机器学习工程师

毕业项目开题报告

项目：Jigsaw恶毒评论分类
Toxic Comment Classification Challenge

报告人：李铮琦

问题描述

Problem Statement

Jigsaw在kaggle平台上举办了一场文本分类比赛，旨在对于网络社区部分恶毒评论进行区分鉴别。在该赛题中，需要建立一个可以区分不同类型的言语攻击行为的模型，该赛题一共提供了以下六种分类标签，需要根据提供的训练数据进行模型训练学习：

- toxic（辱骂的）
- severe toxic（严重辱骂的）
- obscene（淫秽的）
- threat（威胁的）
- insult（侮辱的）
- identity hate（身份仇恨的）

模型需要针对测试集中的每一条评论预测其所有标签的概率值。

项目背景
Domain Background

本项目属于自然语言处理（Natural Language Processing，NLP）中的文本分类问题。自然语言处理是人工智能和语言学领域的分支学科，此领域探讨如何处理及运用自然语言，意在让计算机把输入的语言变成有意义的符号和关系，然后根据目的再处理。[1]文本分类（text categorization）是自然语言处理中一类比较常见且重要的问题，通过分析文本的内容，依据属性（如主题、感情色彩等）对文本进行归类，目前在邮件过滤、信息检索、情感分析等领域有广泛的应用。[2]实现智能（准确且快速）地文本分类具有巨大的商业前景，它使得将海量的文本信息归类、做出理解判断，例如挖掘一款产品的用户评论，能够指导开发者改进产品的用户体验[3]。

在二十世纪八十年代以前，基于“转换-生成文法”的方法过于复杂，且并不能得到理想的效果；随着语料库建设和语料库语言学的崛起，大规模真实文本结合机器学习的方法逐渐走入人们的视野。[4]前期的机器学习方法（例如决策树）仍不能摆脱人为规则的方式，无法处理语料库中未曾见过的文本，性能未有提升；其后随着“隐形马尔可夫模型”的引入，以概率论为依托，结合机器学习中的神经网络结构，以及计算性能的飞跃（使用GPU、TPU进行运算），实现了计算速度和准确率上的长足进步，机器学习方法才得以成为自然语言处理的主流选择。

2013年，Tomas Mikolov 团队相继发表了《Distributed Representations of Words and Phrases and their Compositionally》、《Efficient Estimation of

---

[1] https://zh.wikipedia.org/wiki/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80%E5%A4%84%E7%90%86

[2] https://zh.wikipedia.org/wiki/%E6%96%87%E6%A1%A3%E5%88%86%E7%B1%BB

[3] https://www.jianshu.com/p/12de34cec389

[4] https://zh.wikipedia.org/wiki/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80%E5%A4%84%E7%90%86

Word Representations in Vector Space》2篇论文，创新性地提出Word2vec方法[5]，相比此前的“one-hot representation”编码方式，增强了词与词之间的相关性，并有效降低了词向量的空间维度[6]，打开了自然语言处理在文本分类上的新局面。随后，基于Word2vec方法，出现了一系列改进方法，例如增强词汇“共现性”的GloVe[7]方法，以及利用分层Softmax实现快速计算的fastText[8]方法，有了更加简洁精准的词向量模型，结合CNN、RNN等神经网络结构，实现了更加高效准确的文档识别[9]，形成了有别于传统词袋模型的“词向量+神经网络”模型，也是现阶段主流的NLP解决方案之一。

从宏观上看，自然语言处理的终极目标，是人与机器之间无障碍的交流，因此，文本分类作为其中一个子功能，不断完善此功能的实现也是现阶段人工智能迈向未来强人工智能的必由之路。通过此项目的契机，我想在自然语言处理方面做一些了解，并实现自己在机器学习领域探索中，继图像识别后的一个新突破。

---

[5] https://zhuanlan.zhihu.com/p/26306795

[6] https://www.cnblogs.com/iloveai/p/cs224d-lecture2-note.html

[7] https://www.leiphone.com/news/201801/QsXLJ2uM7cwgijMz.html

[8] https://cloud.tencent.com/developer/article/1080923

[9] https://zhuanlan.zhihu.com/p/40276005

数据与输入

Datasets and Inputs

数据获取：

本项目所使用的数据集下载链接：
https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

数据说明：

- train.csv - 训练集，包含评论及其二进制标签
- test.csv - 测试集，需要对这些评论进行恶毒性预测，为防止人工贴标，测试集中包含一些非用于评分的评论
- sample_submission.csv - 提交文档的正确格式
- test_labels.csv - 测试集标签，-1值表示非用于评分

