```
[a_{i,j}^l]_{n\times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: V, \mathcal{E}: Node set and edge set of Attr tree
 1: \triangleright Initialize the state of all tokens, each token has three states:
     NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ⊳ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l\in E^l do
               if State_i is Appear and State_i is NotAppear then
11:
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
                    \mathcal{V} \leftarrow \mathcal{V} \cup \{j\}
                    State_i = \mathtt{Fixed}
14:
15:
                    State_i = Appear
16:
               if State_i is Fixed and State_i is NotAppear then
17:
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \cup \{j\}
19:
                     State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
               \mathcal{E} \leftarrow \mathcal{E} \bigcup \{ ([CLS], j) \}
24:
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```

Sentence: Seldom has a movie so closely matched the spirit of a man and his work



Model input is the attribution scores between each token in the sentence and the edges which tokens are related

```
[a_{i,j}^l]_{n\times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: V, \mathcal{E}: Node set and edge set of Attr tree
 1: ▷ Initialize the state of all tokens, each token has three states
      NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ▷ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l \in E^l do
11:
               if State_i is Appear and State_i is NotAppear then
                   \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
                  \mathcal{V} \leftarrow \mathcal{V} \cup \{j\}
13:
14:
                    State_i = \mathtt{Fixed}
15:
                    State_i = Appear
               if State_i is Fixed and State_i is NotAppear then
16:
17:
                   \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \cup \{j\}
19:
                    State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
23:
               \mathcal{E} \leftarrow \mathcal{E} \bigcup \{ ([CLS], j) \}
24:
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```

Status NotAppear Tokens

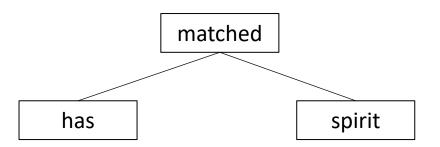
seldom	closely	а
has	matched	man
а	the	and
movie	spirit	his
SO	of	work

```
[a_{i,j}^l]_{n \times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: V, \mathcal{E}: Node set and edge set of Attr tree
 1: ▷ Initialize the state of all tokens, each token has three states:
      NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ▷ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l \in E^l do
10:
11:
               if State_i is Appear and State_i is NotAppear then
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
                  \mathcal{V} \leftarrow \mathcal{V} \cup \{j\}
13:
14:
                    State_i = \mathtt{Fixed}
15:
                    State_i = Appear
16:
               if State_i is Fixed and State_i is NotAppear then
17:
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
19:
                    State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
23:
               \mathcal{E} \leftarrow \mathcal{E} \bigcup \{ ([CLS], j) \}
24:
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```

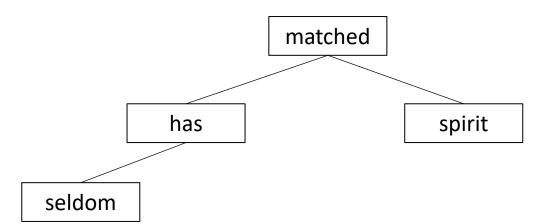
Attribution Tree

matched

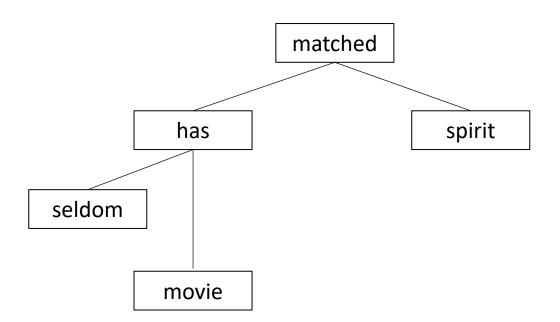
```
[a_{i,j}^l]_{n \times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: V, \mathcal{E}: Node set and edge set of Attr tree
 1: ▷ Initialize the state of all tokens, each token has three states:
      NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ⊳ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l \in E^l do
10:
               if State_i is Appear and State_i is NotAppear then
11:
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
                    \mathcal{V} \leftarrow \mathcal{V} \cup \{j\}
13:
14:
                    State_i = \mathtt{Fixed}
15:
                    State_j = Appear
               if State_i is Fixed and State_i is NotAppear then
16:
17:
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
19:
                     State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
23:
               \mathcal{E} \leftarrow \mathcal{E} \bigcup \{ ([CLS], j) \}
24:
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```



