1. Nghiên cứu cách tối ưu hóa chỉ số recall trong mô hình.

Chỉ số recall trên mỗi label được tính bằng (số ý của label đó và được đoán là label đó (LT)/(LT + số ý của label đó nhưng bị bỏ sót(LF).

Ta thấy LT + LF = tổng số ý của label tương ứng.

Từ đó ta có recall =

Ta biết rằng “tổng số” thì khó để thay đổi, vì vậy, tốt nhất là nên giảm LF (tức là số label bị bỏ sót). Để giảm được chỉ số này, thì có nhiều cách.

Trong bài toán phân lớp, ta thường thấy các cách phân lớp cứng, tức là đầu ra bằng là một số a trong khoảng (0, 1) thì nó sẽ xác định nếu a lớn hơn một “ngưỡng” nào đó, a sẽ nhận giá trị 1, nếu không a là 0.

Tuy nhiên thì ta mong muốn là a dễ nhận giá trị 1 hơn, vì ta đâu muốn nó phải “giết nhầm còn hơn bỏ sót”, ta không chắc a là 0 hay là 1, nhưng cứ cho a bằng 1 nếu có khả năng “hơi” cao. Bình thường, ta hay để “ngưỡng” là 0.5, vậy, hãy thử “giảm về 0.4, 0.3 and so on”. Nhưng cách này có vẻ không hiệu quả trong một mô hình đa label.

Ta có một cách ổn định hơn đó là tăng dữ liệu có label tương ứng, dĩ nhiên, nó vẫn không hiệu quả trong mô hình đa label.

Chưa thật sự có cách tối ưu chỉ số recall đối với mô hình đa nhãn.

1. Sử dụng recall để tuning hyperparameter thay vì accuracy.

Thay vì sử dụng accuracy (không phù hợp với comment-moderator), ta nên sử dụng recall trong các thuật toán hyperparameter tuning.