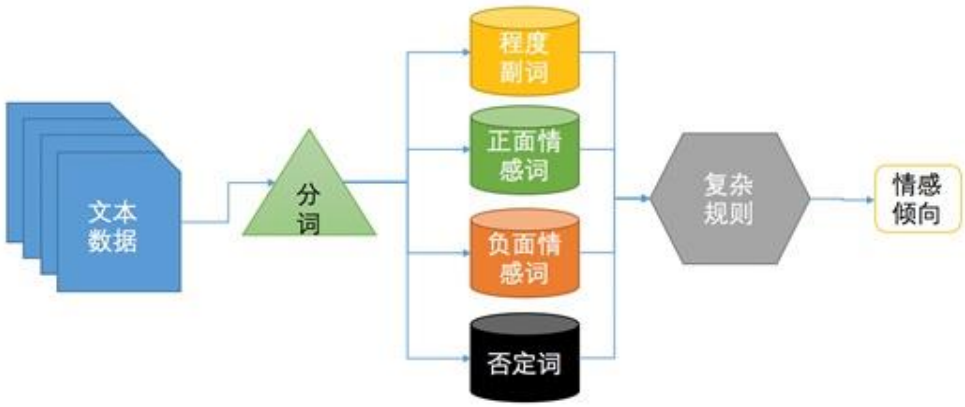
◆ NLP 传统算法

1. **分词——句子切分词语 text classifier**
   分词算法：
   -基于**词典**分词算法
   -基于统计的机器学习算法 **HMM(隐马)、CRF(随机场)**
   -机器学习算法和词典相结合，一方面能够提高分词准确率，另一方面能够改领域适应性??
   中文分词 CRF：CRF(conditional random field)——目前业界较好，较为成熟工具 **CRF++**，可以直接安装
   4-tags 标记
   https://blog.csdn.net/liu_zhlai/article/details/52335527
   结巴中文分词：
   https://github.com/search?q=%E5%88%86%E8%AF%8D&type=Repositories
   HanLP：https://github.com/hankcs/HanLP
   字嵌入+**Bi-LSTM+CRF** 分词器：本质上是序列标注，这个分词器用人民日报的 80 万语料，据说按照字符正确率评估标准能达到 **97.5%的准确率**
2. **设置情感词典**
   情感词典一般包括 5 个词典，即正面情感词典、负面情感词典、否定词典、程度副词词典和行业情感词典
   情感确定，情感逆转，情感加强从而最终影响情感倾向，行业情感，不同行业有特定的情感词，或者说属性情感
3. **基于规则匹配**
   规则匹配扩展性非常差，已经淘汰

已经有现成的框架可供调用，包括分词、实体识别、情感分析等众多功能，比如 pip install 一个库，就搞定了
SnowNLP：https://github.com/isnowfy/snownlp



◆ 基于机器学习算法

基于 Logistic Regression、SVM、随机森林等经典算法

基于规则的方法，机器学习算法多了 2 个步骤：**特征提取和模型训练**

特征提取:

词袋模型（bag of words）：即将文本转换为基于词语的一个向量，向量的每一维度是一个词语，词语可以基于分词得到，也可以基于 N-Gram 模型得到。每一维度的特征取值也可以有多种计算法方式，比如经典的 one-hot 编码和 tf-idf（提取关键词）值。

词袋模型：https://blog.csdn.net/android_ruben/article/details/78238483

N-Gram：https://blog.csdn.net/taoqick/article/details/70755688

模型训练:

基于文本的标注类别和提取好的特征向量，即可以**使用机器学习算法**进行训练，模型训练完成之后即可用于判别文本的情感倾向

这种效果主要取决于特征工程

即**提取的特征是否能足够很好的区别正面和负面情感**

做好特征工程，非常依赖于**人的先验知识**，即需要我们对数据进行足够深入的观察和分析，把那些对**区分正负面情感最有用的 特征 一个一个找出来**。

费时费力

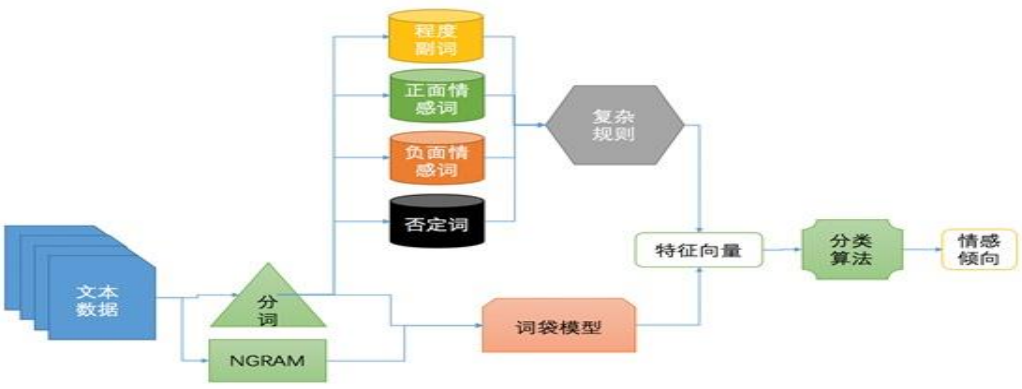
目前进行情感分析，更加精确的方法，是利用机器学习的方法，将“情感分析”转化为一个文本分类问题（比如：正、负、中性），常见的分类算法都可以被使用，如果再 ensemble 一下，效果应该更好。而最重要的问题，就是如何从文本中提取特征，常见的文本特征提取方式包括：

