

# Relazione sull'implementazione di un'RNN

A.A. 2023/2024

Rosso Carlo

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Descrizione del problema . . . . .	2
1.2	Dataset . . . . .	2
<b>2</b>	<b>Descrizione dei modelli</b>	<b>2</b>
<b>3</b>	<b>Risultati</b>	<b>4</b>
3.1	25 unità nascoste . . . . .	4
	Label Confusion Matrix . . . . .	4
	Label Metrics . . . . .	4
	Root Label Confusion Matrix . . . . .	5
	Root Label Metrics . . . . .	5
3.2	50 unità nascoste . . . . .	5
	Label Confusion Matrix . . . . .	5
	Label Metrics . . . . .	5
	Root Label Confusion Matrix . . . . .	6
	Root Label Metrics . . . . .	6
3.3	75 unità nascoste . . . . .	6
	Label Confusion Matrix . . . . .	6
	Label Metrics . . . . .	7
	Root Label Confusion Matrix . . . . .	7
	Root Label Metrics . . . . .	7
3.4	100 unità nascoste . . . . .	7
	Label Confusion Matrix . . . . .	7
	Label Metrics . . . . .	7
	Root Label Confusion Matrix . . . . .	8
	Root Label Metrics . . . . .	8
<b>4</b>	<b>Conclusioni</b>	<b>8</b>

# 1 Introduzione

## 1.1 Descrizione del problema

Il task affrontato dai modelli allenati sullo Sentiment Penn Treebank riguarda la classificazione del sentimento di frasi e frasi parziali (nodi) all'interno degli alberi sintattici. In particolare, il task principale è la rilevazione del sentimento espresso in ogni frase, che può variare da negativo a positivo su una scala a cinque livelli:

- negativo (- -);
- un po' negativo (-);
- neutro (0);
- un po' positivo (+);
- positivo (+ +);

In particolare il task si compone di tre sotto-task:

- **Classificazione del Sentimento Fine-Grained:** I modelli devono prevedere l'etichetta di sentimento per ogni frase e sottofrase all'interno di un albero sintattico. Le etichette sono suddivise in cinque categorie.
- **Composizionalità del Sentimento:** Il task richiede ai modelli di catturare e comporre correttamente i sentimenti delle sottofrasi per prevedere il sentimento della frase completa. Ciò include la gestione di fenomeni linguistici complessi come la negazione e le congiunzioni contrastive.
- **Analisi degli Alberi Sintattici:** Utilizzando gli alberi di parsing sintattico generati dal Stanford Parser, i modelli devono processare e analizzare ogni nodo per comprendere come il sentimento si propaga attraverso la struttura dell'albero.

## 1.2 Dataset

La descrizione del dataset viene omessa in quanto presente in un documento a parte.

# 2 Descrizione dei modelli

Viene richiesto di implementare la *Recursive Neural Network* (RNN) descritta nel paper Socher et al. 2013. Quindi l'implementazione viene svolta a partire dal codice fornito dagli autori del paper. In particolare seguono i passaggi svolti per l'implementazione:

1. **Clone del repository:** ho clonato il repository <https://github.com/stanfordnlp/CoreNLP.git>;
2. **Installazione delle dipendenze:** ho seguito le istruzioni nel README del repository per installare le dipendenze necessarie;

