Master in Data Science

D5: NN-based DDI

Mining Unstructured Data

Year
2023/24

Date

Team members
Alícia Chimeno Sarabia and Gabriel Vayá Abad

1 Introduction

This task revolves around **relation extraction**, aiming to extract relations between entities expressed in the text. Specifically, the objective is to extract relationships between drugs from biomedical texts, known as **drug_interaction(drug,drug)**.

This task also pursues a classification task of each drug-drug interaction according to one of the four following types: **advise, effect, mechanism, int**. Accordingly, the challenge called for a five-way classification of sentences for each drug-pair:

- Advise: the sentence notes a recommendation or advice related to the concomitant use of the two drugs. Ex: UROXATRAL should NOT be used in combination with other alpha-blockers.
- Effect: the sentence states the effect of the drug interaction, including pharmacodynamic effect or mechanism of interaction. Ex: Quinolones may enhance the effects of the oral anticoagulant, warfarin.
- Mechanism: the sentence describes a pharmacokinetic mechanism. Ex: Grepafloxacin is a competitive inhibitor of the metabolism of theophylline.
- Int: the sentence mentions a drug interaction but doesn't provide any additional information. Ex: "The interaction of omeprazole and ketoconazole has been established.").
- None: the sentence does not show an interaction between the two drugs.

To achieve accurate relation extraction and classification, we employ neural networks, exploring various architectures and configurations. By experimenting with different models and hyperparameters, we aim to determine the optimal setup that maximizes the performance of relation extraction from biomedical texts.

2 Experiments description

2.1 Experimented Architectures and Hyperparameters

2.1.1 Embedding Dimension

We experimented with different embedding dimensions to evaluate their impact on the model's performance.

2.1.2 Max Length and Suffix Length Values

Different maximum length and suffix length values were experimented with to determine the optimal configuration for input sequences.

2.2 Experimented Input Information

2.2.1 Number of LSTM Units

We experimented with different numbers of LSTM units (e.g., 30, 64, 128) to identify the best configuration for capturing sequential dependencies.

2.2.2 Used Optimizer

Various optimizers such as Adam, RMSprop, and SGD were tested to find the most effective optimization strategy for training the model.

2.2.3 Number and Kind of Layers or Activation Functions

The number and types of layers (e.g., LSTM, Conv1D) and activation functions were varied to explore different architectural configurations.

2.3 Combining LSTM and CNN Layers

We experimented with combinations of LSTM and Conv1D layers, adjusting parameters like the number of filters and kernel size to potentially capture more complex patterns.

2.4 Adding Dropout

Dropout layers with different rates were added to prevent overfitting and improve the generalization capability of the model.

2.5 Evaluation Metrics

The performance of different configurations was evaluated using precision, recall, and F1-score to ensure a comprehensive assessment of the models.

3 Code Description

3.1 Best Network Model

```python
from tensorflow.keras.layers import Input, Embedding, Conv1D, GlobalMaxPooling1D,
    Bidirectional, LSTM, concatenate, Dropout, Dense
from tensorflow.keras.models import Model

def build_network(codes):
    # sizes
    n_words = codes.get_n_words()
    max_len = codes.maxlen
    n_labels = codes.get_n_labels()

    # Input Layer
    inpt = Input(shape=(max_len,))
    emb = Embedding(input_dim=n_words, output_dim=300, input_length=max_len)(inpt)

    # Convolutional Layers
    conv1 = Conv1D(filters=128, kernel_size=5, activation='relu')(emb)
    conv2 = Conv1D(filters=128, kernel_size=5, activation='relu')(conv1)
    pool = GlobalMaxPooling1D()(conv2)

    # Bidirectional LSTM Layer
    lstm = Bidirectional(LSTM(units=128, return_sequences=True))(emb)
    lstm = Bidirectional(LSTM(units=128, return_sequences=False))(lstm)

    # Feature Combination
    combined = concatenate([pool, lstm])
    dropout = Dropout(0.5)(combined)

    # Additional Dense Layer
    dense = Dense(256, activation='relu')(dropout)

    # Output Layer
    out = Dense(n_labels, activation='softmax')(dense)

    model = Model(inputs=inpt, outputs=out)
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['
    accuracy'])

    return model
```

Listing 1: Best Network Model Code

3.1.1 Explanation

The final model architecture consists of the following components:

- **Input Layer**: The input layer expects sequences of a maximum length (`max_len`).
- **Embedding Layer**: Converts input tokens into dense vectors of fixed size (300).
- **Convolutional Layers**: Two Conv1D layers with 128 filters and a kernel size of 5, both using the ReLU activation function, followed by a GlobalMaxPooling1D layer to reduce the dimensionality of the output.
- **Bidirectional LSTM Layers**: Two Bidirectional LSTM layers with 128 units each. The first LSTM layer returns sequences, while the second does not.
- **Feature Combination**: The outputs from the convolutional and LSTM layers are concatenated.
- **Dropout Layer**: A dropout layer with a dropout rate of 0.5 to prevent overfitting.

