



# AI 挑战项目结题汇报

基于多模态数据的 shopee 商品匹配

# 项目要求

对每一个 post 求出一个 label\_group. 其中特征数据为：

- image
- title

其中每一个 label\_group 对应商品相同.

使用  $f1$  分数评估，即精确率与召回率的调和平均数.

|   | posting_id       | image                                | image_phash      | title                                             | label_group |
|---|------------------|--------------------------------------|------------------|---------------------------------------------------|-------------|
| 0 | train_129225211  | 0000a68812bc7e98c42888dfb1c07da0.jpg | 94974f937d4c2433 | Paper Bag Victoria Secret                         | 249114794   |
| 1 | train_3386243561 | 00039780dfc94d01db8676fe789ecd05.jpg | af3f9460c2838f0f | Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO... | 2937985045  |
| 2 | train_2288590299 | 000a190fd715a2a36faed16e2c65df7.jpg  | b94cb00ed3e50f78 | Maling TTS Canned Pork Luncheon Meat 397 gr       | 2395904891  |
| 3 | train_2406599165 | 00117e4fc239b1b641ff08340b429633.jpg | 8514fc58eafea283 | Daster Batik Lengan pendek - Motif Acak / Camp... | 4093212188  |
| 4 | train_3369186413 | 00136d1cf4edede0203f32f05f660588.jpg | a6f319f924ad708c | Nescafe \xc3\x89clair Latte 220ml                 | 3648931069  |

↑ 样本（用户上传的商品）的id  
↑ 样本的图片  
↑ 样本图片的哈希值  
↑ 样本的标题  
↑ 样本的分类（类别编号）

# 结题设计排名

 shopee2 - Version 2  
Succeeded (after deadline) · 6h ago · Notebook shopee2 | Version 2

0.768      0.781     

目前 Private Score 分数 0.768, Public Score 分数 0.781.

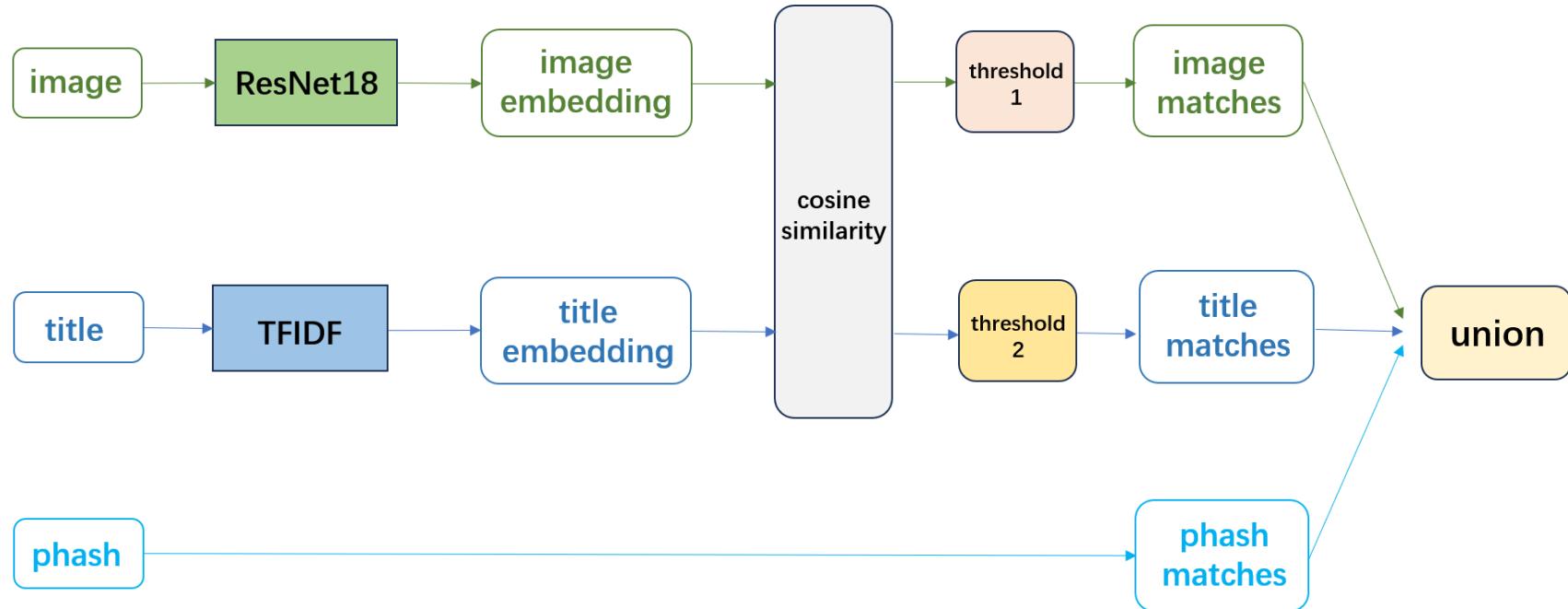
在原比赛排名第5.

| # | △   | Team                          | Members                                                                                                                                                                                                                                                  | Score | Entries | Last |
|---|-----|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|---------|------|
| 1 | —   | Upstage, Making AI Beneficial |                                                                                       | 0.780 | 194     | 2y   |
| 2 | —   | Iyaka & tkm                   |                                                                                       | 0.779 | 251     | 2y   |
| 3 | ▲ 1 | Btbpanda                      |                                                                                                                                                                         | 0.779 | 124     | 2y   |
| 4 | ▼ 1 | Watercooled                   |    | 0.777 | 182     | 2y   |
| 5 | —   | tereka U Ahmet U Alvor        |    | 0.767 | 278     | 2y   |

# baseline回顾

在baseline中， 我们采用 TFIDF 提取文本特征， ResNet18 提取图片特征.