3. **Check del modello fornito:** ho verificato che il modello fornito funzionasse correttamente con il comando
- ```
java -cp "*" -mx5g edu.stanford.nlp.sentiment.SentimentPipeline  
-file examples/sample-maven-project/sample-english.txt
```
- Non ha funzionato;
4. **Correzione:** ho sostituito il comando con
- ```
mvn exec:java -Dexec.mainClass="edu.stanford.nlp.sentiment.SentimentPipeline"  
-Dexec.args="-file examples/sample-maven-project/sample-english.txt"
```
- e ha funzionato;
5. **Lettura del codice:** ho letto un po' del codice per orientarmi nella codebase. Ho capito che il modello che tratta il problema che mi interessa si trova all'interno del package `edu.stanford.nlp.sentiment`;
6. **Implementazione del modello:** ho individuato la classe che si occupa dell'allenamento del modello, ovvero `SentimentTraining`;
7. **Studio di `SentimentTraining`:** ho studiato il codice di `SentimentTraining` per capire come funziona il training del modello. Non solo, ho anche compreso quali sono i parametri in input della classe e come sono utilizzati;
8. **Costruzione del comando di allenamento:** ho costruito il seguente comando per allenare il modello:
- ```
mvn exec:java -Dexec.mainClass="edu.stanford.nlp.sentiment.SentimentTraining"  
-Dexec.args="-train -model rnn.ser.gz -trainpath train.txt -devpath dev.txt  
-nousetensors -lowercasewordvectors -numhid 100 -randomseed 42"
```
- In particolare, la classe specificata si occupa di allenare un modello generico. Di seguito spiego gli argomenti passati al comando:
- `-train`: specifica che si vuole allenare il modello;
  - `-model ../rnn.ser.gz`: specifica il percorso in cui salvare il modello allenato;
  - `-trainpath ../train.txt`: specifica il percorso del file o della directory di training;
  - `-devpath ../dev.txt`: specifica il percorso del file o della directory di validazione;
  - `-nousetensors`: specifica che non si vogliono utilizzare i tensori. In questo modo si passa dall'RNTN descritto nel paper all'RNN, proprio come spiegato all'interno del paper medesimo;
  - `-lowercasewordvectors`: specifica che si vogliono utilizzare i vettori delle parole in minuscolo, si tratta di una scelta personale, a dire il vero non so se sia stata usata anche dagli autori del paper;
  - `-numhid 100`: specifica il numero di hidden units;
  - `-randomseed 42`: specifica il seed per la generazione dei numeri casuali, in modo da rendere riproducibile l'esperimento.

9. **Setup dei dataset:** ho scaricato i dataset nel formato opportuno, si trovano al seguente indirizzo: [https://nlp.stanford.edu/sentiment/trainDevTestTrees\\_PTB.zip](https://nlp.stanford.edu/sentiment/trainDevTestTrees_PTB.zip). Poi li ho scompattati e li ho spostati nella directory del progetto;
10. **Allenamento del modello:** ho eseguito il comando di allenamento del modello. Il training è durato circa 6 ore;
11. **Test del modello:** ho testato il modello con il comando
  - mvn exec:java -Dexec.mainClass="edu.stanford.nlp.sentiment.Evaluate"
  - Dexec.args="-model rnn.ser.gz -treebank test.txt" .

## 3 Risultati

Ho deciso di implementare 4 versioni del modello, rispettivamente con 25, 50, 75 e 100 hidden units. Riporto le matrici di confusione, la precision, la recall e l'f1-score per ciascun modello, oltre che l'accuratezza e l'accuratezza delle radici.

### 3.1 25 unità nascoste

- **Label Accuracy:** 0.784891 %;
- **Root Label Accuracy:** 0.396833 %;

#### Label Confusion Matrix

| Guess/Gold   | 0    | 1    | 2     | 3     | 4    | Marg. (Guess) |
|--------------|------|------|-------|-------|------|---------------|
| 0            | 809  | 1028 | 696   | 329   | 105  | 2967          |
| 1            | 864  | 5340 | 2930  | 834   | 154  | 10122         |
| 2            | 257  | 2378 | 50767 | 2859  | 348  | 56609         |
| 3            | 47   | 391  | 1875  | 5959  | 1227 | 9499          |
| 4            | 31   | 118  | 280   | 1017  | 1957 | 3403          |
| Marg. (Gold) | 2008 | 9255 | 56548 | 10998 | 3791 |               |