数据使用：

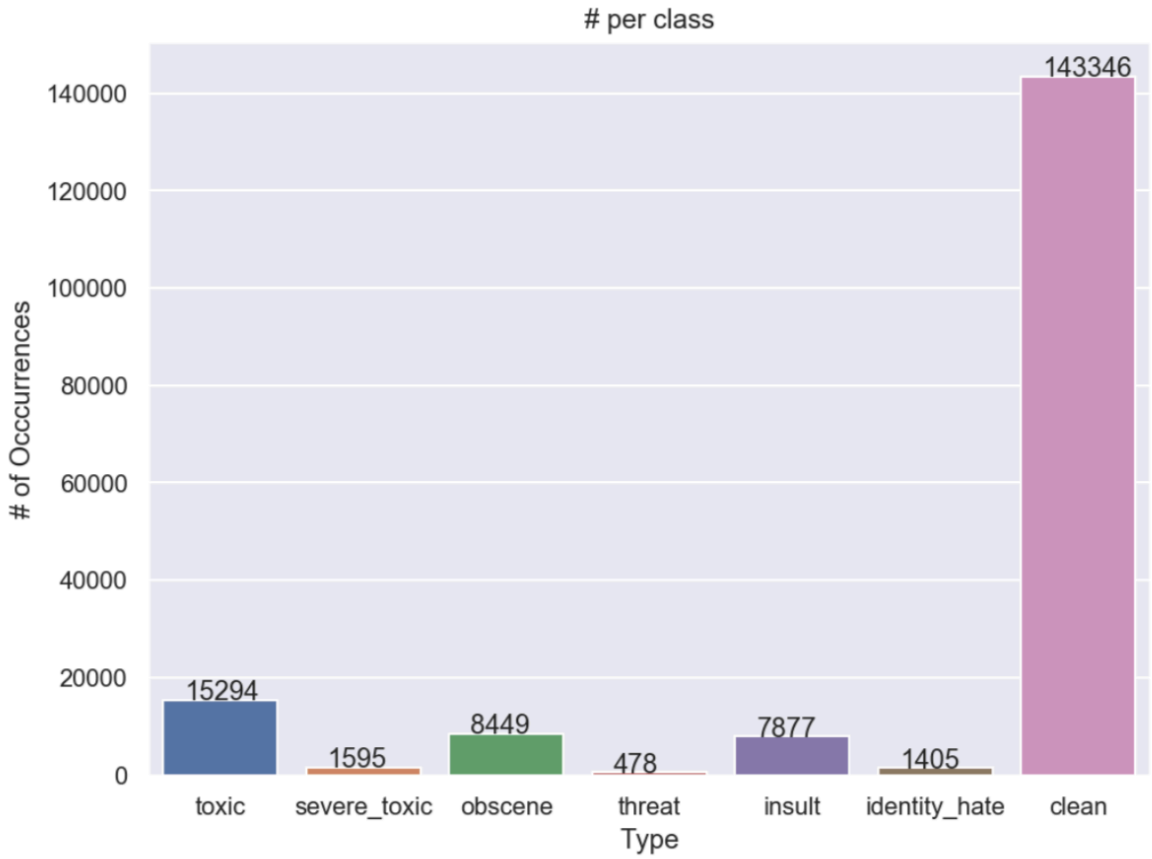
下载完成后，在JupyterNotebook中，使用pandas.read_csv('xxx.csv') 即可获得相应数据；需要通过在train.csv数据集上进行验证集划分、建模，在test.csv数据集上进行测试。

