==========

* 1. **最小开销包装器**

为了提高包装器的健壮性，本文提出了一种最小开销包装器。首先利用文档数据中页面的特征，构造一组抽取规则，生成候选的包装器；针对不同时间短的网页快照，通过模型学习出参数使得编辑开销尽可能小，并通过评估选出最优包装器。

**4.3.1 编辑开销**

在第2部分，我们描述了对一个网页可能的编辑操作。现在我们描述这些操作的对抗编辑开销。令为所有标记的集合。我们假设有一个开销函数，来自，用代表将一个标签节点替换为的开销，表示插入一个标签的节点的开销，表示删除一个标签节点的开销。注意，。另外，我们假设三角不等式成立，即，。

给定一个编辑脚本，表示的开销，就是中每一次操作开销的总和。

**4.3.2 抽取可信度**

我们需要确定在给定版本中抽取的可信度。注意，我们定义的可信度不是传统意义上的。它不是[0,1]中的一个标准化数字，而是一个非负数，它给出了抽取好坏的一个指标。可信度越大，抽取越准确。

如果和差别很大，抽取的可信度就会很低。如果中所有改变都在远离D-node节点的部分，可信度就会很高。基于这个结论，我们定义抽取一个新版本的可信度如下。

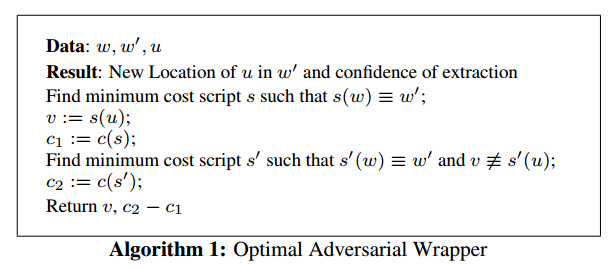
令为把网页变成开销最小的编辑脚本，即。抽取的节点为。同样，把网页变成开销最小的脚本，即，但是没有把映射到与相应的节点。定义抽取可信度为二者之差。就是需要抵消最优对抗包装器所需的额外开销。如果差别很大，抽取的节点就很容易从其他节点中分离出来，抽取就不容易出错。

例如，看图3中例子。如果我们用包装器“抽取的第二个叶子”，包装器的可信度就是0，因为有一个开销相同的编辑脚本（即，顶部那个），得到的分裂点不同。

为计算可信度，我们需要解决以下约束编辑脚本的问题：给定两个树和，两个节点和，计算出和之间最小的编辑脚本，约束是没有映射到。张的方法本来是解决无约束最小脚本开销问题，扩展成有约束的版本很容易。计算可信度与计算成对的树开销，复杂度一样大，。

**4.3.3 抽取算法**

抽取算法和可信度计算过程见算法1。



该算法输入是：有标记的网页，新版本网页，以及中D-node节点。算法首先找到最小开销的脚本满足：与同构，使用脚本映射D-node节点到新网页中，并求出编辑脚本的开销。然后找到最小开销的脚本满足：与同构，但没有映射到，同时求出编辑脚本的开销。包装器的抽取可信度即。

输出为中D-node节点的位置以及抽取可信度。

=&……￥&（\*……%&=========================

|  |  |
| --- | --- |
|  |  |

2