#### Label Metrics

- **Class 0:** Precision = 0.27267, Recall = 0.40289, Specificity = 0.97322, F1-score = 0.32523
- **Class 1:** Precision = 0.52756, Recall = 0.57699, Specificity = 0.9348, F1-score = 0.55117
- **Class 2:** Precision = 0.8968, Recall = 0.89777, Specificity = 0.77576, F1-score = 0.89728
- **Class 3:** Precision = 0.62733, Recall = 0.54183, Specificity = 0.95056, F1-score = 0.58145
- **Class 4:** Precision = 0.57508, Recall = 0.51622, Specificity = 0.98165, F1-score = 0.54406

### Root Label Confusion Matrix

| Guess/Gold   | 0   | 1   | 2   | 3   | 4   | Marg. (Guess) |
|--------------|-----|-----|-----|-----|-----|---------------|
| 0            | 129 | 198 | 90  | 71  | 30  | 518           |
| 1            | 124 | 311 | 129 | 99  | 30  | 693           |
| 2            | 17  | 67  | 72  | 52  | 20  | 228           |
| 3            | 5   | 41  | 75  | 150 | 104 | 375           |
| 4            | 4   | 16  | 23  | 138 | 215 | 396           |
| Marg. (Gold) | 279 | 633 | 389 | 510 | 399 |               |

### Root Label Metrics

- **Class 0:** Precision = 0.24903, Recall = 0.46237, Specificity = 0.79855, F1-score = 0.32371
- **Class 1:** Precision = 0.44877, Recall = 0.49131, Specificity = 0.75777, F1-score = 0.46908
- **Class 2:** Precision = 0.31579, Recall = 0.18509, Specificity = 0.91433, F1-score = 0.23339
- **Class 3:** Precision = 0.4, Recall = 0.29412, Specificity = 0.86765, F1-score = 0.33898
- **Class 4:** Precision = 0.54293, Recall = 0.53885, Specificity = 0.90006, F1-score = 0.54088

### 3.2 50 unità nascoste

- **Label Accuracy:** 0.794286 %;
- **Root Label Accuracy:** 0.424434 %;

### Label Confusion Matrix

| Guess/Gold   | 0    | 1    | 2     | 3     | 4    | Marg. (Guess) |
|--------------|------|------|-------|-------|------|---------------|
| 0            | 448  | 289  | 53    | 11    | 7    | 808           |
| 1            | 999  | 4746 | 1896  | 480   | 98   | 8219          |
| 2            | 405  | 3436 | 52196 | 3213  | 346  | 59596         |
| 3            | 144  | 748  | 2319  | 6741  | 1863 | 11815         |
| 4            | 12   | 36   | 84    | 553   | 1477 | 2162          |
| Marg. (Gold) | 2008 | 9255 | 56548 | 10998 | 3791 |               |

### Label Metrics

- **Class 0:** Precision = 0.55446, Recall = 0.22311, Specificity = 0.99553, F1-score = 0.31818
- **Class 1:** Precision = 0.57744, Recall = 0.5128, Specificity = 0.95265, F1-score = 0.54321

- **Class 2:** Precision = 0.87583, Recall = 0.92304, Specificity = 0.71595, F1-score = 0.89882
- **Class 3:** Precision = 0.57055, Recall = 0.61293, Specificity = 0.92914, F1-score = 0.59098
- **Class 4:** Precision = 0.68316, Recall = 0.38961, Specificity = 0.99131, F1-score = 0.49622

#### Root Label Confusion Matrix

| Guess/Gold   | 0   | 1   | 2   | 3   | 4   | Marg. (Guess) |
|--------------|-----|-----|-----|-----|-----|---------------|
| 0            | 60  | 64  | 10  | 1   | 2   | 137           |
| 1            | 152 | 317 | 117 | 67  | 13  | 666           |
| 2            | 38  | 125 | 120 | 73  | 26  | 382           |
| 3            | 27  | 123 | 131 | 293 | 210 | 784           |
| 4            | 2   | 4   | 11  | 76  | 148 | 241           |
| Marg. (Gold) | 279 | 633 | 389 | 510 | 399 |               |

