泰比
ABBYY

如何测试OCR SDK

White Paper

目录

介绍

选择软件开发工具包 (SDK) 是一项非常重要的任务，因为在选择阶段做出的任何决策都会对您的应用程序和业务产生长期影响，并且在后期更换技术时可能会伴随着诸多困难。因此，必然要进行详尽而周密的测试和评估过程。但是，由于以下原因，可能对 OCR SDK 产品的评估赋予了挑战：

1.为了测试 OCR 引擎，您需要一个测试工具和一个大型样本图像数据库。

2.目前有很多 OCR 供应商提供多种 SDK 解决方案，虽然已经有一些来自可靠来源的公开测试，但是这些大多是在一些普遍情况下进行的测试，学术性比实际性要更高。实际具有适用性的测试结果应该是在现实条件下由计划用例来确定的。我们将在下面详细讨论这个问题。

3. 必须要用多种语言测试多个参数: 词汇水平、符号准确度、Office 文件格式的保留、创建PDF 文件的大小等等。有些参数可以自动化测试，而剩下的参数只能用眼睛检查。对于不同的任务或者场景，您可能需要测试不同的参数。

4. 为了针对特定任务调整OCR 引擎，在大多数情况下，开发人员应该至少具有OCR技术的基本知识。

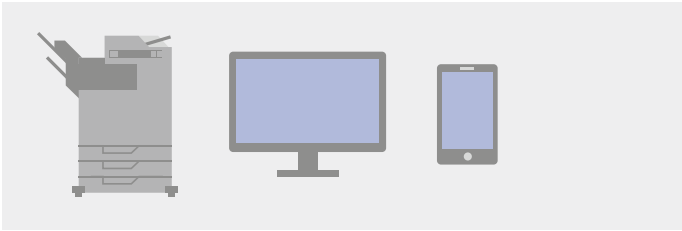
开始之前

由于无法凭空测算 OCR SDK 的关键特性，所以并没有一个明确的答案来衡量特定 OCR SDK 的准确性。正如汽车的速度和效率有时取决于道路条件或燃料质量一样，OCR SDK的准确性和速度完全取决于任务和技术条件，例如服务器参数、操作系统、场景、文档类型，文档质量等等。

图像库

首先，您需要准备一个图像库进行测试。以下是需要记住的关键事项：

1. **庞大的图像库。**您需要有足够大的图像库，为了获得高度可靠的结果，我们建议您收集几千张图像。 如果这太复杂的话，至少也要收集几百张图像，因为图像的数量会影响测试结果的可靠性。如果您的测试文件只包含 100 页文档，则单个文档页面已占1% 的准确度，这样会对整体精度测试结果有很大影响 (如果您在文档中进行测试，请参阅 “如何计算精度”) 。
2. **文件类型。** 收集与您将在识别中处理的文档类型相对应的图像。 例如，如果您要处理发票，收集发票文档样本，如果您计划处理协议，则收集协议文档样本等。这一点很重要，因为不同的OCR引擎可以更好的处理不同类型的文档 (因为最初在不同的工作场景下创建了不同的 OCR 引擎) 。 我们建议在测试中使用与识别转换中相同比例的文档类型。 这将更加贴近您在后续识别任务中可以预测的准确率。
3. **图像源** (包括: 扫描文档、手机照片、不同类型的PDF) 。 仅对捕获的图像进行测试，方法与在识别过程中捕获的图像相同。 举例来说，如果您使用扫描文档来测试 OCR SDK ，而您在工作中处理的是智能手机拍摄的照片，这样是不起作用的。 原因是相同的: 不同的OCR引擎可以更好地处理来自不同来源的文档。



4. **真实的文档。** 测试 OCR 引擎的最佳方法是对要在工作中处理的文档的实际样本进行测试。使用专为测试而人工创建的假文档来测试 OCR 引擎并不会产生实际结果。 另外，请避免使用作为一个 OCR SDK 的测试基础的图像库来测试另一个 OCR SDK。 每个供应商都会为您提供他们产品相关的 OCR 图片，甚至为您提供他们调整 SDK的文档。采用同一组测试文件的其他供应商的产品测试结果可能会很糟糕。

