

Vol. 34 No. 05: AAAI-20 Technical Tracks 5 / AAAI Technical Track: Natural Language Processing

特征工程 联邦学习 注意力机制 生成一个东西-情感分类、文本分类等等。

网络有效性攻击

新闻文本知识发现系统：有假新闻/关税检测器，现在需要基于注意力机制生成一个Yes、No的问题，用户可以提出问题，我们基于文本给出问题的答案。（可以参考 选择题判别器 概念提取）

更广义的，有任意的知识库，从这之中提取到对应的知识，生成模型。

其实是一种文本蕴含问题。

机器学习算法通常容易受到敌对示例的攻击，这些示例与原始示例相比有着不可察觉的变化，但可以愚弄最先进的模型。通过公开恶意制作的对抗性示例，有助于评估甚至提高这些模型的健壮性。在本文中，我们介绍了textfooler，一个简单但强大的生成对抗性文本的基线。通过将其应用于两个基本的自然语言任务，文本分类和文本裁剪，我们成功地处理了三个目标模型，包括强大的预训练Bert和广泛使用的卷积和递归神经网络。我们展示了该框架的三个优点：（1）有效的它在成功率和扰动率方面优于以前的攻击，（2）保持效用的它保留了语义内容、语法性和人类分类的正确类型，以及（3）它能高效地生成对抗性文本，计算复杂度与文本长度成线性关系。

尽管现有的对抗性示例研究在图像和语音领域取得了成功（Szegedy等人，2013年；Carlini和Wagner，2018年），但由于文本数据的离散性，这一工作仍具有挑战性，因此需要对其进行处理。

前人的工作：以前的工作几乎不符合以下的三个要求。

例如，拼写错误（Li等人2018；Gao等人2018）、单字擦除（Li、Monroe和Jurafsky 2016）以及短语插入和删除（Liang等人2017）等方法会导致句子不自然。

形式上，除了能够愚弄目标模型外，自然语言攻击系统的输出还应满足三个关键的效用保留属性：

（1）人类预测一致性: 人类预测应保持不变。

（2）语义相似性: 根据人类的判断，精心制作的示例应具有与源相同的含义。

（3）语言流利性: 产生的例子应该看起来自然、合乎语法。

此外，几乎没有任何工作攻击新兴起的文本分类bert模型。

在这项工作中，我们介绍了textfooler，一个简单但强大的黑盒设置中自然语言攻击的基线，在这种情况下，不需要访问模型架构或参数。贡献：

1. 我们设计了一个更全面的范例来创建语义和语法上相似的对抗性示例，以满足上述三个要求。具体地说，我们首先确定目标模型的重要单词，然后优先使用语义最相似、语法最正确的单词替换它们，直到预测被更改。
2. 我们成功地应用这个框架分别在五个文本分类任务和两个文本蕴涵任务中攻击了三个最先进的模型。在对抗性的例子中，我们可以将所有任务中几乎所有目标模型的准确度降低到10%以下，而只有不到20%的原始单词受到干扰。
3. 此外，我们还验证了生成的示例：（1）被人类评估者正确分类，（2）在语义上与原始文本相似，（3）在语法上被人类接受。