#### Root Label Metrics

- **Class 0:** Precision = 0.43796, Recall = 0.21505, Specificity = 0.96012, F1-score = 0.28846
- **Class 1:** Precision = 0.47598, Recall = 0.50079, Specificity = 0.77869, F1-score = 0.48807
- **Class 2:** Precision = 0.31414, Recall = 0.30848, Specificity = 0.85612, F1-score = 0.31128
- **Class 3:** Precision = 0.37372, Recall = 0.57451, Specificity = 0.71118, F1-score = 0.45286
- **Class 4:** Precision = 0.61411, Recall = 0.37093, Specificity = 0.94865, F1-score = 0.4625

### 3.3 75 unità nascoste

- **Label Accuracy:** 0.792264 %;
- **Root Label Accuracy:** 0.387783 %;

#### Label Confusion Matrix

| Guess/Gold   | 0    | 1    | 2     | 3     | 4    | Marg. (Guess) |
|--------------|------|------|-------|-------|------|---------------|
| 0            | 682  | 687  | 246   | 107   | 42   | 1764          |
| 1            | 835  | 4721 | 1919  | 568   | 177  | 8220          |
| 2            | 429  | 3582 | 53098 | 4300  | 643  | 62052         |
| 3            | 54   | 235  | 1187  | 5556  | 1545 | 8577          |
| 4            | 8    | 30   | 98    | 467   | 1384 | 1987          |
| Marg. (Gold) | 2008 | 9255 | 56548 | 10998 | 3791 |               |

### Label Metrics

- **Class 0:** Precision = 0.38662, Recall = 0.33964, Specificity = 0.98657, F1-score = 0.36161
- **Class 1:** Precision = 0.57433, Recall = 0.5101, Specificity = 0.95229, F1-score = 0.54031
- **Class 2:** Precision = 0.8557, Recall = 0.93899, Specificity = 0.6563, F1-score = 0.89541
- **Class 3:** Precision = 0.64778, Recall = 0.50518, Specificity = 0.95781, F1-score = 0.56766
- **Class 4:** Precision = 0.69653, Recall = 0.36508, Specificity = 0.99235, F1-score = 0.47906

### Root Label Confusion Matrix

| Guess/Gold   | 0   | 1   | 2   | 3   | 4   | Marg. (Guess) |
|--------------|-----|-----|-----|-----|-----|---------------|
| 0            | 125 | 171 | 55  | 34  | 11  | 396           |
| 1            | 108 | 269 | 108 | 83  | 42  | 610           |
| 2            | 35  | 158 | 153 | 145 | 47  | 538           |
| 3            | 10  | 29  | 59  | 195 | 184 | 477           |
| 4            | 1   | 6   | 14  | 53  | 115 | 189           |
| Marg. (Gold) | 279 | 633 | 389 | 510 | 399 |               |

### Root Label Metrics

- **Class 0:** Precision = 0.31566, Recall = 0.44803, Specificity = 0.85966, F1-score = 0.37037
- **Class 1:** Precision = 0.44098, Recall = 0.42496, Specificity = 0.78377, F1-score = 0.43282
- **Class 2:** Precision = 0.28439, Recall = 0.39332, Specificity = 0.78858, F1-score = 0.3301
- **Class 3:** Precision = 0.40881, Recall = 0.38235, Specificity = 0.83412, F1-score = 0.39514
- **Class 4:** Precision = 0.60847, Recall = 0.28822, Specificity = 0.95914, F1-score = 0.39116

## 3.4 100 unità nascoste

- **Label Accuracy:** 0.782930 %;
- **Root Label Accuracy:** 0.397738 %;