精度测量

如何将OCR结果与原始文本进行比较:

有几种方法可以将OCR结果与原始文本进行比较:

1. **肉眼估计。** 评审OCR测试结果并以某种方式对其进行评级。 优点: 此方法速度快, 您可以从列表中取消选择最不合适的OCR引擎。 局限性: 因为这样操作太耗时间, 无法在数百张图片上进行。 因此, 您无法统计得到可靠的测试结果。
2. **跟踪Microsoft Word中的更改。** 优点: 该方法适用于少量文档的测试。局限性: 测试量低, 结果不可靠。 除源文档外, 还需要具有DOCX文件格式的原始文档, 以便将它们与OCR结果进行比较。
3. **Peek-a-Boo。** 优点: 此方法可以将文件转化为可编辑格式, 并且适合测试OCR引擎提供的布局保留。 局限性: 仅适用于少量测试, 并且测试结果不可靠。 我们强烈建议您不要使用此方法进行数据捕获。
4. **已验证的源图像基础 + 已解析的结果转换为XML格式 (参考数据)** 。 优点: 此方法将为您提供最具统计可靠性的结果。 它独立于语言, 适合大批量测试。 局限性: 此方法需要一些额外的时间来准备测试基础。

在本文档中, 我们采用第四种方法描述精度测量, 因为它提供了最可靠和可重复的结果。

衡量什么

有几种方法可以衡量准确度:

1. 通过计算正确识别的符号 (字符) 的百分比。
2. 通过计算正确识别的单词的百分比。 只有在单词中的所有字符都被正确识别时, 才认为该单词被正确识别。 **注意:** 对于方法1和2, 您将需要包含具有正确字符的参考数据作为测试基础。
3. 通过计算文档结构的正确检测率。

注意: 对于最后一种方法, 在您的测试基础参考数据除了需要包含正确的字符以及它们的坐标信息外, 还需要包含每个文档的结构信息。这些可以在 XML 中通过特殊实体指定的页眉、页脚、文本列、打印体、图像对象、背景等被定义。

字符错误率可以这样计算:

$$CER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$$

变量说明:

- S是替换的数量,
- D是删除的数量,
- I 是插入的数量,
- C是更正的数量,
- N是参考数据中的字符数量 (N = S + D + C) 。

单词错误率可以这样计算:

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$$

变量说明:

- S是替换的数量,
- D是删除的数量,
- I 是插入的数量,
- C是更正的数量,
- N是参考数据中的字符数量 (N = S + D + C) 。

在此查看更多信息 (https://en.wikipedia.org/wiki/Word_error_rate)。

要计算S, D, I和C, 您需要计算LevenshteinDistance, 比如借助以下算法:

```
// len_s and len_t are the number of characters in string s and t respectively
int LevenshteinDistance (const char *s, int len_s, const char *t, int len_t)
{
  int cost;

  /* base case: empty strings */
  if (len_s == 0) return len_t;
  if (len_t == 0) return len_s;

  /* test if last characters of the strings match */
  if (s[len_s - 1] == t[len_t - 1])
      cost = 0;
  else
      cost = 1;

  /* return minimum of delete char from s, delete char from t, and delete char from both */
    return  minimum(LevenshteinDistance(s,  len_s  -  1,  t,  len_t       )  +  1,  /*insertions*/
                                    LevenshteinDistance(s, len_s     , t, len_t - 1) + 1, /*deletions*/
                                LevenshteinDistance(s, len_s - 1, t, len_t - 1) + cost); /*substitutions*/
}
```

在此处查看更多信息 (https://en.wikipedia.org/wiki/Levenshtein_distance#Computing_Levenshtein_distance).