形式化定义：

给定一组句子，和一组相应的标签，我们有一个预先训练好的模型、 它将输入文本空间X映射到标签空间Y。

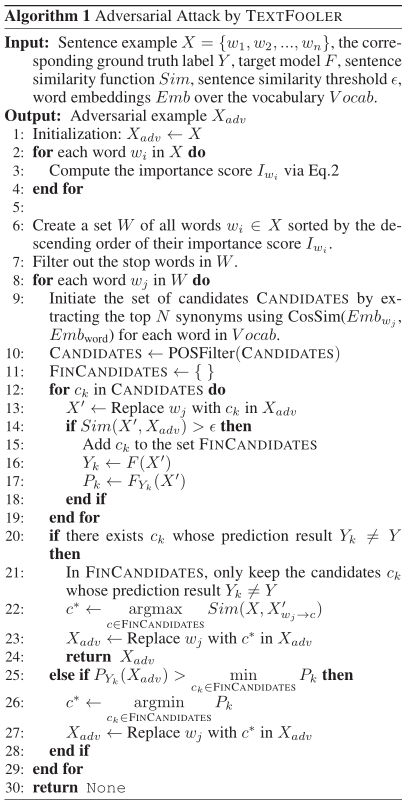
目标是设计对抗样例，使得：





Sim是一个相似性函数，是一个最小相似性。Sim在NLP中通常是语义和句法相似性度量。

方法（注意是黑盒攻击，我们只能用输入和输出作参考）：

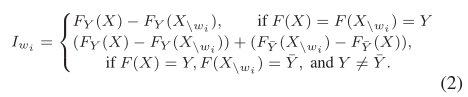


第一步：词重要性排序（蓝色框）

给出n个单词组成的句子，我们观察到只有一些关键词作为预测模型f的影响信号，即BERT关注一些单词的统计线索。因此，我们创建了一个选择机制来选择对最终预测结果影响最大的单词。使用这个选择过程，我们将更改最小化，从而尽可能地保持语义相似性。

注意，在白盒场景中，重要单词的选择并不重要，因为它可以通过检查模型f的梯度轻松解决，而大多数其他单词是不相关的。然而，在本文较为常见的黑箱设置下，模型梯度是不可用的。因此，我们创建如下选择机制：

我们用得分来衡量一个词对分类结果的影响。我们在删除单词后表示的句子为，用表示他们标签的预测得分。由此重要性得分公式如下：



根据重要度评分对单词进行排序后，我们进一步过滤掉来自NLTK和spaCy库的停止词，如“the”、“when”和“none”。这个简单的过滤步骤对于避免语法破坏非常重要。

第二步：单词转换器（其余行）

对于在第一步中获得高重要性得分的单词，我们需要设计一个单词替换机制。一个合适的替换词需要满足以下条件:

(1)与原替换词具有相似的语义，

(2)适合周围的语境，

(3)迫使目标模型做出错误的预测。

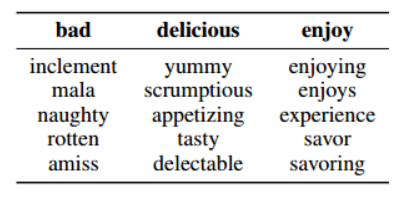
为了选择符合这些标准的替换词，我们提出以下方法：

2.1 同义词提取（第9行）

我们为所选单词wi的所有可能替换收集一个候选集CANDIDATES。CANDIDATES的选择是根据词汇表中每一个词的余弦相似度，选最接近的N个同义词。

为了表示单词，我们使用单词嵌入Glove向量。这些单词向量是专门为寻找同义词而精心设计的，因为它们在SimLex-999上实现了最先进的性能，SimLex-999是一个旨在衡量不同模型判断单词之间语义相似度的数据集。

Glove的替换例子：



利用这组嵌入向量，我们识别出余弦相似度大于δ的前N个同义词。值得注意的是，增大或降低δ都将产生更多样化的同义词候选；然而，对手句和原句之间的语义相似度会降低。在我们的实验中，经验设置n为50，δ为0.7，在多样性和语义相似度控制之间取得了平衡。

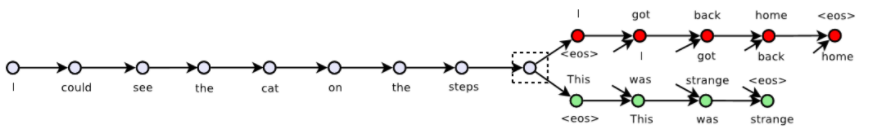
2.2 POS检查（第10行）

在单词wi的CANDIDATES集合中，我们只保留与wi具有**相同词性**(POS)的词。这一步是为了确保大部分文本的语法是正确的。

2.3 语义相似性检查（第11-19行）

对于每个剩余的单词，我们将其代替原始词wi代入句子X中，得到对抗例。我们使用目标模型f来计算相应的预测得分。我们还计算了源句和对抗句之间的语义相似度。具体来说，我们使用Universal Sentence Encoder (USE) 将两个句子编码成高维向量，并使用它们的余弦相似度得分作为语义相似度的近似。单词的相似度得分高于预设阈值被放入最终候选池。