数据特点：

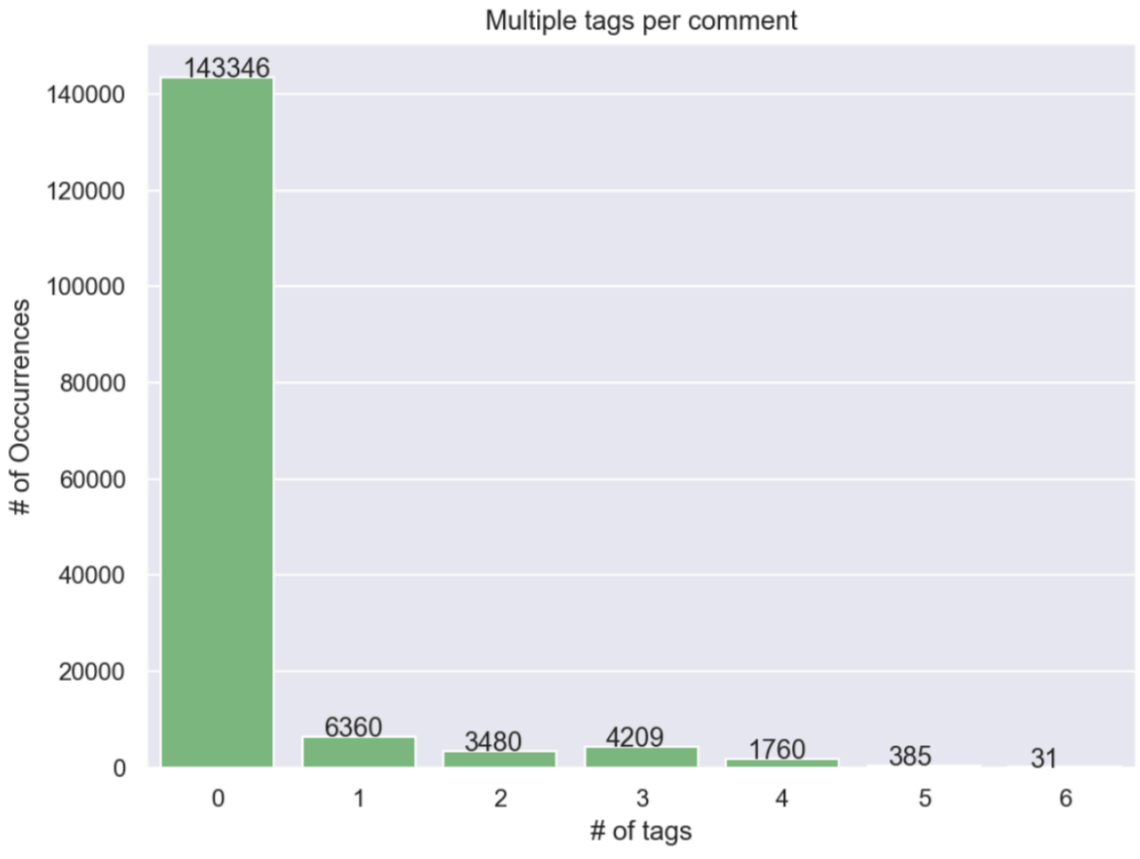
非规范性：评论中存在一些如'\n'的转义字符和大小写

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

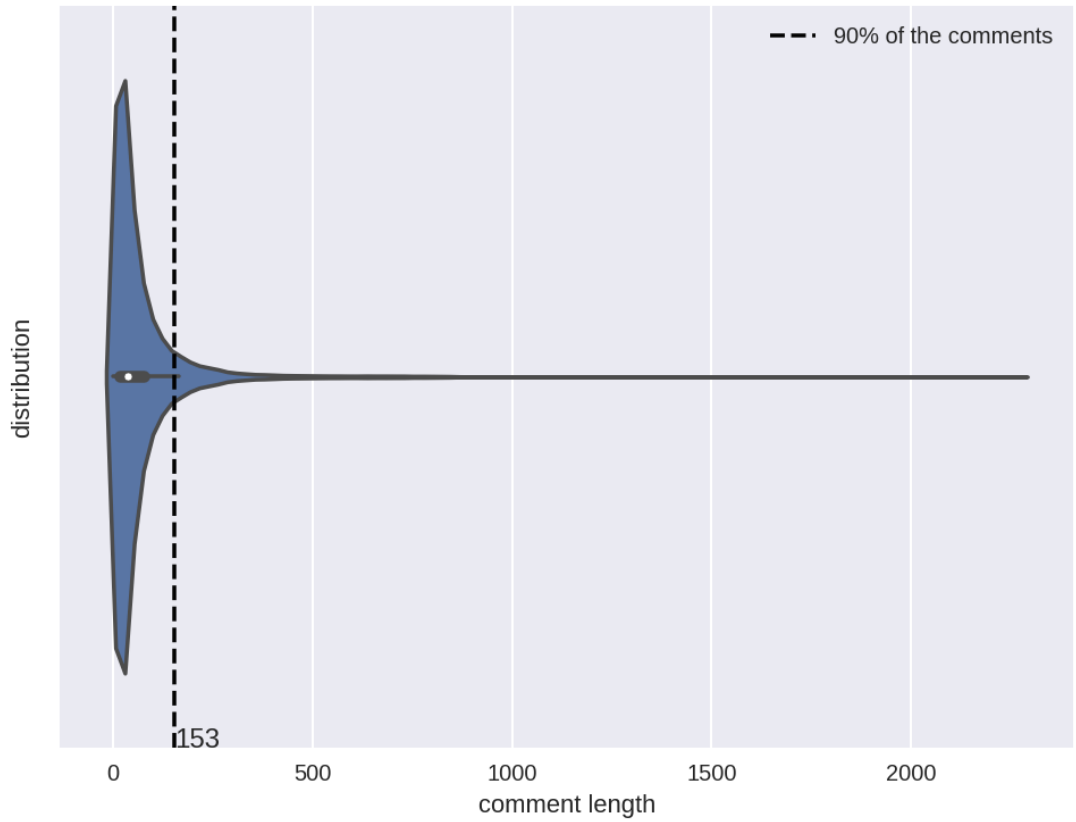
非平衡数据集：训练集159571条评论中，123346条（将近90%）均是“非攻击性”评论