建议仅在单词的顺序清晰时才应用此算法。 例如, 将其应用于在页面上检测到的文本块, 但不应用于整个页面。

对方法的选择取决于您的项目或者测试场景，然后**选取最优方法。**

可搜索的PDF。如果要将图像转换为可搜索的PDF，则应计算单词的正确识别百分比，因为最终用户将使用单词来检索整个OCR结果，而不是通过符号来检索。

可编辑的格式。如果要将图像转换为DOCX，XLSX等，则需要计算正确识别的符号的百分比并评估布局的保留水平。

数据捕获。如果要基于OCR结果实现自己的数据捕获，最好计算正确识别的单词的百分比。或者，分别计算正确识别的字段的百分比和有错误的字段中的平均错误数可能会更好。 这将给出需要手动编辑的字段数量以及每个字段需要编辑的数量。

对于数据捕获，定义重要的文档是一个不错的办法。 例如，不需要特别识别发票上的页脚，但找到“发票日期”和“总计”等字段至关重要。 牢记这一点将使您能够确定哪个OCR引擎可以更好地满足您的特定任务。 数据捕获通常需要比其它方案有更高的准确性。 在数据捕获方案中，参考数据应包括用于定位这些字段的字段和这些字段的关键字。

如何计算精度

有几种方法可以计算一组文档的平均准确度：

1. 确定被正确识别的符号/单词在整个测试文档中占的百分比
2. 首先计算在每个文档中找到多少个正确的符号/单词，然后计算一组中全部文档的平均百分比。
3. 对于多页文档：首先分别测量每页的准确度，然后计算文档中所有页面的平均准确度，最后是计算测试集中所有文档的平均准确度。

还有一个值得一提的方面 —— 错误处理逻辑。 通常，每个未正确识别的符号/单词将意味着被减1分。 然而，某些类型的错误不那么重要甚至可以忽略，而其他类型的错误可能在计划的处理场景中会产生严重后果，并且应该减超过 1 分来处理。

选择合适的方法：计算方法和错误处理逻辑的选择取决于预期方案。

可搜索的PDF。 如果将您的方案转换为可搜索的PDF, 最好首先计算每个文档中找到的正确识别的符号/单词的百分比, 然后计算一组中所有文档的平均百分比。 重要的是要了解用户能够找到多少包含特定关键字的文档。 错误处理: 错误处理逻辑中应忽略标点符号, 因为它不用于搜索。

数据捕获。 对于此方案, 最好计算正确识别的关键字的百分比和在每个文档中找到的正确提取的字段值, 然后计算集合中所有文档的平均百分比。 通过这些数字, 您可以了解以100% 精确自动捕获的文档的数量以及需要审核的字段数量。

对于文档内关键字段值中发现的错误, 结果应该更严格地处理。 另一方面, 如果可以正确地找到这些关键词并且这些错误在一个文档到另一个文档中或多或少地存在重复错误, 则用于定位字段的关键字中的错误就不那么关键了。 这些关键词后来可以使用 Two-Pass OCR方法重新OCR 扫描。

可编辑的格式。 对于这种情况, 建议计算多少单词、表格、页脚等等, 对于集合中所有被正确识别的文档, 在计算每个文档的这些度量标准时是没有用的, 因为在这种情况下唯一重要的事情是估计使用应用程序的用户总共需要做多少次更正。

衡量速度

记住 OCR 过程需要以下步骤很重要:

1. 初始化引擎
2. 图像处理 (包括预处理, 分析, 识别和合成)
3. 取消初始化引擎

这些步骤中的每一个都可能显著影响最终速度标准。 但是, 并非所有这些步骤都适用于所有方案。

例如, 当一次处理大量文档 (批处理) 时, 不需要为每个文档初始化引擎, 因此不需要测量每个文档处理的初始化时间。如果在处理中图像将通过RAM传输进行处理 (以确保高速), 那么应在测试算法中实现相同的逻辑。在这种情况下从磁盘打开图像时, 就不适用于测试的条件, 因为最终速度标准会有很大差异。