注：USE使用了Transformer的Encoder部分，无监督训练使用skip-thought的方法，也就是通过Encoder得到当前句子向量，预测句子前边（红色）、后边（绿色）句子中的词，使用GRU训练得到。

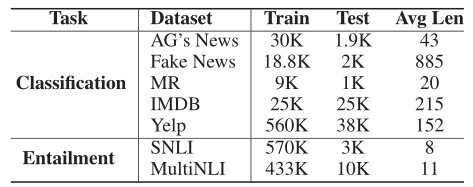


2.4 对抗样例的终止（第20-30行）

在最终的候选池FINCANDIDATES中，如果存在任何已经可以改变目标模型预测的候选词，那么我们将在这些获胜的候选词中选择语义相似度得分最高的词。但如果不是，那么我们选择使y成立的得分最低的单词作为wi的最佳替换词，并重复步骤2来转换下一个选择的单词。

实验：

我们研究了对抗性攻击在两个重要的NLP任务上的有效性，即文本分类和文本蕴涵。下表总结了数据集的统计信息。从测试集中随机选择1000个例子来评估我们的算法。



各数据集介绍：

分类问题：为了研究该模型的鲁棒性，我们使用了具有不同属性的文本分类数据集，包括新闻主题分类、假新闻检测、句子级和文档级情感分析，平均文本长度为几十到几百个单词。

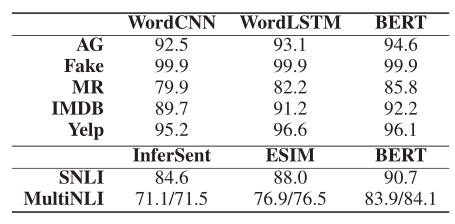
1. AG的新闻(AG): 基于四个新闻主题的句子级别分类:世界、体育、商业和科学/技术。我们将每篇新闻文章的标题和描述字段串联起来。
2. 假新闻检测(Fake):对新闻文章是否假进行文档级分类。数据集来自Kaggle Fake News challenge。
3. MR:对积极和消极电影评论的句子级情绪分类。我们使用90%的数据作为训练集，10%作为测试集，遵循论文的实践。
4. IMDB:对正面和负面影评进行文档级情感分类。
5. Yelp极性(Yelp):正面评论和负面评论的文档级情感分类。评分为1和2的评论被标记为负面，评分为4和5的评论被标记为正面。

文本蕴含：

1. SNLI:一个来自图像字幕的570K句对数据集。任务是判断两句话之间的关系:第二句话是否可以由暗含、矛盾或与第一句的中性关系衍生出来。
2. multicli:一个多类型包含数据集，包括转录演讲、流行小说和政府报告。与snli相比，它包含更复杂的语言与各种书面和口语英语文本。

攻击的模型：

对于每个数据集，我们在训练集中训练了三个最先进的模型，并获得了与原始实现相似的测试集准确性得分，如表所示。然后生成语义上与测试集相似的对抗性例子来攻击训练的模型，并使它们产生不同的结果。



在句子分类任务中，我们关注：

1. 基于单词的卷积神经网络(WordCNN) (Kim 2014)，我们使用了3、4和5三个窗口大小，并为每个窗口大小使用了100个过滤器，dropout为0.3。
2. 基于单词的长-短时记忆(WordLSTM) (Hochreiter和Schmidhuber 1997)，我们使用了一个具有150个隐藏单元的1层双向LSTM, dropout为0.3。对于这两个模型，我们使用了200维的手套词嵌入，预训练了来自维基百科和Gigawords (Pennington, Socher, and Manning 2014)的6B的token。
3. 最先进的双向编码器表示BERT (Devlin et al. 2018)，我们使用了带有768个隐藏单元和12个头的12层BERT模型，带有110m参数，这被称为基础未套壳（base-uncased）版本

我们还在文本蕴涵任务上实现了三个目标模型：标准的InferSent(Conneau et al. 2017)、ESIM(Chen et al. 2016)和微调的BERT。