多标签：在“攻击性”评论中，存在一条评论从属于多个标签的现象



评论长度倾斜：90%的评论长度都小于153字



根据以上数据特点，使用训练集数据时，要对其进行规范化，比如过滤掉评论中的转义字符、限制每条评论的长度（加快计算效率），必要时可进行数据重采样，将“非攻击性”评论的数量降低至“攻击性”评论数量的水平。

评估标准

Evaluation Metrics

根据训练集数据特征，此问题是多标签分类问题，对于每个标签，分类器进行的就是一个二元分类：该评论是否属于该标签。对于二元分类模型，可使用交叉熵损失函数（binary crossentropy）作为目标函数，表达式如下：

$$c = min\{-\frac{1}{N}\sum_{i=1}^{N}\left[y_i \ln p_i + (1-y_i)\ln(1-p_i)\right]\}$$

其中：

y_i表示样本是否属于该类，是为1，否为0

p_i表示样本预测为该类的概率

N为样本总数

因此，衡量整体分类性能的总目标函数如下所示：

$$C = \frac{1}{m}\sum_{i=1}^{m} c_i$$

m为标签总数，即总目标函数是所有分类交叉熵的均值。

在训练的过程中，可观测每个周期的训练损失、训练准确率、验证损失、验证准确率，损失即C，准确率如下所示：

$$accuracy = \frac{\sum_{i=1}^{N} I(y_i = \hat{y}_i)}{N}$$

其中：

y_i为样本真实标签

$\hat{y}_i$为预测标签

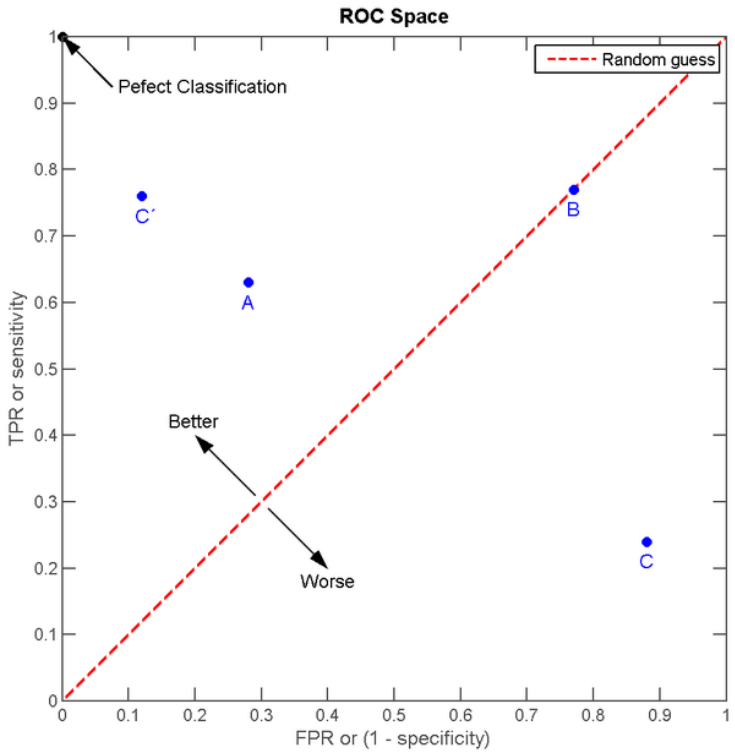
当y_i与$\hat{y}_i$两者相等时，I=1，否则I=0

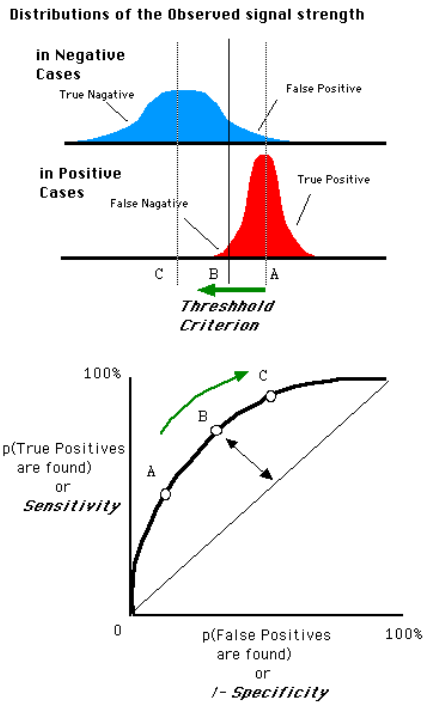
对于观测每一类的分类性能，可使用ROC曲线（Receiver Operating Characteristic curve）作为评估标准。分类器性能越好，其曲线就约靠近（1, 1）点，曲线下方的面积就越大，换言之，ROC曲线下方的面积AUC（Area under the Curve）可以衡量分类器在一种标签上的分类表现，面积越大，分类性能越好。取所有分类AUC的均值，可以作为整体的分类评价指标。

		真实值		总数
		p	*n*	
预测输出	*p′*	真阳性 (TP)	伪阳性 (FP)	P'