然后获取余弦相似值小于阈值的匹配，将文本得到的匹配和图像得到的匹配通过取并集的方式相接.



# 训练模型

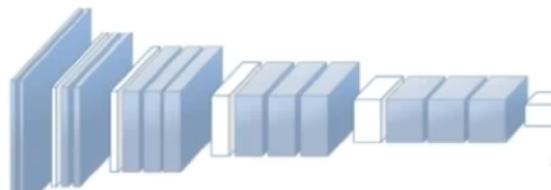
baseline中未使用到给出的训练数据，预训练的参数不能很好地符合本次比赛数据.

我们初期使用 softmax loss 来训练图像和文本模型.

## Softmax-Loss



Data



Network

$$\frac{e^{f_j}}{\sum_j e^{f_j}}$$

Softmax

$$f_i \in \mathbb{R}^d \quad W \in \mathbb{R}^{d \times n}$$

Logit

$$\frac{e^{f_{\eta_i}}}{\sum_j e^{f_j}}$$

Probability

$$\frac{e^{f_{\eta_i}}}{\sum_j e^{f_j}}$$

Ground Truth  
One Hot Vector

$$-\log\left(\frac{e^{f_{\eta_i}}}{\sum_j e^{f_j}}\right)$$

Cross-entropy  
Loss

Loss

# 开放集合目标检测

根据其他参赛者的分析，我们得出此次比赛属于**开集检测**，即测试集的 label\_group 大多没有出现在训练集中。

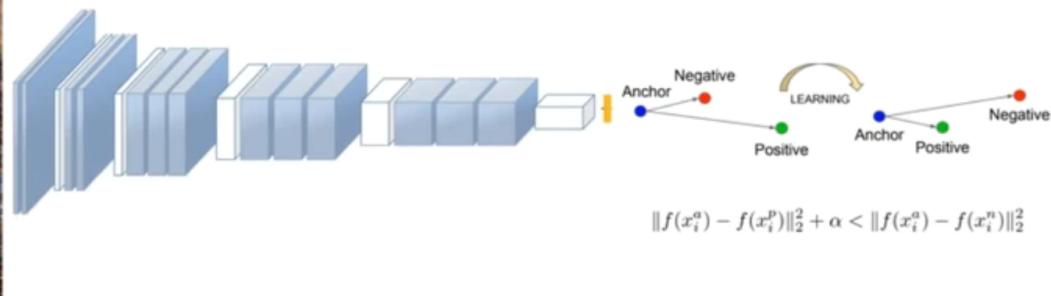
虽然大部分 label\_group 不在训练集中，但可以利用训练集增大不同大类（如衣服，食物，家具）之间的距离。

优点：稳定、表现好

缺点：耗时长

## Triplet-Loss

**Embedding Target:** Intra-class Compactness + Inter-class Discrepancy



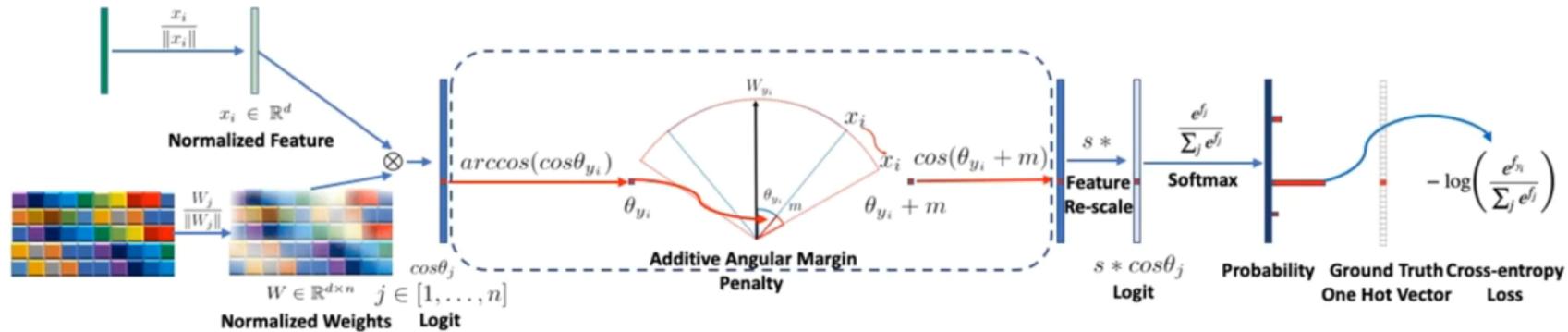
# Arcface

Triplet-loss 针对 sample-to-sample 距离的差异化，但本次比赛训练数据较大。

Arcface 则针对 sample-to-class 距离的差异化，有以下优点：

1. 能够有大量的图像比较（图与图，或者图与类）；
2. 有边界的概念；
3. 系统支持大规模训练

## ArcFace



# 图像处理模型 NFNet

High-Performance Large-Scale Image Recognition Without Normalization

- NFNet = Normalization Free Net.
- Batch Normalization: 防止梯度消失和梯度爆炸以及训练不稳定.但也增大计算成本.
- 自适应梯度裁剪模块: 基于逐单元梯度范数与参数范数的单位比例来裁剪梯度的算法.