自动评估：

我们首先将攻击前的目标模型在原始测试样本上的准确性作为原始准确性报告。然后我们测量目标模型的准确性与从测试样本制作的对抗样本，表示为攻击后的准确性。通过比较这两个准确性得分，我们可以评估攻击的成功程度，**原始和攻击后的准确度差距越大，说明我们的攻击越成功**。

1. 准确性
2. 被干扰单词的百分比，即被干扰单词的数量与文本长度的比率。
3. 使用use来衡量原文和对抗文本之间的语义相似度。
4. 被干扰词的百分比和语义相似度得分，一起评估原始文本和对抗文本在语义上的相似度。
5. 攻击系统对目标模型的查询次数，并获取输出概率得分。这个指标可以显示攻击模型的效率。

人类评估：

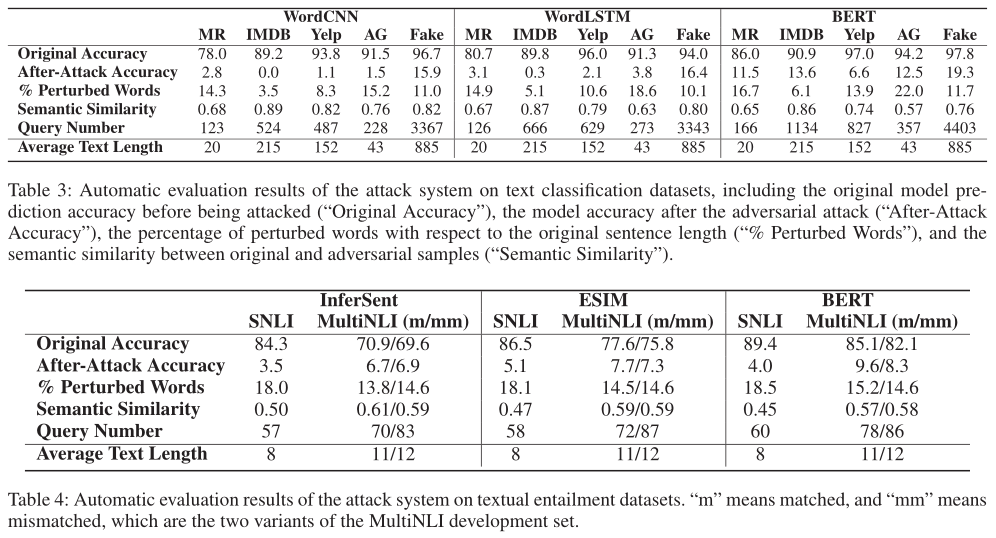
我们在三个标准上进行人类评价:语义相似性，语法性和分类准确性。我们随机选取每个任务的100个测试句子生成对抗性例子，一个针对MR数据集上的WordLSTM，另一个针对SNLI数据集上的BERT。

首先，我们将原始文本和对抗性文本混合在一起，并请人工法官按照Likert 1-5的等级对它们的语法性进行评分。接下来，我们通过让人对原始句和对抗性句的混合句进行分类来评估分类的一致性，然后计算两种分类结果的一致性率。最后，我们通过让人判断生成的对抗句是否与源句相似、歧义或不同来评估原始句和对抗句的语义相似度。每项任务由两名独立的人工裁判完成，他们都是英语为母语的人。志愿者具有大学学历，并在开始批注前通过了一批测试。

结果：

总的来说，从我们的结果可以看出，textfooler在对两个任务进行有限数量的修改时，达到了很高的成功率。无论多么长文本序列，无论多么准确的目标模型，textfooler总是从最先进的降低精度值低于15% (除了fake数据集)和微扰比率不到20%的词(除了BERT目标模型下的AG数据集除外)。

例如，当使用WordLSTM模型将IMDB数据集的准确率从89.8%降低到0.3%时，它平均只扰动了5.1%的单词。值得注意的是，我们的攻击系统使WordCNN模型在IMDB数据集上完全错误(达到了0%的精度)，只有3.5%的单词摄动率。在平均长度为215个单词的IMDB数据集中，系统仅为每个样本扰动10个或更少的单词来进行成功的攻击。这意味着我们的攻击通过巧妙的操作，系统能够成功的完全误导分类器，使其给出错误的预测。