- **Dense Layer**: A dense layer with 256 units and ReLU activation.
- **Output Layer**: A dense layer with a number of units equal to the number of labels (`n_labels`) using the softmax activation function to output probability distributions over the classes.

The model is compiled using categorical cross-entropy loss and the Adam optimizer, with accuracy as the evaluation metric.

4 Experiments and results

To determine the optimal model, it's essential to assess both the machine learning algorithm and the capabilities of the feature extractor. Our evaluation of experiments will focus on key metrics such as **precision**, **recall**, and **F-1 score**.

4.1 Initial state

```
                    tp   fp   fn #pred #exp P    R    F1
------------------------------------------------------------------------------
brand               47    2  327   49  374 95.9% 12.6% 22.2%
drug              1561  123  345 1684 1906 92.7% 81.9% 87.0%
drug_n               7   43   38   50   45 14.0% 15.6% 14.7%
group              530   98  157  628  687 84.4% 77.1% 80.6%
------------------------------------------------------------------------------
M.avg             -----                   71.8% 46.8% 51.1%
------------------------------------------------------------------------------
m.avg             2145  266  867 2411 3012 89.0% 71.2% 79.1%
m.avg(no class)   2220  191  792 2411 3012 92.1% 73.7% 81.9%
```

Figure 1: Initial state

4.2 Embedding dimension

4.2.1 50

```
               tp   fp   fn #pred #exp P R F1
------------------------------------------------------------------------------
advise              72   34   69  106  141 67.9% 51.1% 58.3%
effect             134   69  178  203  312 66.0% 42.9% 52.0%
int                 14    3   14   17   28 82.4% 50.0% 62.2%
mechanism           82   77  179  159  261 51.6% 31.4% 39.0%
------------------------------------------------------------------------------
M.avg             -----67.0% 43.9% 52.9%
------------------------------------------------------------------------------
m.avg              302  183  440  485  742 62.3% 40.7% 49.2%
m.avg(no class)    329  156  413  485  742 67.8% 44.3% 53.6%
```

4.2.2 300

```
          tp   fp   fn #pred #exp P R F1
```

```
-----------------------------------------------------------------------------
advise              84   56   57  140   141 60.0% 59.6% 59.8%
effect             155  130  157  285   312 54.4% 49.7% 51.9%
int                 15    2   13   17    28 88.2% 53.6% 66.7%
mechanism           82   71  179  153   261 53.6% 31.4% 39.6%
-----------------------------------------------------------------------------
M.avg            -----64.1% 48.6% 54.5%
-----------------------------------------------------------------------------
m.avg              336  259  406  595   742 56.5% 45.3% 50.3%
m.avg(no class)    387  208  355  595   742 65.0% 52.2% 57.9%
```

Embedding Dimension	Precision	Recall	F1 Score
150	71.8%	46.8%	51.1%
50	67.0%	43.9%	52.9%
300	64.1%	48.6%	54.5%

Table 1: Embedding Dimension

4.3 Max length and sufix length values

4.3.1 50

```
       tp   fp   fn #pred #exp P R F1
-----------------------------------------------------------------------------
advise              72   39   69  111   141 64.9% 51.1% 57.1%
effect             139   66  173  205   312 67.8% 44.6% 53.8%
int                 14    4   14   18    28 77.8% 50.0% 60.9%
mechanism           82  109  179  191   261 42.9% 31.4% 36.3%
-----------------------------------------------------------------------------
M.avg            -----63.3% 44.3% 52.0%
-----------------------------------------------------------------------------
m.avg              307  218  435  525   742 58.5% 41.4% 48.5%
m.avg(no class)    328  197  414  525   742 62.5% 44.2% 51.8%
```