1. 梯度裁剪算法: 在更新  $\theta$  之前, 以如下公式裁剪梯度. (梯度向量  $G = \frac{\partial L}{\partial \theta}$ ,  $L$  为损失值,  $\theta$  为模型所有参数向量,  $\lambda$  为需要调整的超参数)

$$G \rightarrow \begin{cases} \lambda \frac{G}{\|G\|} & \text{if } \|G\| > \lambda, \\ G & \text{otherwise.} \end{cases}$$

一个超参数 $\lambda$ 来控制所有层的梯度计算, 需要改进.

# 图像处理模型 NFNet

2. AGC算法：记  $W^l \in \mathbb{R}^{N \times M}$  为第  $l$  层的权重矩阵,  $G^l \in \mathbb{R}^{N \times M}$  为对应于  $W^l$  的梯度矩阵,  $\|\cdot\|_F$  表示 Frobenius 范数, 则有:

$$\|W^\ell\|_F = \sqrt{\sum_{i=1}^N \sum_{j=1}^M (W_{i,j}^\ell)^2}$$

定义第  $l$  层上第  $i$  个单元的梯度矩阵为  $G_i^\ell$  (表示  $G^l$  的第  $i$  行),  $\lambda$  是一个超参数,  
 $\|W_i\|_F^* = \max(\|W_i\|_F, \epsilon)$ ,  $\epsilon$  默认为  $10^{-3}$ , AGC 算法的裁剪公式为:

$$G_i^\ell \rightarrow \begin{cases} \lambda \frac{\|W_i^\ell\|_F^*}{\|G_i^\ell\|_F} G_i^\ell, & \text{if } \frac{\|G_i^\ell\|_F}{\|W_i^\ell\|_F^*} > \lambda, \\ G_i^\ell, & \text{otherwise.} \end{cases}$$

# 文本处理模型 Sentence-BERT

若用 BERT 获取一个句子的向量表示，一般有两种方式：

1. 用句子开头的 CLS 经过 BERT 的向量作为句子的语义信息（这种更常用）.
2. 用句子中的每个 token 经过 BERT 的向量，加和后取平均，作为句子的语义信息.

然而，实验结果显示，在文本相似度任务上，使用上述两种方法得到的效果并不好.

| Model                      | STS12        | STS13        | STS14        | STS15        | STS16        | STSb         | SICK-R       | Avg.         |
|----------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Avg. GloVe embeddings      | 55.14        | 70.66        | 59.73        | 68.25        | 63.66        | 58.02        | 53.76        | 61.32        |
| Avg. BERT embeddings       | 38.78        | 57.98        | 57.98        | 63.15        | 61.06        | 46.35        | 58.40        | 54.81        |
| BERT CLS-vector            | 20.16        | 30.01        | 20.09        | 36.88        | 38.08        | 16.50        | 42.63        | 29.19        |
| InferSent - Glove          | 52.86        | 66.75        | 62.15        | 72.77        | 66.87        | 68.03        | 65.65        | 65.01        |
| Universal Sentence Encoder | 64.49        | 67.80        | 64.61        | 76.83        | 73.18        | 74.92        | <b>76.69</b> | 71.22        |
| SBERT-NLI-base             | 70.97        | 76.53        | 73.19        | 79.09        | 74.30        | 77.03        | 72.91        | 74.89        |
| SBERT-NLI-large            | 72.27        | <b>78.46</b> | <b>74.90</b> | 80.99        | 76.25        | <b>79.23</b> | 73.75        | 76.55        |
| SRoBERTa-NLI-base          | 71.54        | 72.49        | 70.80        | 78.74        | 73.69        | <b>77.77</b> | 74.46        | 74.21        |
| SRoBERTa-NLI-large         | <b>74.53</b> | 77.00        | 73.18        | <b>81.85</b> | <b>76.82</b> | 79.10        | 74.29        | <b>76.68</b> |

Table 1: Spearman rank correlation  $\rho$  between the cosine similarity of sentence representations and the gold labels for various Textual Similarity (STS) tasks. Performance is reported by convention as  $\rho \times 100$ . STS12-STS16: SemEval 2012-2016, STSb: STSbenchmark, SICK-R: SICK relatedness dataset.

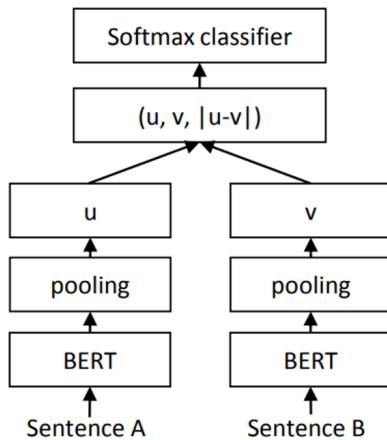
# 文本处理模型 Sentence-BERT

SBERT对预训练的BERT进行修改：

在 BERT 的输出结果上增加了一个 Pooling 操作，从而生成一个固定维度的句子 embedding.

使用 (Siamese) 和三级 (triplet) 网络结构来获得语义上有意义的句子嵌入，以此获得定长的句子嵌入.

对于分类问题有以下流程.

