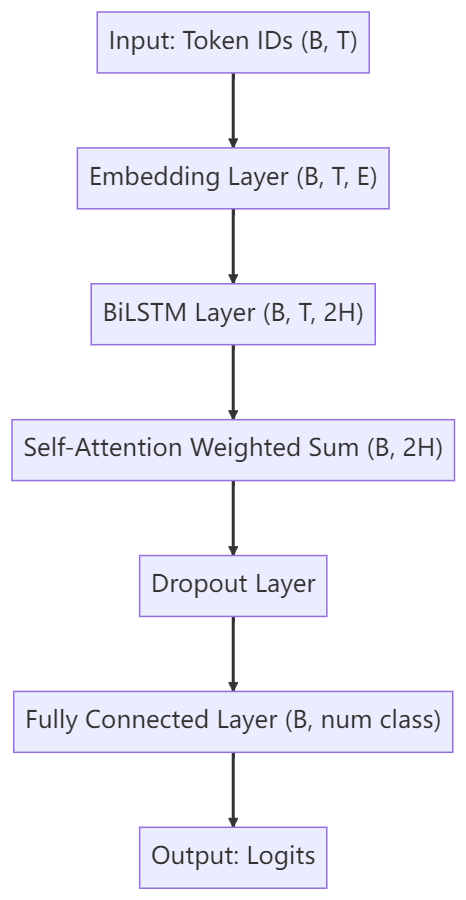
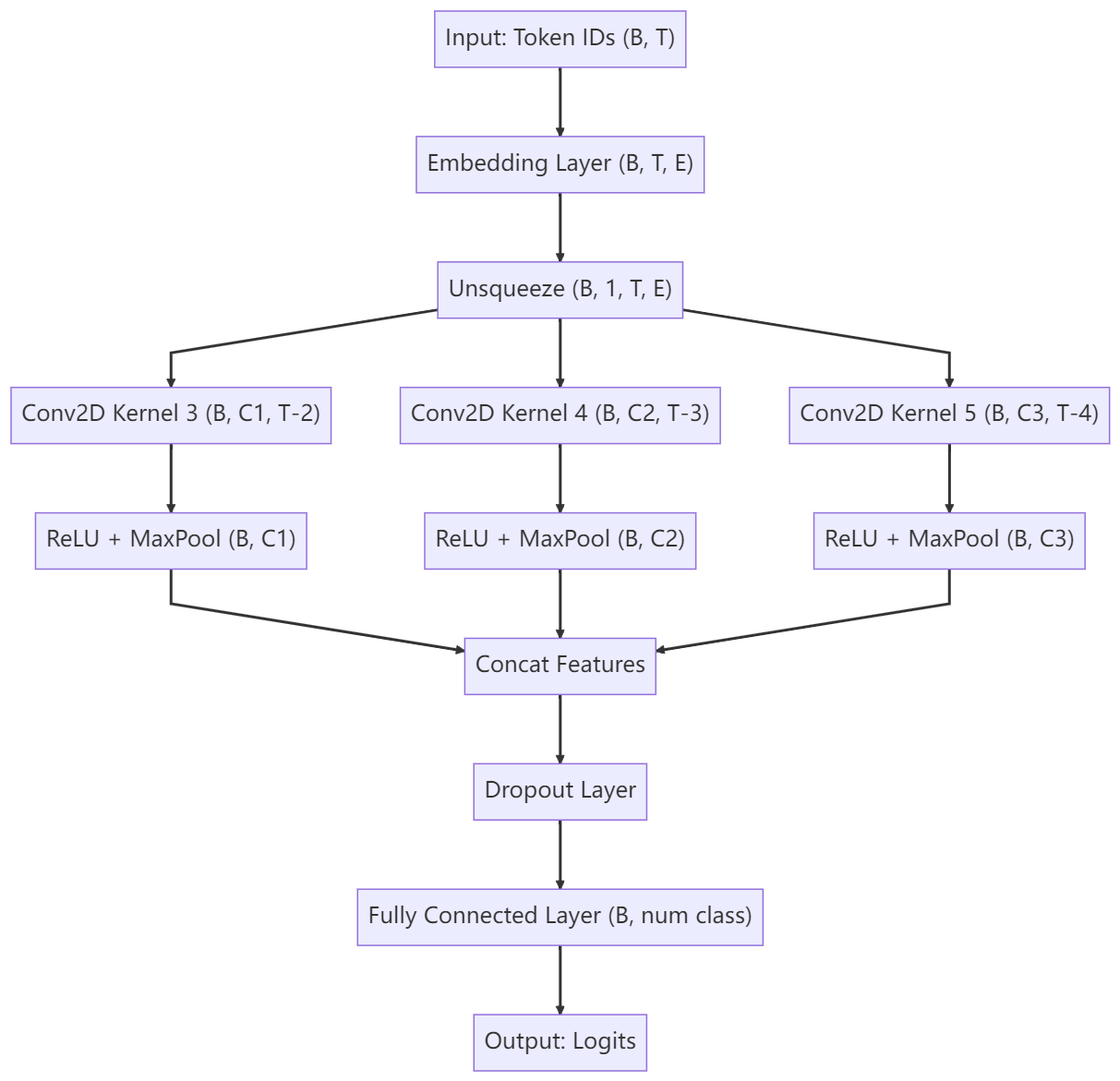
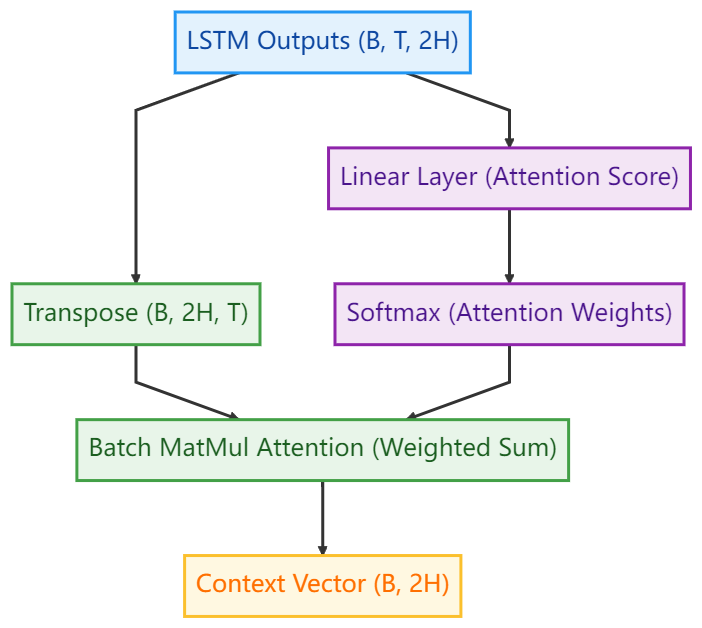
## 带注意力机制的BiLSTM



