PERFORMANCE EVALUATION PROTOCOL FOR TEXT RECOGNITION IN VIDEO ANALYSIS AND CONTENT EXTRACTION (VACE-II)

DRAFT - 1.1

Submitted to
Advanced Research and Development Activity

Technical Monitors: Terrence Adams, John Garofolo, John Prange

by

Rangachar Kasturi, Professor
Phone: (813) 974-3561, Fax: (813) 974 5456
Email: r1k@csee.usf.edu
Dmitry Goldgof, Professor
Padmanabhan Soundararajan, Postdoctoral Fellow
Vasant Manohar, Student
Matthew Boonstra, Student

Computer Science & Engineering
University of South Florida, ENB 118
4202 E. Fowler Ave
Tampa, FL 33620-5399

Date: April 18, 2005

Contents

1 Introduction

This evaluation focuses on technologies developed to recognize the content of text objects in video. While detection and tracking are necessary, they are not evaluated in this task. The text detection and tracking tasks will occur in the VACE core evaluations, and contributors to the text recognition task will not be scored on text position and tracking information. However, contributors will be responsible for both detecting and recognizing the text in a sequence, but will only be scored on their recognition performance.

This document defines the task and evaluation protocol for a portion of the 2004-2005 VACE evaluations. The specific task this document supports is text recognition in the broadcast news and surveillance domains.

2 Text Recognition Task

The goal of the text recognition task is to recognize text objects in a video sequence. This task does not require the system to track these text blocks in a video frame; that part of the task is relegated to the text detection and tracking portion of the VACE framework. The text will be annotated at the word level according to the annotation guidelines.

The performance of the task will be scored at the frame level and will be based on how accurate the system recognizes the characters in each word in the frame. The system output tags must be generated according to the rules specified in the annotation guidelines and are to be formatted as described in Section 8. The text is transcribed at the word level. Text which is annotated as unevaluable by the evaluators and annotators will not be evaluated. To keep things tractable in the first cycle of evaluations, only alpha-numeric characters will be considered, capitalization and word-external punctuation will be filtered from both the system output and reference transcripts. Word-internal punctuations such as hyphens and apostrophes will not be filtered. Also, line breaks constitute word boundaries, so wrapped words will be treated as separate text tokens. At a higher level, special cases which will not be evaluated are:

1. Scrolling text.
2. Dynamic Text (see Ref [2] for definitions).
3. Reference Text with Readability Levels Greater Than 1.

For this particular task, annotation tags will include:

1. Video Filename.
2. Object id (unique for the frame).
3. BBox location parameters upper left corner, height, width and rotation attributes .
4. The transcription of each word (each BBox contents).

3 Source Data

Unless specified otherwise, the provided video sequences will be in MPEG-2 format with a resolution of 720x480 NTSC based 29.97 frames per second or 704x480 (DVD). The chrominance format is 4:2:0 and the I (Intra), P (Predicted) and B (Bi-directional Predicted) encoding sequence can be of length of 12 or 15. The ground truth is a plain text file in XML format with the required tags. Each sequence will be in the range of 1 to 4 minutes long. While all of the video frames will be provided, only the I-frames will be annotated and evaluated. This will allow for a greater amount of video data to be annotated for the evaluation than would otherwise be possible, and with minimal loss of information.

3.1 Scope

The dataset includes data from the TV broadcast news domain. Tentatively, the plan is also to include surveillance video data.

3.2 VACE-II 2004-2005 Datasets

This section describes the dataset to be developed to support the 2004-2005 evaluations. A complete set of training and test data that will be supported for each task and domain are shown in Table 1 for the data statistics which include the planned breakdown of Micro Corpus/Training/Evaluation data. The sequences and times shown are estimates and could change based on data availability and annotation complexity.

	DATA	NUMBER OF SEQUENCES	TOTAL MINUTES	AVERAGE MINUTES PER SEQUENCE
PER DOMAIN	MICRO-CORPUS	5	10	–
	TRAINING	5	10	2.5
	TESTING	25	62.5	2.5

Table 1: VACE-II Corpus Partitioning for the Text Recognition Task.

For the 2005 evaluation, given the available resources and time, the following core tasks/domains will be supported with annotated data as indicated in Table 2.

TASK	**DOMAIN**			
	Meeting Room NIST Meeting Room Project	Broadcast News LDC Broadcast (ABC & CNN)	UAV *(Pending availability)*	Surveillance *(Pending availability)*
Text Detect & Track (English)	–	Y	–	?
Text Detect & Track (Chinese)	–	Y	–	?
Text Detect & Track (Arabic)	–	Y	–	?
Text Recognition (English)	–	Y	–	?
Text Recognition (Chinese)	–	–	–	?
Text Recognition (Arabic)	–	Y (2006)	–	?

Table 2: Task Versus Domain Support Matrix (– =No, Y=Yes, ? = Unsure).