即使bert这种相比非预训练模型WordLSTM和WordCNN，达到看似“健壮”性能，我们的攻击模型仍然可以减少其预测精度5 - 7倍的分类任务(例如,Yelp数据集从95.6%到6.8%)和9-22倍NLI任务(例如,从89.4%到4.0%的SNLI数据集)，这是前所未有的。我们精心设计的对抗性例子有助于研究BERT模型的可解释性。

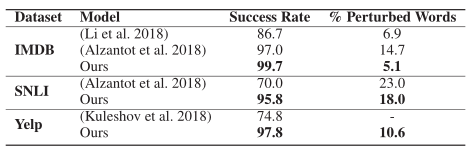
从表3和表4中可以得出另外两个观察结果。(1)原始精度越高的模型，一般越难以被攻击。例如，BERT模型在所有数据集上都比WordCNN具有更高的攻击后精度和摄动词率。(2)对于所有三种目标模型，Fake dataset的攻击后精度都远远高于其他所有分类数据集。我们在实验中发现，攻击系统很容易将真实新闻转化为假新闻，而将真实新闻转化为假新闻则要困难得多，这与直觉是一致的。

比较表3和表4中的语义相似度得分和扰动词比，我们发现两者的结果有很高的正相关。从经验上看，当文本长度超过10个单词时，语义相似度度量变得更加稳定。由于文本分类数据集的平均文本长度都在20个单词以上，而文本蕴涵数据集的平均文本长度在10个单词左右或以下，所以我们需要分别对待这两个任务的语义相似度得分。因此，我们对每个任务的词摄动率和语义相似度进行了线性回归分析，得到文本分类和文本蕴涵任务的r-squared值分别为0.94和0.97。如此高的r平方值表明我们提出的语义相似度与被摄词率具有高度的相关性(负相关)，这两者都可以作为评价原始文本改变程度的很好的自动度量。

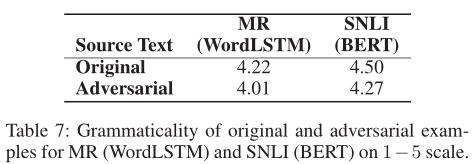
我们在表3和表4的最后一行中包含每个数据集的平均文本长度，以便可以方便地与查询数进行比较。查询号几乎与文本长度成线性关系，其比率为(2,8)。较长的文本与较小的比率相关，这验证了textfooler的效率。

攻击率：

从表结果中，我们可以看到，我们的系统通过攻击成功率(通过错误预测的数量除以对抗例子的总数计算)和被摄字率击败了之前最先进的模型。



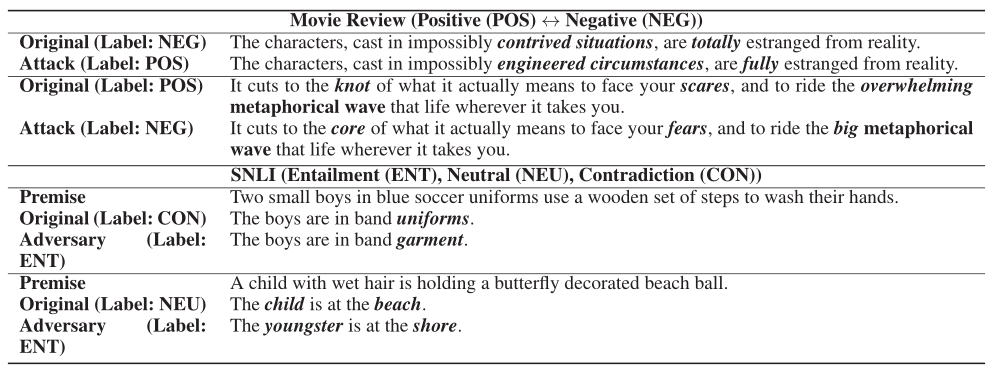
人类评价：



如表所示，在两个数据集中，对抗性文本的语法性都接近于原始文本。通过明智地替换同义词，textfooler生成平滑的输出。然后，我们要求人类评价者给一组打乱的原始样本和对抗样本分配分类标签。原始判决和反对判决的整体一致性相对较高，MR和SNLI的一致性分别为92%和85%。虽然我们的对抗性句并不是在每一种情况下都是完美的，但这表明，从人类的角度来看，大多数对抗性句都具有与原句相同的属性。