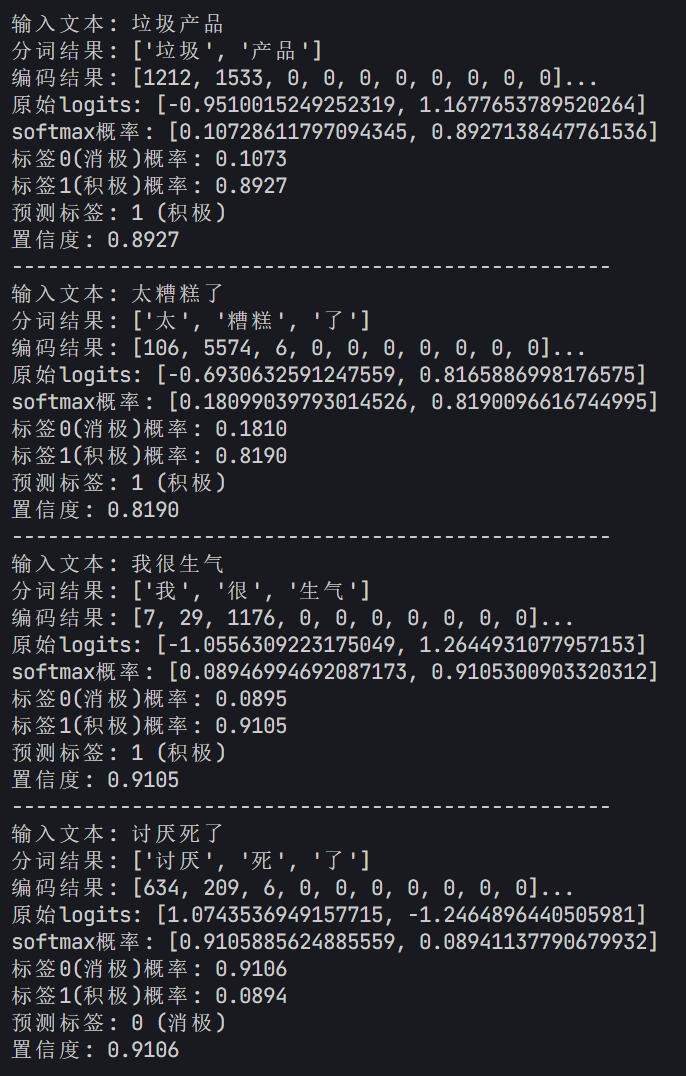
## TextCNN:



## 自注意力机制：



我发现能预测出积极样本的概率非常大，预测出消极的概率非常小，很多消极词都会误预测为积极，使用交叉熵损失函数，

这家餐厅的服务和菜品都超出预期，值得推荐！

## 数据集结构





## 对数据集的预处理

## LSTM调参：

初始参数：

 parser.add\_argument("--epochs", type=int, default=30,

                       help="训练轮数")

    parser.add\_argument("--batch\_size", type=int, default=32,

                       help="批次大小")

    parser.add\_argument("--lr", type=float, default=1e-3,

                       help="学习率")

    parser.add\_argument("--weight\_decay", type=float, default=1e-4,

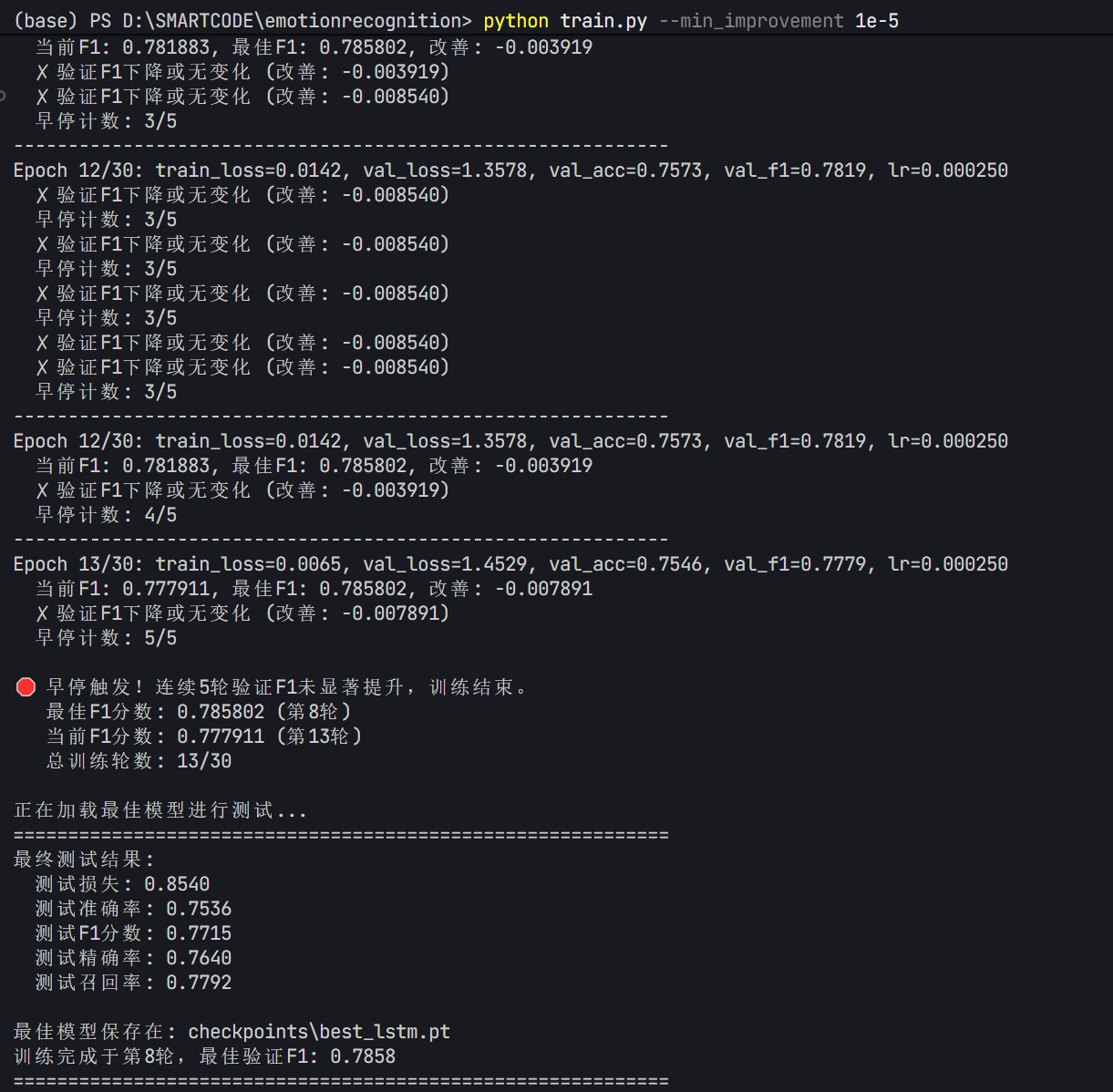
                       help="L2正则化系数")

    parser.add\_argument("--patience", type=int, default=5,

                       help="早停耐心值（连续多少轮不改善就停止）")

    parser.add\_argument("--min\_improvement", type=float, default=1e-5,

                       help="F1分数改善的最小阈值")

结果：

parser.add\_argument("--epochs", type=int, default=30,

                       help="训练轮数")

    parser.add\_argument("--batch\_size", type=int, default=32,

                       help="批次大小")

    parser.add\_argument("--lr", type=float, default=5e-4,

                       help="学习率")

    parser.add\_argument("--weight\_decay", type=float, default=1e-4,

                       help="L2正则化系数")

    parser.add\_argument("--patience", type=int, default=7,

                       help="早停耐心值（连续多少轮不改善就停止）")

    parser.add\_argument("--min\_improvement", type=float, default=1e-5,

                       help="F1分数改善的最小阈值")

结果：