4.3.2 300

```
       tp   fp   fn #pred #exp P R F1
-----------------------------------------------------------------------------
advise              79   43   62  122   141 64.8% 56.0% 60.1%
effect             162  105  150  267   312 60.7% 51.9% 56.0%
int                 14    2   14   16    28 87.5% 50.0% 63.6%
mechanism           83   68  178  151   261 55.0% 31.8% 40.3%
-----------------------------------------------------------------------------
M.avg            -----67.0% 47.4% 55.0%
-----------------------------------------------------------------------------
m.avg              338  218  404  556   742 60.8% 45.6% 52.1%
m.avg(no class)    372  184  370  556   742 66.9% 50.1% 57.3%
```

Max Length	Precision	Recall	F1 Score
150	71.8%	46.8%	51.1%
50	63.3%	44.3%	52.0%
300	67.0%	47.4%	55.0%

Table 2: Max Length

4.4 Conv1D Layer

Increased the **number of filters** and **kernel size** to potentially capture more complex patterns.

4.5 Number of LSTM units

We experimented with different numbers of LSTM units (30, 64, 128).

LSTM layer w 128 units

```
     tp    fp    fn #pred #exp  P  R  F1
------------------------------------------------------------------------------
advise              71    52    70   123   141 57.7% 50.4% 53.8%
effect             177   109   135   286   312 61.9% 56.7% 59.2%
int                 17     7    11    24    28 70.8% 60.7% 65.4%
mechanism          118    66   143   184   261 64.1% 45.2% 53.0%
------------------------------------------------------------------------------
M.avg            -----63.6% 53.3% 57.9%
------------------------------------------------------------------------------
m.avg              383   234   359   617   742 62.1% 51.6% 56.4%
m.avg(no class)    462   155   280   617   742 74.9% 62.3% 68.0%
```

Add a LSTM layer w 64 units

```
             tp    fp    fn #pred #exp  P  R  F1
------------------------------------------------------------------------------
advise             100    93    41   193   141 51.8% 70.9% 59.9%
effect             207   202   105   409   312 50.6% 66.3% 57.4%
int                 18     2    10    20    28 90.0% 64.3% 75.0%
mechanism          166   153    95   319   261 52.0% 63.6% 57.2%
------------------------------------------------------------------------------
M.avg            -----61.1% 66.3% 62.4%
------------------------------------------------------------------------------
m.avg              491   450   251   941   742 52.2% 66.2% 58.3%
m.avg(no class)    583   358   159   941   742 62.0% 78.6% 69.3%
```

Add a LSTM layer w 30 units

```
          tp    fp    fn #pred #exp  P  R  F1
------------------------------------------------------------------------------
advise              88    75    53   163   141 54.0% 62.4% 57.9%
effect             205   155   107   360   312 56.9% 65.7% 61.0%
int                 17     7    11    24    28 70.8% 60.7% 65.4%
mechanism          114    47   147   161   261 70.8% 43.7% 54.0%
------------------------------------------------------------------------------
M.avg            -----63.1% 58.1% 59.6%
------------------------------------------------------------------------------
m.avg              424   284   318   708   742 59.9% 57.1% 58.5%
m.avg(no class)    489   219   253   708   742 69.1% 65.9% 67.4%
```

LSTM Units	Precision	Recall	F1 Score
128	63.6%	53.3%	57.9%
64	61.1%	66.3%	62.4%
30	63.1%	58.1%	59.6%

Table 3: Number of LSTM Units

4.6 Combinations of LSTM and CNN layers

30 ,30

```
     tp   fp   fn #pred #exp P R F1
------------------------------------------------------------------------------
advise             93   63   48   156   141 59.6% 66.0% 62.6%
effect            199  129  113   328   312 60.7% 63.8% 62.2%
int                18    1   10    19    28 94.7% 64.3% 76.6%
mechanism         128   90  133   218   261 58.7% 49.0% 53.4%
------------------------------------------------------------------------------
M.avg            -----68.4% 60.8% 62.7%
------------------------------------------------------------------------------
m.avg             438  283  304   721   742 60.7% 59.0% 59.9%
m.avg(no class)   510  211  232   721   742 70.7% 68.7% 69.7%
```

60 - 60

```
             tp   fp   fn #pred #exp P R F1
------------------------------------------------------------------------------
advise             83   53   58   136   141 61.0% 58.9% 59.9%
effect            171   69  141   240   312 71.2% 54.8% 62.0%
int                16    3   12    19    28 84.2% 57.1% 68.1%
mechanism          99   48  162   147   261 67.3% 37.9% 48.5%
------------------------------------------------------------------------------
M.avg            -----71.0% 52.2% 59.6%
------------------------------------------------------------------------------
m.avg             369  173  373   542   742 68.1% 49.7% 57.5%
m.avg(no class)   423  119  319   542   742 78.0% 57.0% 65.9%
```

