```
  content_type = doc.meta['Content-Type']

  payload.seek(0)
  data = payload.read()
  doc.set_size(payload.tell())

  if content_type == 'application/pdf' and self.detect_scanned_pdf:
    tmp_path = os.path.join(self.tmp_root,
      '%s-%s.pdf' % (self.wid, doc.docid[0:8]))

    if not os.path.exists(self.tmp_root):
      os.makedirs(self.tmp_root)

    with open(tmp_path, 'wb') as fp:
      fp.write(data)

    if self._is_pdf_scanned(tmp_path):
      data_tiff = self._convert_pdf_to_tiff(tmp_path)

      if data_tiff:
        data = data_tiff
        filename = '%s.tiff' % filename
        content_type = 'image/tiff'

    os.remove(tmp_path)

  headers = {
    'Content-Disposition': 'attachment; filename=%s' % filename,
    'Content-type': content_type,
  }

  if self.ocr_languages:
    headers['X-Tika-OCRLanguage'] = self.ocr_languages

  connection = self.config[helper.INJECTOR].get_http_connection()
  connection.request('PUT', '/tika', data, headers)
  response = connection.getresponse()
  text = response.read().strip().decode('utf-8')
  response.close()
  doc.text = text

@classmethod
def _is_pdf_scanned(cls, path):
  pdf_info = cls._get_pdf_info(path)

  if not pdf_info:
    return False

  try:
    page_count = int(pdf_info['pages'])
  except (KeyError, ValueError):
    return False

  images_info = cls._get_pdf_images_info(path)
```

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

from __future__ import absolute_import, unicode_literals

import os
import subprocess

import gransk.core.abstract_subscriber as abstract_subscriber
import gransk.core.helper as helper


class Subscriber(abstract_subscriber.Subscriber):
  """
  Class for uploading documents to Apache Tika and reading text response.
  Tika is an open source tool that is capable of parsing a vast number (>200)
  of document formats.
  """
  CONSUMES = [helper.EXTERNAL_EXTRACTOR]

  SERVICES = []

  def _accept(self, doc):
    return doc.meta['size'] < self.max_size

  def setup(self, config):
    """
    Define maximum size of document to upload.

    :param config: Configuration object.
    :type config: ``dict``
    """
    self.config = config

    self.max_size = config.get(helper.TIKA_MAX_SIZE, 1024 * 1024 * 64)
    self.ocr_languages = config.get(helper.OCR_LANGUAGES)
    self.detect_scanned_pdf = config.get(helper.DETECT_SCANNED_PDF, True)

    self.tmp_root = os.path.join(config[helper.DATA_ROOT], 'files', '.tmp')
    self.wid = config[helper.WORKER_ID]

  def consume(self, doc, payload):
    """
    Upload document to Apache Tika and add result to document as text.

    :param doc: Document object.
    :param payload: File pointer beloning to document.
    :type doc: ``gransk.core.document.Document``
    :type payload: ``file``
    """
    if not self._accept(doc):
      return

    filename = os.path.basename(doc.path)
```