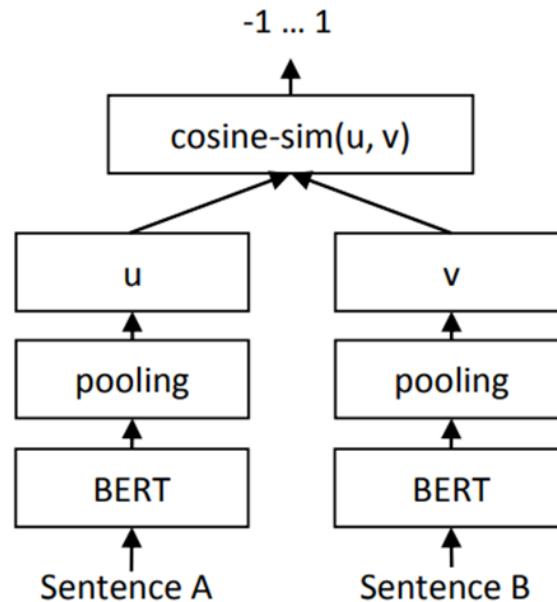


$u, v$  分别表示输入的两个句子的向量表示， $|u - v|$  表示去两个向量按元素相减的绝对值， $W_t$ 是可训练的权重矩阵，输出为：

$$o = \text{softmax}(W_t(u, v, |u - v|))$$

# 文本处理模型 Sentence-BERT

对于回归问题：计算两个句子  $u, v$  embedding 向量的余弦相似度（取值范围在 -1,1 之间）



# 文本处理模型 Sentence-BERT

为什么要将向量  $u, v, |u - v|$  拼接在一起？文章针对不同的 concatenation 方法做了一系列实验，结果显示这种方法效果最好（同时结果也表明：Pooling 策略影响较小，向量组合策略影响较大）

|                          | NLI          | STSb         |
|--------------------------|--------------|--------------|
| <i>Pooling Strategy</i>  |              |              |
| MEAN                     | <b>80.78</b> | <b>87.44</b> |
| MAX                      | 79.07        | 69.92        |
| CLS                      | 79.80        | 86.62        |
| <i>Concatenation</i>     |              |              |
| $(u, v)$                 | 66.04        | -            |
| $( u - v )$              | 69.78        | -            |
| $(u * v)$                | 70.54        | -            |
| $( u - v , u * v)$       | 78.37        | -            |
| $(u, v, u * v)$          | 77.44        | -            |
| $(u, v,  u - v )$        | <b>80.78</b> | -            |
| $(u, v,  u - v , u * v)$ | 80.44        | -            |

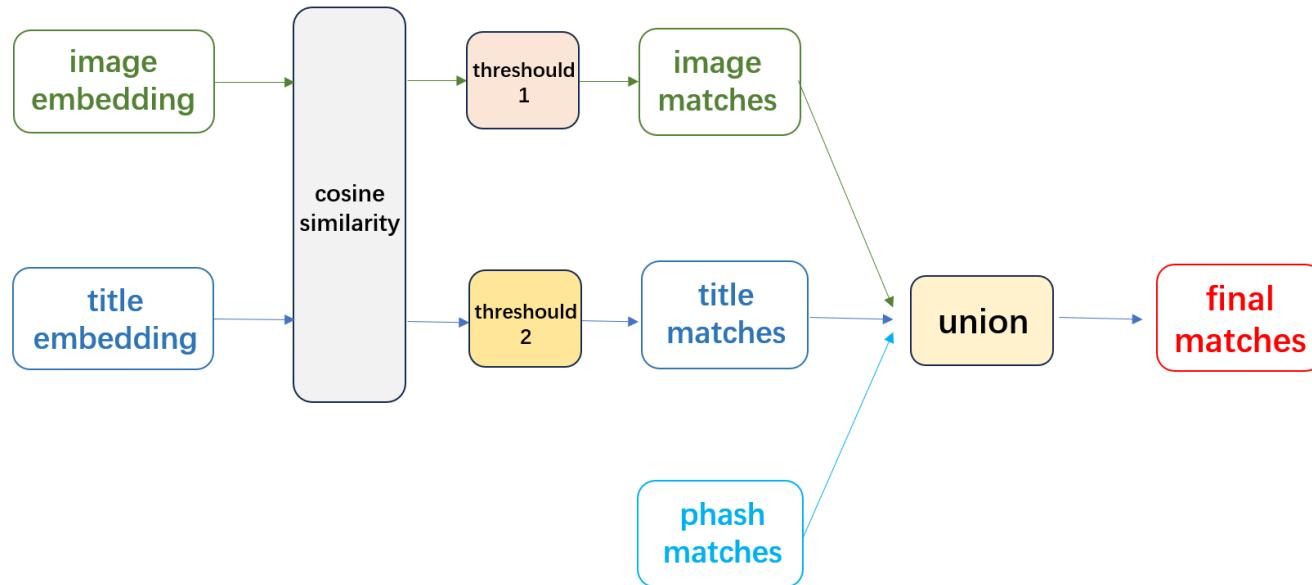
Table 6: SBERT trained on NLI data with the classification objective function, on the STS benchmark (STSb) with the regression objective function. Configurations are evaluated on the development set of the STSb using cosine-similarity and Spearman’s rank correlation. For the concatenation methods, we only report scores with MEAN pooling strategy.

# 融合策略

单模型的局限：同一商品因为拍摄角度或者 title 描述方向的原因可能导致失配.

基于此，我们采取对两个模型得出的匹配结果取并集的融合策略.

对多个模型取并集也是一种正则化的形式（类似集成学习中的 bagging）.



## Min2

根据数据集的特性, 每个 label\_group 的大小至少为 2 至多为 50.

采用 Min2 原则: 我们令最近的点被匹配, 即使其未达到阈值 threshold. 即强制每组大小至少为2.

