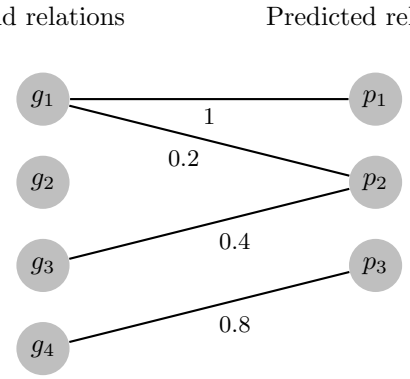
This year, we introduce partial matching for argument extraction as the exact match is found to be too stringent and might not reflect the true performance of the parser. Evaluation is done in four steps.

1 Compute matching score

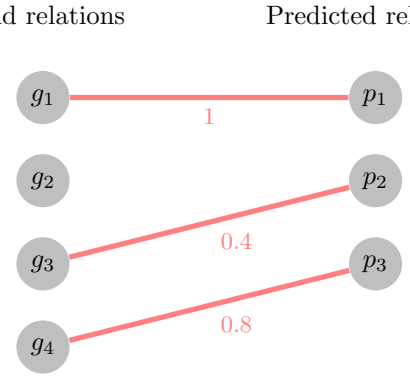
For all possible pair of gold and predicted relations (g_i, p_j), compute the average between F_1 score of Arg1 and F_1 score of Arg2 tokenwise. We can visualize the scores in bipartite graph. An edge only exists for a non-zero pair.

$$\text{Score}(g_i, p_j) = (F_1 \text{from Arg1} + F_1 \text{from Arg2})/2$$



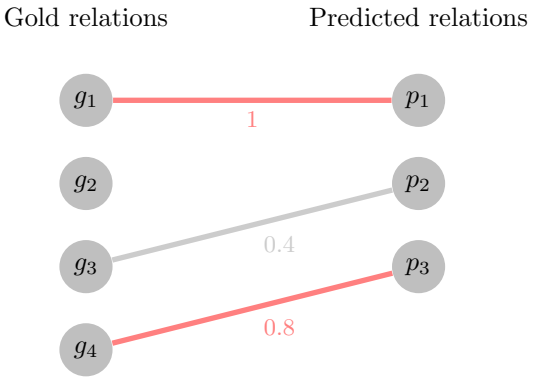
2 Prune and align

Prune the graph such that 1) the sum of the edges is maximized and 2) the final graph is a one-to-one mapping. The scorer will try all combinations to find the best alignment.



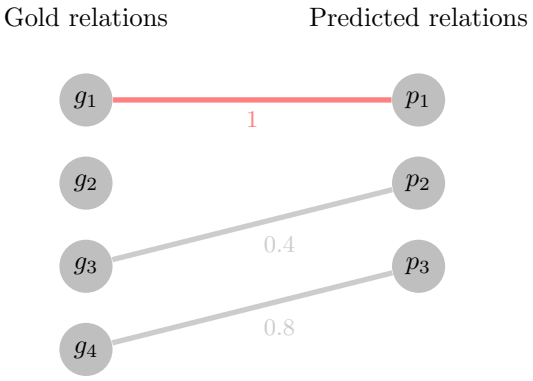
3 Thresholding

If the threshold for partial matching is 0.7, then we have correctly identified arguments for 2 relations. We can then compute evaluation metrics for argument extraction.



Precision $= \frac{2}{3} = 0.66$. Recall $= \frac{2}{4} = 0.5$. $F_1 = 2 * \frac{0.66*0.5}{0.66+0.5} = 0.56$.

If the threshold for partial matching is 1.0 i.e. the perfect matching we used in 2015, then we have correctly identified arguments for 1 relation.



Precision $= \frac{1}{3} = 0.33$. Recall $= \frac{1}{4} = 0.25$. $F_1 = 2 * \frac{0.33*0.25}{0.33+0.25} = 0.28$, which is a lot lower.

4 Score sense classification

Suppose we set the threshold at 0.7, then we have to look at the senses of the relations whose arguments are extracted correctly i.e. correctly enough. So we have to look at (g_1, p_1) and (g_4, p_3). If the senses for (g_1, p_1) are correct, and the senses for (g_4, p_3) are not correct, then the overall parser performance is:

Precision $= \frac{1}{3} = 0.33$. Recall $= \frac{1}{4} = 0.25$. $F_1 = 2 * \frac{0.33*0.25}{0.33+0.25} = 0.28$.