	n′	伪阴性 (FN)	真阴性 (TN)	N'

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$





此外，模型的训练时长也可作为评估标准之一，好的模型在训练时长上不应该过大。

基准模型

Benchmark Model

根据项目要求，所实现的模型在线上得分需要达到kaggle private leaderboard的top20%。本项目一共有4501名参赛者（团队），达到前20%的分数至少要到达第910名的成绩，即0.9862，考虑排名并列的情况，实际需达到的成绩为0.9863（排名第900，top19.8%）。

private leaderboard 基于测试集75%的数据，根据实际提交的结果来看，对于同一个输出数据的重复提交，分数是固定的，说明kaggle使用测试集中固定的一部分作为评分基准。

项目设计
Project Design

本项目的解决方案之一：Word Embedding+CNN

实施流程：

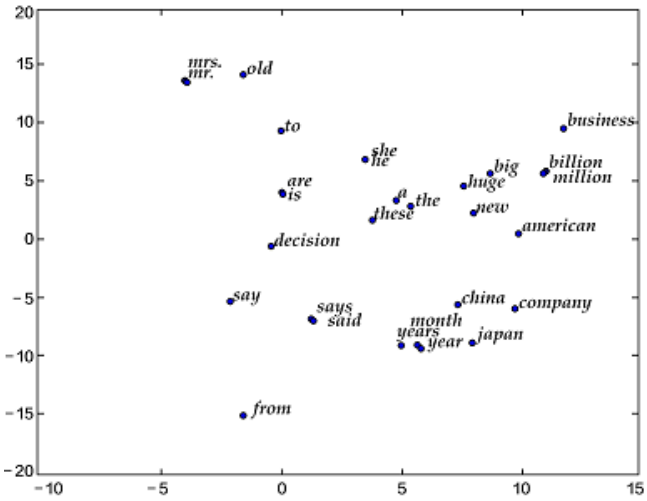
- **数据清洗**
  将全部数据集中的每条评论进行清洗，统一大小写，去除标点、数字、转义字符及对评论不产生影响的停用词（介词、代词等）
  例如：
  清洗前的评论：
  `"Explanation\nWhy the edits made under my username Hardcore Metallica Fan were reverted? They weren't vandalisms, just closure on some GAs after I voted at New York Dolls FAC. And please don't remove the template from the talk page since I'm retired now. 89.205.38.27”`
  清洗后的评论
  `'explanation edits made username hardcore metallica fan reverted vandalisms closure gas voted new york dolls fac please remove template talk page since retired'`