但如果最近点距离依旧超过min2\_threshold, 说明该数据可能较为特殊, 为保证精确率放弃此次匹配.

```
idx = np.where(distances[k, :] < threshold)[0] # 距离小于阈值为邻居, 包括 k 自己  
ids = indices[k, idx] # 找到 k 的邻居, 包括 k 自己  
  
if len(ids) <= 1 and distances[k, 1] < min2_threshold: # 没有邻居, 找最近一个  
    ids = np.append(ids, indices[k, 1])
```

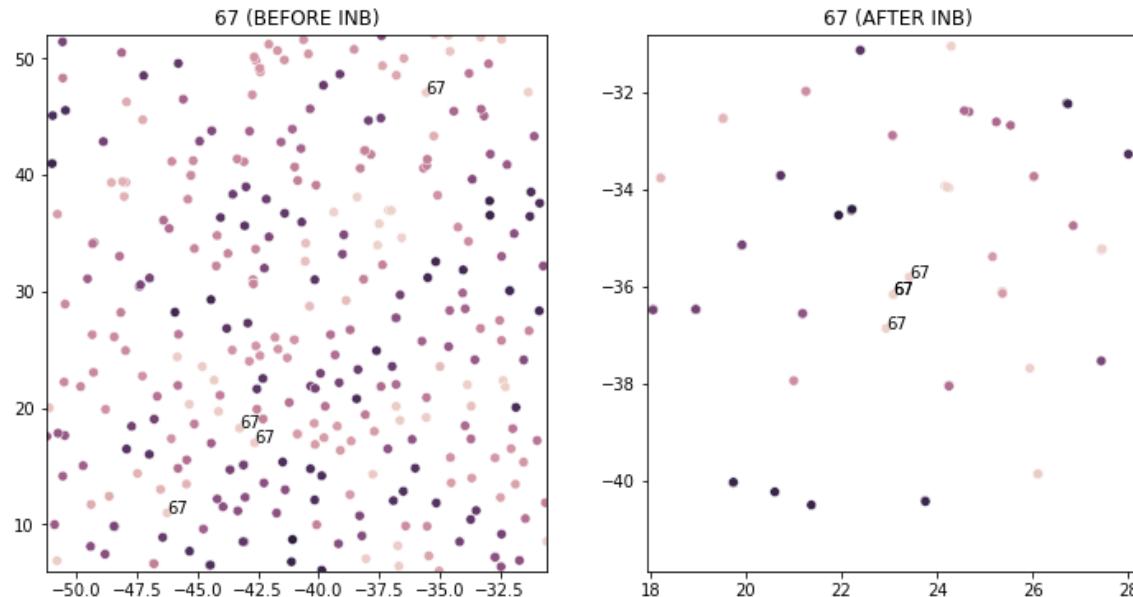
由于大小为2的组较多, 因此min2算法可以较为明显提高 F1 分数.

本次项目 min2\_threshold 取0.6较为合适.

# Neighborhood Blending

使用近邻搜索和Min2阈值化得到图片的(matches, similarities)对，构造图，未通过阈值和Min2条件的节点不连接。

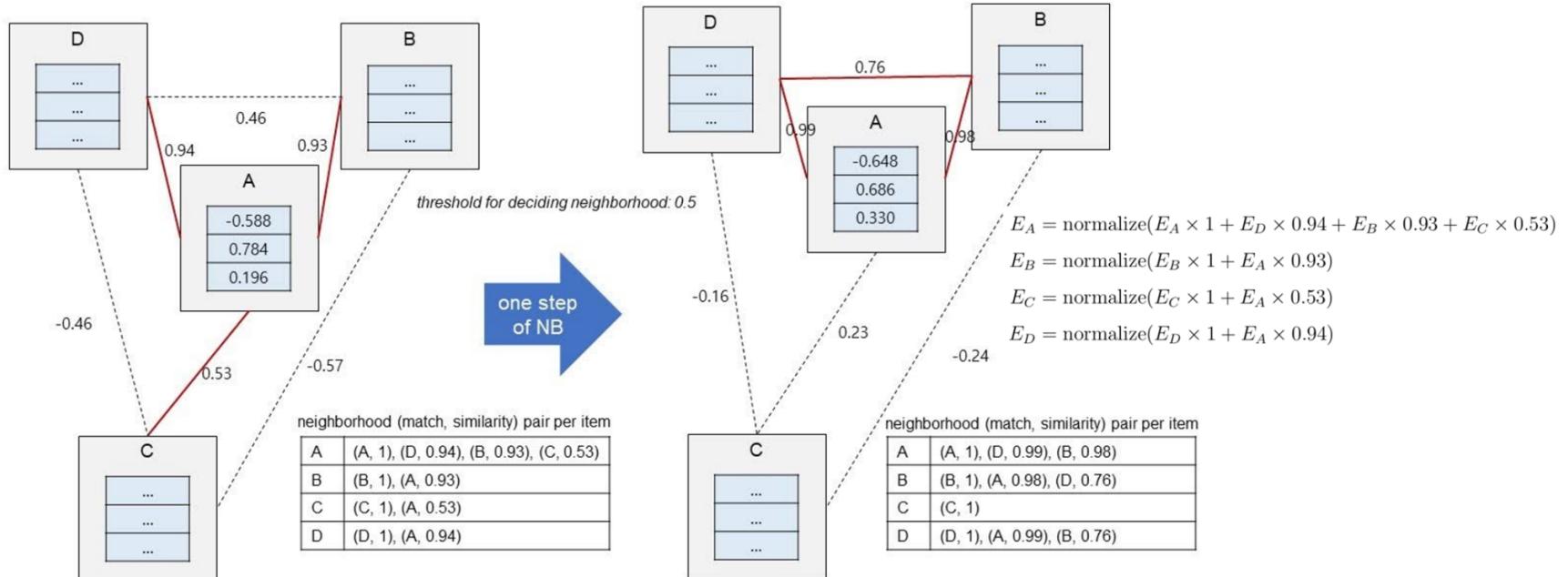
利用邻域信息细化查询图片的嵌入向量，通过相似性加权求取邻域嵌入向量的和，添加到查询嵌入向量中，称为NB (Neighborhood Blending)，该方法可以更有效地聚合同类点。