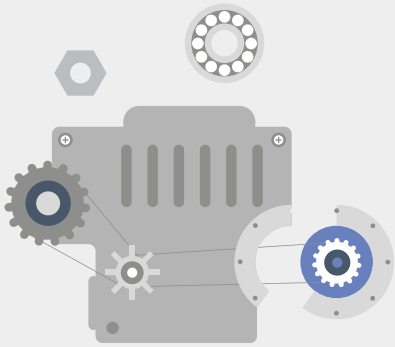
这就是为什么只测量您在处理中预期的那些步骤的速度很重要。 小提示:

- 为了获得更可靠的结果，建议多次运行速度测试并计算平均时间，因为根据与操作系统的交互，结果可能会有多个百分点变化。
- 始终使用您要在工作中处理文档的真实样本，在现实生活中进行测试。 例如，如果您计划处理BMP文件，请不要在测试集中使用JPG文件，因为速度可能会有很大差异。

- 确保您正在学习在测试的OCR SDK的API、代码示例和教程 (如果有)、以了解它为速度优化提供的工具 (例如，对象重用)。 如果速度对您至关重要，需要根据您的需要进行优化，切勿使用引擎的默认设置。
- 尝试各种设置以查看可以关闭的选项以提高运行速度 。

产品分发大小

- **Ad-hoc场景。** 产品分发大小会影响初始化时间，如果每次有新任务进行处理时都启动引擎，我们建议您检查产品分发的大小。
- **移动应用程序。** 移动应用程序的分发大小将始终是移动应用程序开发人员考虑的重要因素，这就是OCR引擎大小是最关键参数之一的原因。
- **浏览器插件。** 分发体积越大，插件体积就越大，会导致浏览器插件的安装时间越长。
- **驱动。** 由于驱动通常要通过互联网定期更新，因此较大分发体积可能会产额外的负担。 驱动程序有时可以在RAM中运行，但是RAM的大小有限。如果将驱动程序根据请求运行到RAM中，则大的分发体积将导致额外的延迟。
- **Web服务。** 如果将OCR集成到Web服务中，SDK的分发体积将对扩展服务的速度产生负面影响，从而影响为提供足够适用性所需的处理能力。



有两种主要方法可以减少SDK分发体积:

1.排除对您的任务不那么重要的语言的词典。

2.排除引擎代码的一部分，这些部分可以提高准确性但不是那么关键，因此在排除后，准确度仍然可以接受。 您应根据自己的情况和用户期望自行决定您的任务可接受的准确度。但是排除并不总是可行的，或者并不总是很容易的。 在很多情况下，您必须创建自定义SDK分发。

供应商的支持和可靠性

选择OCR引擎时，技术特性并不是唯一需要考虑的重要事项。

OCR引擎是否得到很好的支持？了解您可能的供应商是如何处理功能请求的，检查好文档和代码示例的可用性以及他们的支持人员对其引擎的了解程度。

还要了解该公司是否曾经做过与您类似的项目 (询问参考资料，检查案例研究和网站)。 这可以表明供应商在您的特定方案和文档类型上拥有的专业知识水平。OCR支持不是一项简单的任务，这就是为什么特定领域的实践很重要的原因。

引擎是否定期更新？ 一些OCR引擎多年未升级，而其他OCR引擎定期更新。 更新很重要，因为它们包括客户要求的错误修复和增加新的功能，或者需要符合行业变化。

如果您计划在您的业务中使用OCR，请注意，在某些时候您需要在引擎中修复或改进某些内容。

我们希望，上述建议能够帮助您根据明确且可重复的事实做出正确的决定。 如果您有任何其它问题或想要提供反馈，请通过 sales_3A@abbyy.com 与我们联系！

ABBYY Emerging Markets (3A)
(Asia, Africa, South America)
Otradnaya str. 2b/6, 127273,
Moscow, Russia
Phone: +7 495 783 3700
Fax: +7 495 783 2663
E-mail: sales_3A@abbyy.com
Support: support@abbyy.com

泰比 ABBYY