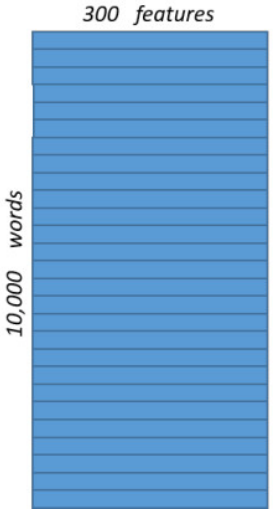
- **词典映射**
  将过滤之后的评论离散为单个词汇，每个单词对应唯一一个整数，组成一个词典，即形如 {单词：词序号} 的字典，每个词的序号顺序是基于词频的，高频词汇的序号越小。
  例如：

```
{'article': 1,
 'wikipedia': 2,
 'page': 3,
 'would': 4,
 'one': 5,
 'talk': 6,
 'like': 7,
```

```
    'please': 8,
    'see': 9,
    'think': 10
  ……}
```

- **词向量映射**
  词向量的物理含义：将一个单词用有限维度的向量映射在高维空间，如下如所示（仅以二维空间示例）：



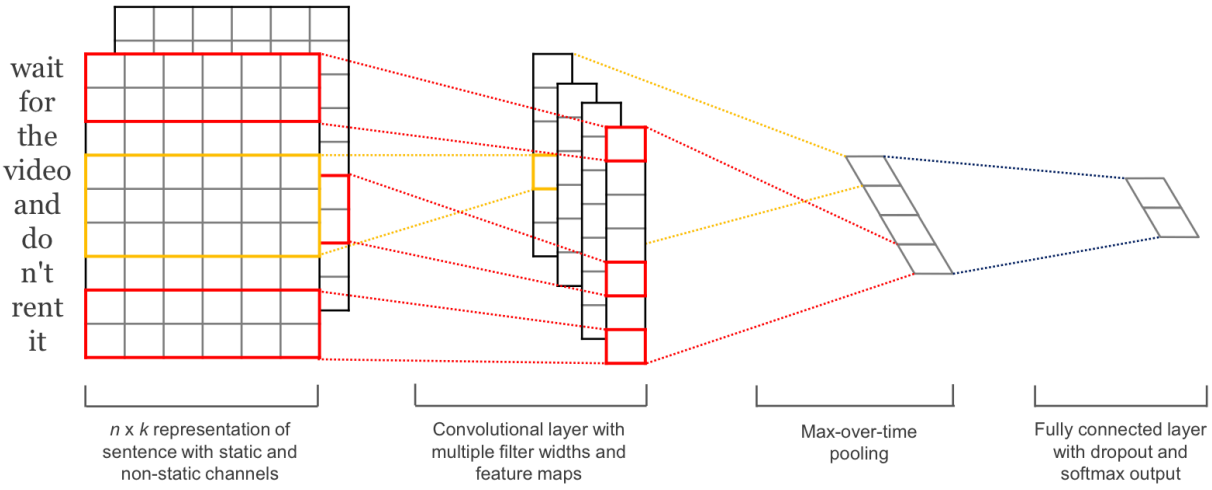


  使用预先训练的词向量，将字典中的每一个单词用一个词向量替代，形成特征矩阵。
  例如：
  假设规定词典词汇量的上限是10000，每个单词的特征向量为300维，那么特征矩阵的维度就是（10000，300），这个特征向量的意义是将数据集中前10000个高频词汇映射为10000个300维的向量，矩阵中的行数对应一个词的词典索引数（如右上图所示）。

- **输入处理**
  CNN网络的输入是每个评论单词组成的二维特征矩阵，每条评论长短不一，此时需要设置一个适当的长度值maxlen（例如为10），将评论长度统一。
  例如：
  创建一个（10，300）的矩阵，当某条评论单词数量小于maxlen（例如为5）时，找出句中5个单词对应的300维特征向量，放入矩阵的前5行，随后用零填充剩余的5行；同理可知，当评论单词数量大于maxlen

时，会截取前10个单词，找出对应特征向量，再进行填充。
总之，每条评论的输入矩阵必然是（10，300）维度的，评论越短，矩阵越稀疏。

- **搭建 CNN 网络**



输入层：特征矩　卷积层：提取特征　池化层：特征降维　全连接层：映射特征

- **训练模型**
在训练过程中，需要将训练集划分一部分为验证集，返回每轮的指标（损失、准确率），观测训练效果，根据此调整相关的超参数，或是优化网络结构，使模型达到较好的结果。理想情况下，每个训练周期的损失下降，准确率上升，为了防止模型过拟合，可使用随机丢弃（Dropout）或设置早期停止（Earlystopping）。

- **提交结果**
训练完毕后，使用测试集生成预测概率的csv格式文件，提交至竞赛平台查看得分。