实例：

该表显示了一些意义几乎相同的句子的典型例子，这些例子导致目标模型的分类存在矛盾。

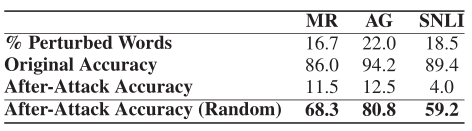


（差异感知）我们请人类评价者们决定每一个抗辩样本是否保留原判决的意思。他们需要决定是否合成对抗前的样本与所提供的原始句子相似、歧义或不同。我们将相似度设为1，歧义度设为0.5，不相似设为0，得到的句子相似度在MR上为0.91，在SNLI上为0.86，这表明原始文本和对抗文本的感知差异很小。

消融实验：

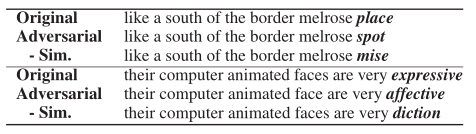
去掉Step 1，随机选择要扰动的单词后，三个数据集的攻击后精度提高了45%以上，这表明没有单词重要性排序步骤，攻击是无效的。

理由：关键词重要度排序过程是该算法的关键，它可以准确有效地定位对目标模型预测影响最大的关键词。该策略还可以减少干扰词的数量，尽可能保持语义相似度。

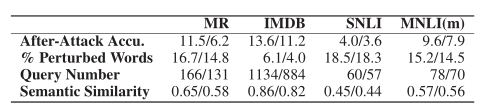


将step2的句子相似度替换为单词相似度，结果是很差的。

理由：通过单词嵌入提取的同义词是嘈杂的，所以直接将它们作为对抗样本注入文本可能会显著地改变语义。通过应用句子级语义相似度约束，我们可以获得更多的相关同义词作为好的替换。



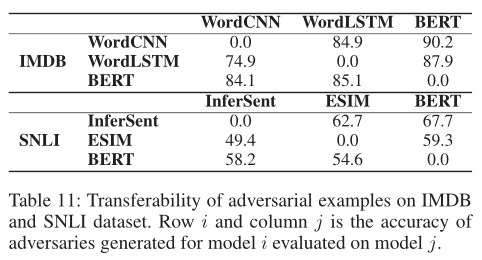
（第二行是用了语义相似度，第三行没有用）



（/前是语义相似度，后是没有语义相似度，对比数据是bert的一列）

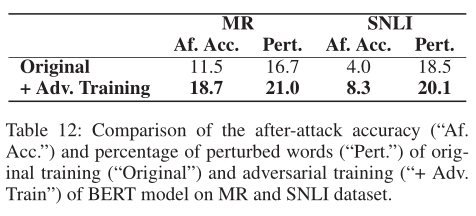
可迁移性：

我们考察了对抗性文本的可转移性，也就是说，基于一个模型的对抗性样本是否也可以愚弄另一个模型。为此，我们从IMDB和SNLI测试集中收集了一个目标模型预测错误的对抗性例子，然后测量它们对其他两个目标模型的预测精度。从表11的结果可以看出，模型之间存在一定程度的可转移性，文本蕴涵任务的可转移性高于文本分类任务。此外，基于预测精度较高的模型，即本文的BERT模型生成的对抗样本具有更高的可转让性。



