Tika and Computer Vision - Image Captioning

This page describes how to use the Image Captioning capability of Apache Tika. "Image captioning" or "describing the content of an image" is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. TIKA-2262 introduced a new parser to perform captioning on images. Visit TIKA-2262 issue on Jira or pull request on Github to see the related conversations. Currently, Tika utilizes an implementation based on the paper Show and Tell: A Neural Image Caption Generator for captioning images. This paper presents a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation that can be used to generate natural sentences describing an image. Continue reading to get Tika up and running for image captioning.

Tika and Tensorflow Image Captioning Using REST Server

We are going to start a python flask based REST API server and tell tika to connect to it. All the dependencies and setup complexities are isolated in the docker image.

Requirements:

Docker -- Visit Docker.com and install latest version of Docker. (Note: tested on docker v17.03.1)

Step 1. Setup REST Server

You can either start the REST server in an isolated docker container or natively on the host that runs tensorflow v1.0

a. Using docker (Recommended)

```
three
1 git clone https://github.com/USCDataScience/tika-dockers.git && cd tika-dockers
2 docker build -f Im2txtRestDockerfile -t uscdatascience/im2txt-rest-tika .
3 docker run -p 8764:8764 -it uscdatascience/im2txt-rest-tika
```


Sample output from API:
Note: MAC USERS:

If you are using an older version, say, 'Docker toolbox' instead of the newer 'Docker for Mac',
you need to add port forwarding rules in your Virtual Box default machine.

1. Open the Virtual Box Manager.
2. Select your Docker Machine Virtual Box image.
3. Open Settings -> Network -> Advanced -> Port Forwarding.
4. Add an appname, Host IP 127.0.0.1 and set both ports to 8764.

b. Without Using docker

If you chose to setup REST server without a docker container, you are free to manually install all the required tools specified in the docker file.

Note: docker file has setup instructions for Ubuntu, you will have to transform those commands for your environment.

**Step 2. Create a Tika-Config XML to enable Tensorflow parser.**

A sample config can be found in Tika source code at tika-parsers/src/test/resources/org/apache/tika/parser/recognition/tika-config-tflow-rest.xml

Here is an example:

```xml
<properties>
    <parsers>
        <parser class="org.apache.tika.parser.recognition.ObjectRecognitionParser">
            <mime>image/jpeg</mime>
            <mime>image/png</mime>
            <mime>image/gif</mime>
            <params>
                <param name="apiBaseUri" type="uri">http://localhost:8764/inception/v3</param>
                <param name="captions" type="int">5</param>
                <param name="maxCaptionLength" type="int">15</param>
                <param name="class" type="string">org.apache.tika.parser.captioning.tf.TensorflowRESTCaptioner</param>
            </params>
        </parser>
    </parsers>
</properties>
```

**Description of parameters:**

<table>
<thead>
<tr>
<th>Param Name</th>
<th>Type</th>
<th>Meaning</th>
<th>Range</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiBaseUri</td>
<td>uri</td>
<td>HTTP URL that will be used to create apiUri &amp; healthUri</td>
<td>any HTTP URL</td>
<td><a href="http://localhost:8764/inception/v3">http://localhost:8764/inception/v3</a></td>
</tr>
</tbody>
</table>
### captions
- **int** Number of captions to output
  - a non-zero positive integer
  - 3 to receive 3 captions

### maxCaptionLength
- **int** Maximum length of a caption
  - a non-zero positive integer (recommended >=15)
  - for 15 the sentence length of a caption won't be greater than 15

<table>
<thead>
<tr>
<th>class</th>
<th>string</th>
<th>Name of class that Implements Object recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>org.apache.tika.parser.recognition.tf.TensorflowRESTCaptioner</td>
</tr>
</tbody>
</table>

### Step 3. Demo

```bash
$ java -jar tika-app/target/tika-app-1.17-SNAPSHOT.jar \
--config=tika-parsers/src/test/resources/org/apache/tika/parser/recognition/tika-config-tflow-im2txt-rest.xml \ 
```

The input image is:

*blocked URL*

And, the output is

**Toggle line numbers**

```html
<ol id="captions">
  <li id="0">
    a man standing next to a dog on a leash . [en](confidence = 0.000167)
  </li>
  <li id="1">a man standing next to a dog on a bench . [en](confidence = 0.000167)</li>
  <li id="2">a man and a dog are sitting on a bench . [en](confidence = 0.000138)</li>
  <li id="3">a man and a dog sitting on a bench . [en](confidence = 0.000131)</li>
  <li id="4">a man and a dog are sitting on a bench . [en](confidence = 0.000092)</li>
</ol>
```

### Questions / Suggestions / Improvements / Feedback ?

1. If it was useful, let us know on twitter by mentioning @ApacheTika
2. If you have questions, let us know by using Mailing Lists
3. If you find any bugs, use Jira to report them