# Neighborhood Blending

**NB (Neighborhood Blending)**  
based on query expansion / DBA

$$E'_i = [E_{neighbours}] \cdot [Similarity_{neighbours}]^T$$



# 单位提取

在数据集中，我们发现部分数据图像一致，文本相似，但因为**单位**不同而导致失配。

我们通过**正则表达式**提取了所有的weight, length, pieces, memory, volume 单位。

若两个 title 在相同度量制下数值不同，则失配。



|      | title                                | label_group | image_phash      |
|------|--------------------------------------|-------------|------------------|
| 5398 | Kojie San Face Lightening Cream 30g  | 217804619   | faa4bd7b4294a18c |
| 5399 | Kojie San Face Lightening Cream 30gr | 2134514629  | faa4bd7b4294a18c |



|      | title                            | label_group | image_phash      |
|------|----------------------------------|-------------|------------------|
| 9902 | Indonesia Fresh Sari Lemon 250ml | 511311727   | b366ce33cc898ccc |
| 9903 | Indonesia Fresh Sari Lemon 500ml | 387288450   | b366ce33cc898ccc |

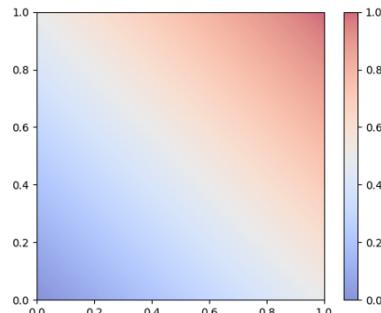
# 融合策略的改进

在融合两个模型的时候，直接从匹配取并集的一个"缺点": 需要先通过 Threshold 得到匹配.

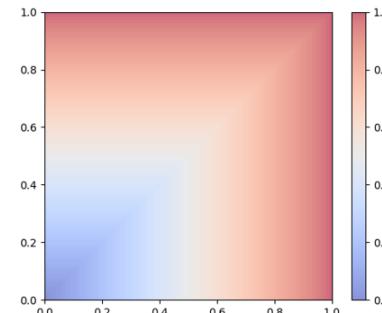
我们考虑先融合余弦相似性，再从相似性直接得到匹配.

令  $a := \text{similarity}_{\text{image}} < x, y >$ ,  $b := \text{similarity}_{\text{text}} < x, y >$

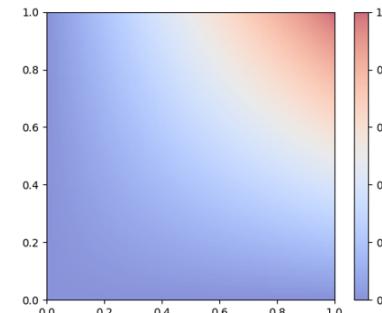
在以下四种方案中，实践证明第四种融合相似度的方法效果最好:



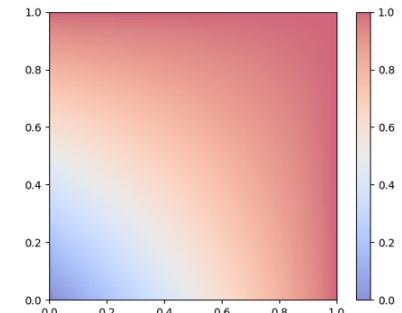
$$\frac{(a+b)}{2}$$



$$\max(a, b)$$

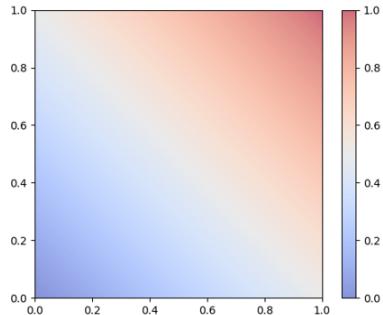


$$a \times b$$

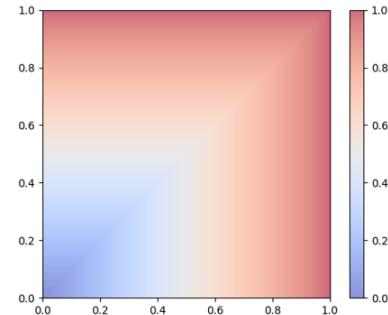


$$a + b - a \times b$$

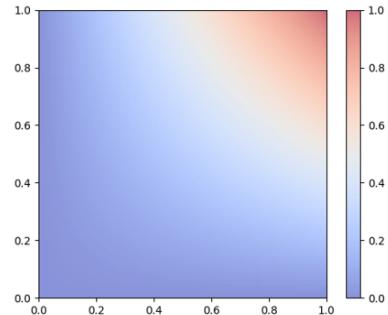
# 概率意义下的并集



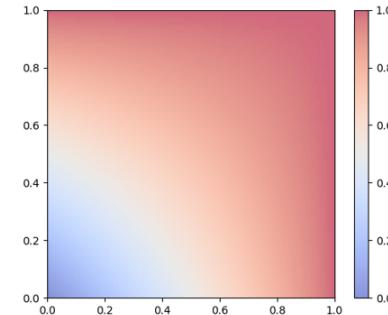
$$\frac{(a+b)}{2}$$



$$\max(a, b)$$



$$a \times b$$

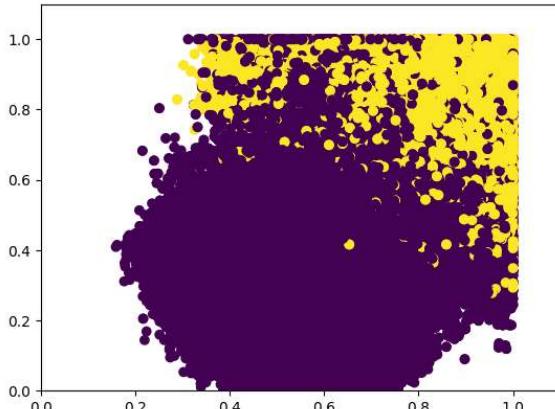


$$a + b - a \times b$$

事实上，余弦相似性可以看作两者匹配的概率：

- $a \in [0, 1], b \in [0, 1]$
- 余弦相似性越大，两者匹配概率越大

而  $a + b - a * b$  即是概率意义下的并集。

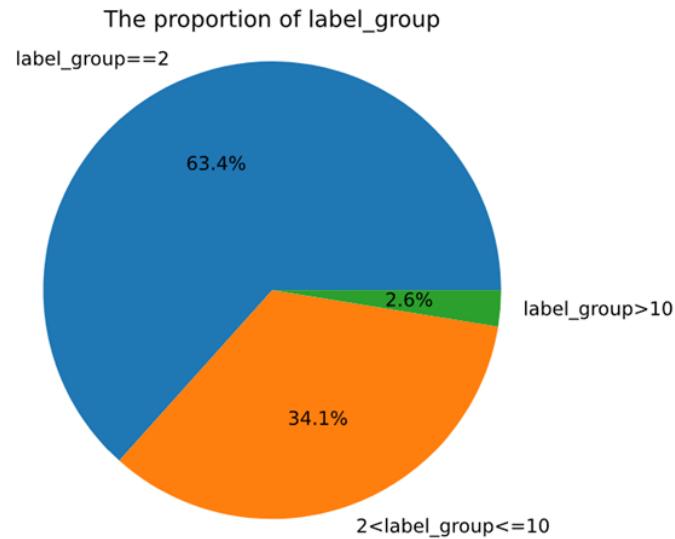


# chisel “凿子”

根据训练集的分布调整阈值 Threshold

我们提出一个假设:

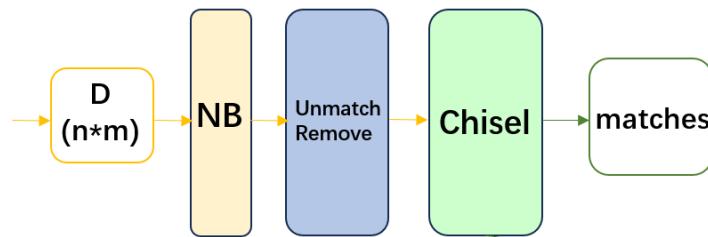
比赛的训练集和测试集的目标 label\_group 长度分布相同



这可能与训练集和测试集划分有关系.

# chisel “凿子”

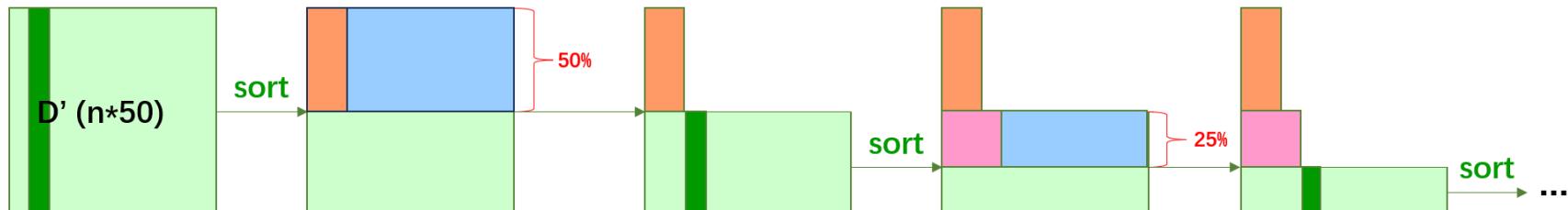
让测试集变成训练集的“形状”