补充对抗样例后的训练：

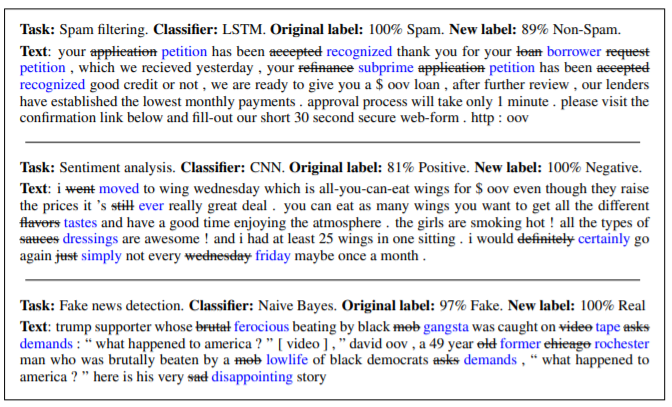
我们的工作为如何通过这些对抗性的例子更好地改进原始模型提供了见解。我们共同对对抗性训练进行了初步实验，将原始数据和对抗性样本(对抗性样本与原始样本具有相同的标签)同时输入模型，观察原始模型是否能够获得更强的鲁棒性。我们从MR和SNLI训练集中收集了愚弄BERT的对抗性例子，并将它们添加到原始训练集中。然后，我们使用扩展的数据从零开始训练BERT，并攻击这个对抗训练模型。从表12的攻击结果中可以看出，经过对抗性再训练后，攻击后的准确率和摄动词率都提高了，说明攻击的难度越大。这揭示了我们攻击系统的一个潜力，我们可以通过生成对抗例子来训练模型来增强其对未来攻击的鲁棒性。



小结：

我们研究了在黑盒设置下，针对最先进的文本分类和文本蕴涵模型的对抗性攻击。大量的实验证明了我们提出的系统textfooler的有效性，这是一个产生目标对抗文本的系统。人类研究证实，生成的对抗性文本是易读的、语法上的，并且在意义上与原始文本相似。

附(ICLR 2018 Conference)：



三个在文本分类任务中使用对抗样本的例子。分别是垃圾邮件分类、情感分析、虚假新闻检测。全是依靠同义词替换实现的。

在自然语言环境中，我们希望改动的样本x'能与原样本x保存相同的意思。为了达到这个目的，我们提出了一种特殊的限制函数c(x,x’)，使两种表述保持相同的句法特征。具体来说，函数c由两个约束组成，分别在两个层次上捕捉句子的相似性。

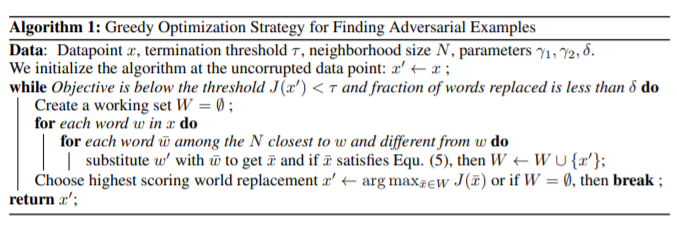
1. 语义层次。我们用思考向量（thought vector）的概念来捕获话语的含义。一个思考向量可以被看作是一个从句子到向量空间的映射，在这个空间中，意义相近的句子是相互接近的。在此意义下，约束被定义为：。其中v和v'是与x和x'相联系的思考向量，是一超参数。有许多计算思考向量的方法，它们可以在我们的优化算法中共存。思考向量是单个词向量的平均值。
2. 句法层次。通常思考向量不能捕捉到句子的句法正确性。比如，重新对句子中的单词进行排序可以得到相同的平均词向量。为此，我们加上了句法约束，它依赖于一个语言模型：特别地，我们要求语言模型的概率在扰动样本与原样本中是相似的：

我们建议在与f相同的数据集上训练语言模型，这会使模型更好地捕捉到x像垃圾邮件或者影评的内容。同时还要求x和x'的语言模型概率是相似的：因此如果x是一个不合文法的句子（正确性较低），则x'也要保持相似的正确性。

前文提到，构造对抗样本就是要解决优化问题



为此可以采用一种贪婪启发式算法（算法1）：

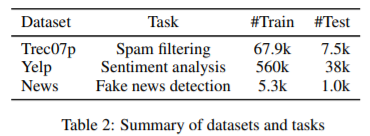


简单地说，我们提出了一个迭代过程，在每个步骤中考虑对一个句子的所有有效的单词变化(只变换一个单词——用同义词进行替换，且需要满足我们的限制)，并选择最能提高目标的那个。

