关系分析）

金融文本生成：

- 研究报告生成（财务分析、行业趋势报告）
- 监管合规（合规声明、文档检查）
- 客户交互（智能金融顾问、投资建议）

大语言模型在金融分析中的最佳实践

提示设计技巧：

- 明确任务界定（指定分析目标和输出格式）
- 思维链设计（引导模型分步思考）
- 角色设定（指定专业角色，如”金融分析师”）

金融特定优化：

- 上下文补充（提供行业背景信息）
- 多模型比较（通用模型与金融专业模型对比）
- 人机协作（模型输出作为专业分析起点）

局限性与注意事项：

- 事实准确性（需要人工验证）
- 偏见风险（保持客观中立）
- 时效性限制（及时更新分析）

从静态向量到大语言模型的演进

- **表示方法演进**：从静态词向量到上下文感知的动态表示
- **架构演进**：从浅层神经网络到深层 Transformer 架构
- **规模演进**：从百万参数到千亿参数
- **应用演进**：从特征提取到端到端文本理解与生成

无监督学习的新范式

- **零样本学习**：无需额外标注数据，直接分类新数据
- **上下文学习**：通过提示中的示例引导模型学习模式
- **涌现能力**：模型规模增长带来质的飞跃
- **提示工程**：通过设计提示引导模型行为
- **检索增强生成 (RAG)**：结合外部知识库和搜索引擎，提高模型输出的准确性和时效性