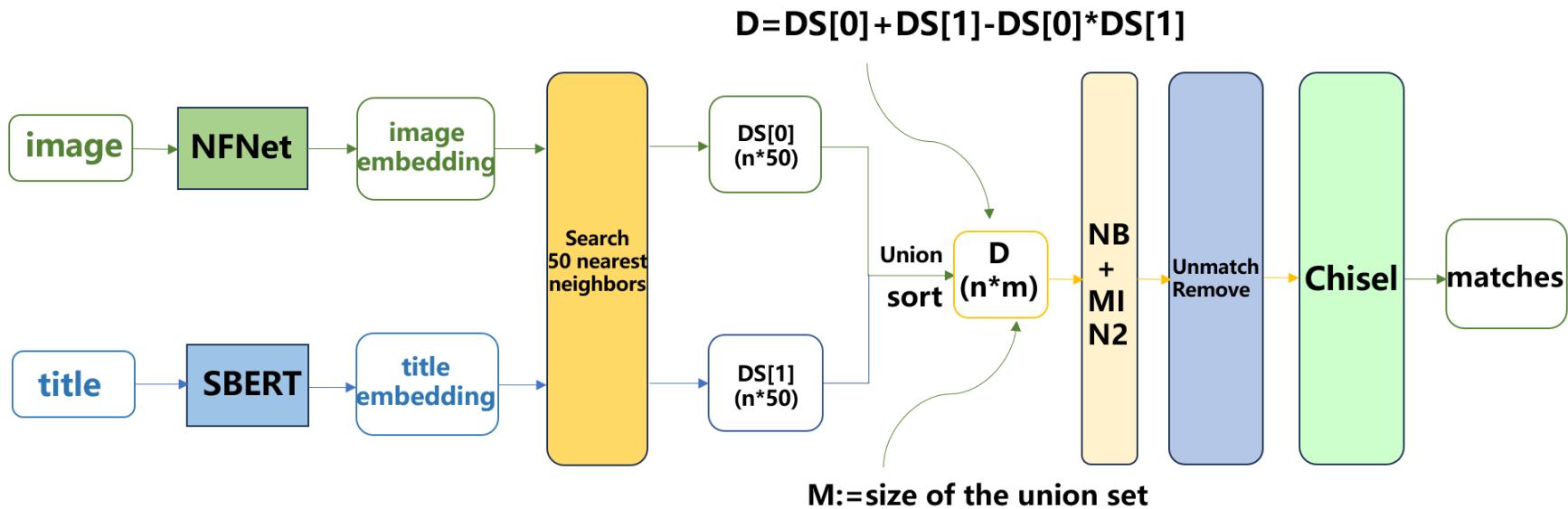
假定训练集train中：  
所含样本数为2的label\_group占总体的50%,  
所含样本数为3的label\_group占总体的25%,  
.....

|   |        |        |        |        |
|---|--------|--------|--------|--------|
| A | A(1.0) | C(0.8) | B(0.5) | D(0.1) |
| B | B(1.0) | A(0.5) | C(0.4) | D(0.2) |
| C | C(1.0) | A(0.8) | B(0.4) | D(0.3) |
| D | D(1.0) | C(0.3) | B(0.2) | A(0.1) |

|   |        |        |        |        |
|---|--------|--------|--------|--------|
| A | A(1.0) | C(0.8) | B(0.5) | D(0.1) |
| B | B(1.0) | A(0.5) |        |        |
| C | C(1.0) | A(0.8) | B(0.4) |        |
| D | D(1.0) | C(0.3) |        |        |



# 结题设计结构



# 结果分析

| models                           | f1       | recall   | precision |
|----------------------------------|----------|----------|-----------|
| resnet50                         | 0.757385 | 0.726135 | 0.925569  |
| resnext50_32x4d                  | 0.760473 | 0.731729 | 0.928756  |
| densenet121                      | 0.762457 | 0.739039 | 0.928803  |
| efficientnet_b3                  | 0.774103 | 0.764059 | 0.912719  |
| eca_nfnet_l0                     | 0.788663 | 0.771427 | 0.911394  |
| bert-base-multilingual-uncased   | 0.817695 | 0.848685 | 0.884845  |
| bert-base-indonesian-1.5G        | 0.812737 | 0.843387 | 0.868974  |
| distilbert-base-indonesian       | 0.808627 | 0.845347 | 0.861303  |
| paraphrase-xlm-r-multilingual-v1 | 0.821114 | 0.848321 | 0.882831  |
| paraphrase-distilroberta-base-v1 | 0.802462 | 0.834882 | 0.872976  |

- 本次比赛使用 eca-NFNet 以及 sentence-transformers/paraphrase-xlm-r-multilingual-v1.
- 本次比赛文本特征重要性略高于图像模型.

# 总结体会

- 模型的提升和调参并没有带来太大的提升.

Resnet18+Tfidf → NfNet-l1+SBERT(with Arcface), 0.762 → 0.768

- 后处理在本次比赛较为重要.

使用 chisel 后, 0.739 → 0.759

使用 Neighborhood Blendingneig 后, 由银牌区 → 前10名

- 使用决策树也是另一种优秀的融合策略.

前几名部分的开源方案用了 LightGBM, XGBoost 等方法替代并集融合.

- 数据集较为嘈杂是限制 f1 分数的一个重要因素.

"...noisy data is expected as that is the real situation in E-commerce since sellers have their own method of marketing, naming and promoting their products."

| posting_id |                  | image                                | image_phash      |                                                   | title | label_group |
|------------|------------------|--------------------------------------|------------------|---------------------------------------------------|-------|-------------|
| 10412      | train_880338666  | 4e0bad1f97ab9bb4a196511d9c65629c.jpg | d5780e316ed58786 | 100Pcs Ikat Karet Rambut Elastis Warna Polos G... |       | 994676122   |
| 22505      | train_2232408411 | a8f94251963c95d2b687fadf4ed9c99a.jpg | bb9ac06684987b6b | 100Pcs Ikat Karet Rambut Elastis Warna Polos G... |       | 2045845868  |
| 32857      | train_1157582002 | f6053a4f010a44f006e168a46e78332c.jpg | f7b9c25864433966 | 100Pcs Ikat Karet Rambut Elastis Warna Polos G... |       | 994676122   |

# 致谢

- 感谢文泉老师细心指导
- 感谢 Kaggle 比赛讨论区各位大神 以及 github 用户jingxuanyang 的比赛总结
- 感谢 ChatGPT, Github Copilot 等AI工具的帮助



感谢聆听

2023.11 第五小组