算法：

1. 算法输入：需要一个目标分类器 f，算法1通过优化目标 J 将 x 转化为 x'。假设 x 是一组n个离散符号（即单词word），用 wi 表示，i=1,2,...n。尽管这个算法是定义在自然语言环境中的，但是可以扩展到相似的通常的离散问题。
2. 优化策略：首先我们定义一个边界向量，例如使其满足，这可以使我们排除掉一些显然不满足条件的对抗改动。或者设置一个关于目标的最小阈值 ，例如目标标签分数的最小期望值，在我们达到这个最小阈值时停止算法。
3. 单词替换：我们用一个合适的词向量空间中的最近邻居（nearest neighbors）替换单词，并考虑N个最近的邻居。这些最近邻居单词很可能出现在相同的文本环境中。为了确保更换的也是同义词,我们使用的glove词向量（glove word vectors）进行后期处理。这确保了向量满足已知的同义词关系所施加的语言约束，并确保具有相似含义的单词在向量空间中彼此接近。

实验：在table2中提到的三个自然语言分类任务中使用对抗样本。

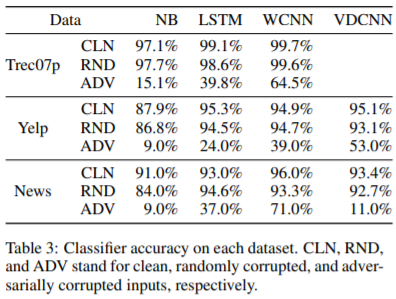


将训练集中的10%用作验证集，所有对抗样本通过测试集产生、评估。同样,我们为每一项任务，在训练集中训练一个卦语言模型（trigram language model）。并且实例化 中的词向量语义约束。

Baseline：

1. 朴素贝叶斯（naive Bayes）：我们将每个文档转换为一个词袋（bag-of words）表示形式，然后按照Wang & Manning(2012)的方法，将单词特征进行二值化，并使用一个多项式模型进行分类。
2. 长短记忆网路（long short-term memory）：我们建立了一个拥有512个隐藏神经元的单层LSTM。在输入LSTM前先进行word2vec词嵌入，变成300维向量。然后，我们对LSTM在每个时间步上的输出进行平均，得到一个特征向量，用于最后的逻辑回归预测情感。
3. 浅单词级卷积网络（shallow word-level convolutional networks）：我们用嵌入层(如在LSTM中)训练一个CNN，一个时间卷积层，然后是最大池化，最后用一个全连接层进行分类。
4. 深字符集卷积网络（deep character-level convolutional networks）

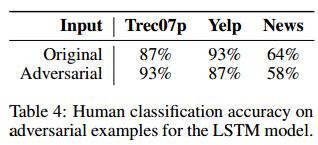
结果：

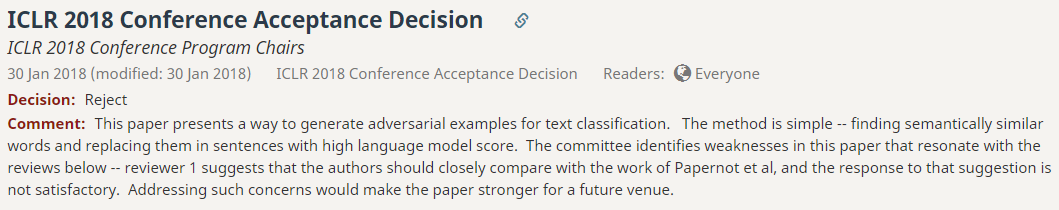


我们手动选择了优化设置，从而在对抗样本的强度和一致性之间做出了合理的权衡。在所有实验中，我们令阈值，临近大小N=15，参数。在情感分析和假消息检测中令，在垃圾信息检测中令。我们还比较了用随机抽样代替算法1中的argmax得到的随机扰动。发现模型对于随机扰动的抵抗能力很强。

人类评估：

我们通过Amazon Mechanical Turk上的人类实验，验证了对抗样本的质量和一致性。首先，我们从测试集中二次抽样了100个样本，包括原始样本与对抗样本，让人类进行标记（例如积极评价或负面评价）。





Text2img