### Label Confusion Matrix

#### Label Metrics

- **Class 0:** Precision = 0.50165, Recall = 0.22759, Specificity = 0.99437, F1-score = 0.31312
- **Class 1:** Precision = 0.58236, Recall = 0.5195, Specificity = 0.95299, F1-score = 0.54914



| Guess/Gold   | 0    | 1    | 2     | 3     | 4    | Marg. (Guess) |
|--------------|------|------|-------|-------|------|---------------|
| 0            | 457  | 345  | 79    | 25    | 5    | 911           |
| 1            | 973  | 4808 | 1961  | 435   | 79   | 8256          |
| 2            | 242  | 2533 | 50450 | 2459  | 222  | 55906         |
| 3            | 251  | 1302 | 3610  | 6734  | 1264 | 13161         |
| 4            | 85   | 267  | 448   | 1345  | 2221 | 4366          |
| Marg. (Gold) | 2008 | 9255 | 56548 | 10998 | 3791 |               |

- **Class 2:** Precision = 0.90241, Recall = 0.89216, Specificity = 0.79057, F1-score = 0.89726
- **Class 3:** Precision = 0.51166, Recall = 0.61229, Specificity = 0.91024, F1-score = 0.55747
- **Class 4:** Precision = 0.5087, Recall = 0.58586, Specificity = 0.97278, F1-score = 0.54456

#### Root Label Confusion Matrix

| Guess/Gold   | 0   | 1   | 2   | 3   | 4   | Marg. (Guess) |
|--------------|-----|-----|-----|-----|-----|---------------|
| 0            | 47  | 62  | 11  | 3   | 1   | 124           |
| 1            | 142 | 273 | 92  | 38  | 10  | 555           |
| 2            | 16  | 58  | 53  | 26  | 12  | 165           |
| 3            | 46  | 176 | 159 | 212 | 82  | 675           |
| 4            | 28  | 64  | 74  | 231 | 294 | 691           |
| Marg. (Gold) | 279 | 633 | 389 | 510 | 399 |               |

#### Root Label Metrics

- **Class 0:** Precision = 0.37903, Recall = 0.16846, Specificity = 0.96012, F1-score = 0.23325
- **Class 1:** Precision = 0.49189, Recall = 0.43128, Specificity = 0.82118, F1-score = 0.4596
- **Class 2:** Precision = 0.32121, Recall = 0.13625, Specificity = 0.9385, F1-score = 0.19134
- **Class 3:** Precision = 0.31407, Recall = 0.41569, Specificity = 0.72765, F1-score = 0.35781
- **Class 4:** Precision = 0.42547, Recall = 0.73684, Specificity = 0.78078, F1-score = 0.53945

## 4 Conclusioni

I risultati ottenuti confermano le affermazioni del paper Socher et al. 2013. Infatti, il modello RNN dimostra una notevole robustezza anche variando il numero di unità nascoste. Inoltre, l'accuratezza sui singoli nodi si attesta intorno all'80%, in linea con quanto riportato nel paper, che indica un'accuratezza del 79.0% per la classificazione fine-grained. Nel nostro esperimento, il modello con 50 unità nascoste ha raggiunto un'accuratezza del 79.4%, un

valore molto vicino a quello del paper.

Anche l'accuratezza relativa al label della root è comparabile: il paper riporta un'accuratezza del 43.2%, mentre il nostro modello con 50 unità nascoste ha raggiunto il 42.2%. Questi risultati suggeriscono che il modello è stato allenato correttamente e che le prestazioni ottenute sono soddisfacenti. Le differenze riscontrate potrebbero essere dovute all'inizializzazione dei pesi, poiché il seed utilizzato nel paper non è stato fornito, impedendoci di valutare se tali differenze derivino da questo aspetto.

## Riferimenti bibliografici

[Soc+13] Richard Socher et al. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Tree-bank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. A cura di David Yarowsky et al. Seattle, Washington, USA: Association for Computational Linguistics, ott. 2013, pp. 1631–1642. URL: <https://aclanthology.org/D13-1170>.