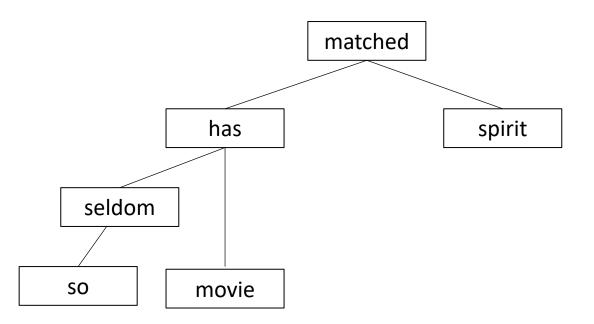
```
[a_{i,j}^l]_{n \times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: V, \mathcal{E}: Node set and edge set of Attr tree
 1: ▷ Initialize the state of all tokens, each token has three states:
      NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ⊳ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l \in E^l do
10:
11:
               if State_i is Appear and State_i is NotAppear then
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
                   \mathcal{V} \leftarrow \mathcal{V} \cup \{j\}
13:
                    State_i = \texttt{Fixed}
14:
15:
                    State_i = Appear
16:
               if State_i is Fixed and State_i is NotAppear then
17:
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
19:
                    State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
23:
24:
              \mathcal{E} \leftarrow \mathcal{E} \bigcup \{ ([CLS], j) \}
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```



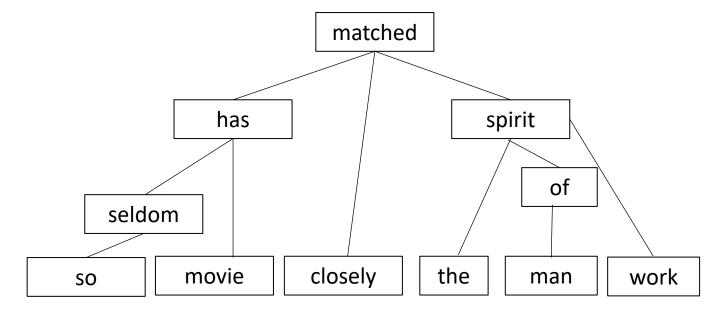
```
[a_{i,j}^l]_{n \times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: V, \mathcal{E}: Node set and edge set of Attr tree
 1: ▷ Initialize the state of all tokens, each token has three states:
      NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ⊳ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l \in E^l do
10:
11:
               if State_i is Appear and State_i is NotAppear then
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
                  \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
13:
14:
                    State_i = \texttt{Fixed}
15:
                    State_i = Appear
16:
               if State_i is Fixed and State_i is NotAppear then
17:
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
19:
                    State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
23:
24:
              \mathcal{E} \leftarrow \mathcal{E} \bigcup \{ ([CLS], j) \}
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```



```
[a_{i,j}^l]_{n \times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: V, \mathcal{E}: Node set and edge set of Attr tree
 1: ▷ Initialize the state of all tokens, each token has three states:
      NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ⊳ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l \in E^l do
10:
11:
               if State_i is Appear and State_i is NotAppear then
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
                  \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
13:
14:
                    State_i = \texttt{Fixed}
15:
                    State_i = Appear
16:
               if State_i is Fixed and State_i is NotAppear then
17:
                    \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
19:
                    State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
23:
24:
              \mathcal{E} \leftarrow \mathcal{E} \bigcup \{ ([CLS], j) \}
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```



```
[a_{i,j}^l]_{n \times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: V, \mathcal{E}: Node set and edge set of Attr tree
 1: ▷ Initialize the state of all tokens, each token has three states:
     NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ⊳ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l \in E^l do
10:
               if State_i is Appear and State_i is NotAppear then
                   \mathcal{E} \leftarrow \mathcal{E} \cup \{(i,j)\}
        \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
14:
                    State_i = \texttt{Fixed}
15:
                    State_i = Appear
16:
               if State_i is Fixed and State_i is NotAppear then
17:
                   \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
                    State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
24:
        \mathcal{E} \leftarrow \mathcal{E} \bigcup \{ ([CLS], j) \}
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```



```
[a_{i,j}^l]_{n\times n}: Attribution scores
Input:
             \{E^l\}_{l=1}^{|l|}: Retained attribution edges
Output: \mathcal{V}, \mathcal{E}: Node set and edge set of Attr tree
 1: ▷ Initialize the state of all tokens, each token has three states:
      NotAppear, Appear, Fixed
 2: for i \leftarrow n, \cdots, 1 do
          State_i = NotAppear
 4: ⊳ Choose the top node of the attribution tree
 5: [AttrAll_i]_n = \sum_{l=1}^{|l|} \sum_{j=1, j \neq i}^n a_{i,j}^l
 6: TopNode = argmax([AttrAll_i]_n)
 7: V \leftarrow \{TopNode\}; State_{TopNode} = Appear
 8: ▷ Build the attribution tree downward
 9: for l \leftarrow |l| - 1, \cdots, 1 do
          for (i,j)_{i\neq j}^l \in E^l do
10:
11:
               if State_i is Appear and State_i is NotAppear then
                   \mathcal{E} \leftarrow \mathcal{E} \cup \{(i,j)\}
13: V \leftarrow V \bigcup \{j\}
14:
                   State_i = \texttt{Fixed}
15:
                    State_i = Appear
16:
               if State_i is Fixed and State_i is NotAppear then
17:
                   \mathcal{E} \leftarrow \mathcal{E} \bigcup \{(i,j)\}
18:
                    \mathcal{V} \leftarrow \mathcal{V} \bigcup \{j\}
                    State_i = Appear
20: \triangleright Add the terminal of the information flow
21: \mathcal{V} \leftarrow \{ [CLS] \}
22: for j \leftarrow n, \cdots, 1 do
          if State_i \in \{Appear, Fixed\} then
      \mathcal{E} \leftarrow \mathcal{E} \cup \{([CLS], i)\}
25: return Tree = \{\mathcal{V}, \mathcal{E}\}
```

Attribution Tree [CLS] matched has spirit of seldom

closely

movie

SO

the

man

work