Bag of words, TF-IDF

为了处理否定词，以及处理常见短语，可以提取 N-gram 作为特征，N 是一个超参数

而“常规”机器学习的困难之处就在于，需要人**工从文本提取特征，且工作量大**，比如 N-gram 时的 N 如何选取。另外，BOW/TF-IDF，每个词都是以 OHE 向量表示，高纬度高稀疏的，缺乏语义，特征表达能力很弱。

词袋模型问题：词汇表的构建 向量稀疏问题 数据量的问题 无序性的问题



◆ 深度学习(CNN RNN LSTM)

1）**无需特征工程**：深度学习可以**自动**从数据中学习出特征和模型参数，省去了大量繁杂的特征工程工作，对行业先验知识的依赖也降低到最小程度。

2）**考虑语义上下文**：深度学习在处理文本数据的时候，往往是先把词语转成词向量再进行计算，**词向量的生成考虑了一个词语的语义上下文信息**，也就解决了词袋模型的局限性。

3）**大幅减少输入特征维度**：由于使用了**词向量**，特征维度大幅减少，可以降低到百的量级，同时也使得文本向量变得“稠密”，模型变得更加稳定。

深度学习主要包含两个步骤

1. **词语转成词向量**

   google 的 word2vec 算法是目前应用最广泛的词向量生成算法，实践证明其效果是非常可靠的，尤其是在衡量两个词语的相似度方面。Word2vec 算法包含了 CBOW（Continuous Bag-of-Word）模型和 Skip-gram（Continuous Skip-gram）模型。简单而言，CBOW 模型的作用是已知当前词 Wt 的上下文环境（Wt-2，Wt-1，Wt+1，Wt+2）来预测当前词，Skip-gram 模型的作用是根据当前词 Wt 来预测上下文（Wt-2，Wt-1，Wt+1，Wt+2）。因此，一次词向量事实上是基于词语的上下文来生成的，也就具备了词袋模型所不具备的表意能力，词向量‘稠密’

   word2vec：https://zhuanlan.zhihu.com/p/26306795

2. **利用深度学习框架进行训练**

词转成固定维度的词向量之后，一个文本也就自然而然可以形成一个矩阵，如图 6 所示。以矩阵作为输入的深度学习算法，第一个想到的自然是在图像识别领域获得过成功的卷积神经网络（CNN）。但 CNN 在文本挖掘领域的运用具有一定局限性，因其每层内部的节点之间是没有连接的，即又丢失了词与词之间的联系。前面已经多次强调，词语的上下文关系对文本挖掘是至关重要的，尤其对情感分析，情感词（“喜欢”）和否定词（“不”）、程度词（“很”）的搭配会对情感倾向产生根本性的影响。因此目前比较广泛使用的是 **LSTM（Long Short-Term Memory，长短时记忆），LSTM 能够“记住”较长距离范围内的上下文对当前节点的影响**

RNN CNN LSTM

CNN LSTM 更快 可以结合 RNN，可能提升不大

https://www.zhihu.com/question/60688178

使用 pre-train 的 word embedding，并将 embedding 层也作为变量，在学习过程中加以 update，往往就可以取得不错的效果，可以作为 baseline model，成为未来进一步提升的基础

最重要的还是要认真研究你的数据，提炼出有意义的特征

instance:

https://blog.csdn.net/diye2008/article/details/53105652?locationNum=11&fps=1

代码推荐：https://zhuanlan.zhihu.com/p/22154606

文章和源码: https://blog.csdn.net/chenzhi1992/article/details/70157761

CNN+NLP ：
https://zhuanlan.zhihu.com/p/30268946?utm_source=wechat_timeline&utm_medium=social&from=timeline

国内 NLP：达观数据 bosonNLP

其它问题：

情感分析的难题：

-反讽问题

-情感标签

-时下网络流行语、新词

-短文本，省略严重，需要结合具体情景分析

Plus：迁移学习(Transfer Learning)——部分解决领域迁移的问题

其核心思想就是，让学习到的特征比较通用，而不去过分迎合某个领域
https://github.com/TURuibo/Transfer-Learning-Sentiment-Classification