16-16

```
        tp   fp   fn #pred #exp P R F1
------------------------------------------------------------------------------
advise             95   80   46   175   141 54.3% 67.4% 60.1%
effect            169  109  143   278   312 60.8% 54.2% 57.3%
int                17    2   11    19    28 89.5% 60.7% 72.3%
mechanism         108   61  153   169   261 63.9% 41.4% 50.2%
------------------------------------------------------------------------------
M.avg            -----67.1% 55.9% 60.0%
------------------------------------------------------------------------------
m.avg             389  252  353   641   742 60.7% 52.4% 56.3%
m.avg(no class)   450  191  292   641   742 70.2% 60.6% 65.1%
```

Adding dropout of 0.1

```
     tp   fp   fn #pred #exp P R F1
------------------------------------------------------------------------------
advise             69   53   72   122   141 56.6% 48.9% 52.5%
effect            197  144  115   341   312 57.8% 63.1% 60.3%
int                16    9   12    25    28 64.0% 57.1% 60.4%
```

```
mechanism          92   85  169  177   261 52.0% 35.2% 42.0%
------------------------------------------------------------------------------
M.avg             -----57.6% 51.1% 53.8%
------------------------------------------------------------------------------
m.avg             374  291  368  665   742 56.2% 50.4% 53.2%
m.avg(no class)   448  217  294  665   742 67.4% 60.4% 63.7%
```

kernel=5

Combination	Precision	Recall	F1 Score
30, 30	68.2%	60.8%	62.7%
64, 64	71.0%	52.2%	59.6%
16, 16	67.1%	55.9%	60.0%
30, 30 + Dropout 0.1	57.6%	51.1%	53.8%

Table 4: Combining LSTM and CNN Layers

4.7 Final model

The final model insights are described in the section 3.1.1. The network starts with an input layer for DDI sentences, followed by embedding and convolutional layers extracting features. Bidirectional LSTM layers capture context, and a dropout layer prevents overfitting. Concatenation and dense layers process features, with softmax output providing interaction category probabilities. This architecture effectively combines convolutional and recurrent layers for accurate drug-drug interaction classification.

```
       tp    fp    fn #pred #exp  P  R  F1
------------------------------------------------------------------------------
advise             96   81   45  177   141 54.2% 68.1% 60.4%
effect            178   76  134  254   312 70.1% 57.1% 62.9%
int                21    9    7   30    28 70.0% 75.0% 72.4%
mechanism         105   64  156  169   261 62.1% 40.2% 48.8%
------------------------------------------------------------------------------
M.avg             -----64.1% 60.1% 63.1%
------------------------------------------------------------------------------
m.avg             400  230  342  630   742 63.5% 53.9% 58.3%
m.avg(no class)   467  163  275  630   742 74.1% 62.9% 68.1%
```

4.8 Test

After training our final model, we evaluated its performance on the test dataset. The table below summarizes the results:

```
         tp    fp    fn #pred #exp  P  R  F1
------------------------------------------------------------------------------
advise            116   59   93  175   209 66.3% 55.5% 60.4%
effect            168  104  118  272   286 61.8% 58.7% 60.2%
int                 6    8   19   14    25 42.9% 24.0% 30.8%
mechanism         171  114  169  285   340 60.0% 50.3% 54.7%
------------------------------------------------------------------------------
M.avg             -----57.7% 47.1% 51.5%
------------------------------------------------------------------------------
m.avg             461  285  399  746   860 61.8% 53.6% 57.4%
m.avg(no class)   531  215  329  746   860 71.2% 61.7% 66.1%
```

4.9 Conclusions

In summary, our project centered on developing a neural network-based approach for classifying drug-drug interactions (DDIs) from biomedical texts. After conducting a series of experiments and optimizations, we designed an advanced model that combines convolutional and recurrent layers. This model effectively captures both local and contextual information, yielding promising results across precision, recall, and F1-score metrics for various interaction categories. By leveraging deep learning techniques, our approach demonstrates the potential of neural networks in accurately extracting and classifying DDIs from unstructured biomedical data, contributing significantly to pharmacovigilance and drug safety research.