3.3 Permitted Side Information

The following information for each domain will be available to the systems in performing the tasks. No other side information should be used. **TBD**

3.4 Formats

As an expedient for this year the ViPER native format will be used for both the system output and reference annotations. Both the input and output files will contain the tags required for evaluation. An example XML file produced by ViPER is shown in Appendix A.

4 Performance Assessment

This section and the following sections will address how the output of the research systems will be evaluated. For the VACE-II evaluations, we have defined a single measure for the text recognition task using a count of the number of insertions, deletions, and substitutions and weights for each class of error. The metric for recognition is called the Text Recognition Accuracy measure (**TRA**) and is described in Section 5. This measure will be considered the primary measure for the text recognition evaluations. It will provide not only a summative measure of the performance of the systems, but will also provide the researchers with a focused tool to use in developing and improving their systems.

Before proceeding further, let's define the terms we will use in describing the performance measure:

1. **Text Object** - the entity of interest (word, in the future, lines, sentences, and paragraphs)
2. **Object class** - a constrained set of objects (e.g. caption text, etc.)
3. **Output box** - a geometric shape produced by an algorithm as a result of detection
4. **Measure** - a formula for measuring an algorithm's performance after an experiment

4.1 Handling Limitations in the Ground Truth

Sometimes we want to exclude certain frames from evaluation because they contain frame-level events which place them outside of the scope of the task. An example of this is that the existence of a crowd of faces in a sequence of frames precludes the annotation of particular faces during those frames for the face detection task. To address this issue, **Don't Care Frames (DCFs)** will be established prior to scoring the test results using information in the reference annotation. In our face detection example, particular frames would be annotated in the reference as containing crowds and would not contain further facial annotations. These frames would need to be excluded from evaluation for the face detection task. The DCFs for each task will be automatically generated using a set of rules applied to the reference annotations for that task. Frames in both the reference and system output which are designated as DCFs will then be automatically ignored by the scoring procedure.

Likewise, sometimes we want to exclude certain objects from the target object class because they contain attributes which place them outside the scope of the task. An example of this is the existence of a synthetic face (cartoon or painting) in a particular frame for the face detection task. To address this issue, **Don't Care Objects (DCOs)** will be established prior to scoring the test results using information in the reference annotations. In our synthetic face example, a face annotated as being synthetic would participate in the one-to-one reference/system-output alignment procedure for the new comprehensive measures, but would not be scored. Therefore, an algorithm would not be penalized for missing the synthetic face, but would also not be rewarded for detecting it. Objects in these DCOs will be effectively treated as not existing in both the reference and system output. Additional secondary diagnostic scoring runs may be made to indicate how well these out-of-scope objects were detected/tracked by turning off certain DCOs[1].

Where **DCOs** are used to annotate objects which can be spatially annotated but which can't be reliably identified, some objects may be too blurry or too difficult to localize and cannot be bounded. To address this problem, **Don't Care Regions (DCRs)** will be used to identify areas in frames which can't be spatially annotated and which are to be eliminated entirely from the mapping and scoring process. Detected objects which fall inside a **DCR** or whose area is contained primarily within a **DCR** will be eliminated prior to the mapping/scoring process and will thus not generate false alarm errors. An example is a region of completely unreadable text which can't be effectively grouped into text boxes.

For all the VACE measures, we assume that the **DCF**s and **DCO**s have been removed from both the ground-truth and the algorithm's output prior to the scoring process.

[1] An additional example of a DCO is text classified as of poor quality for the text detection and tracking tasks.

5 Text Recognition Performance Measure

The performance measure for the recognition task will be based on insertion, deletion and substitutions errors at the word level. The measure requires a unique one-to-one mapping of ground truth and detected text object using some optimization (see Appendix B). The mapping will be performed using spatial information and under special circumstances, using the Character Error Rate as well (see Section 5.2).

5.1 Formulae

The character error rate (CER) is a count based measure which penalizes insertions (I), deletions (D) and substitution (S) errors for the characters in each word. For a single word w, we define the CER(w) below, given NT(w), the total number of reference tokens (characters) in each reference text object.

$$CER(w) = \frac{(I+S+D)}{NT(w)} \tag{1}$$

The numerator of the character error rate is more commonly known in the literature available for string matching as the edit distance. The edit distance is defined as a count of the number of character insertions (I), deletions (D), and substitutions (S) to get from one string to another. The operation is kept case insensitive. For example, as you can see from Figure 5.1, *Raven* has an edit distance of 4 to get to *Crone*, which represents one insertion (a leading *C*), two substitutions (changing *av* to *on*) and one deletion (the *n* at the end of *Raven*).

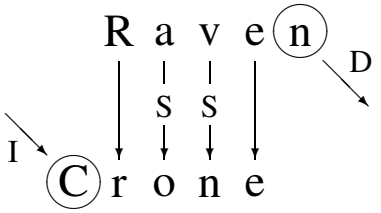

Figure 1: Edit Distance Example 1

As shown in Figure 5.1, the edit distance between the strings *available* and *cavilabte* would be 3. One substitution, one insertion, and one deletion would be necessary to change the second string into the first string. The operations would be to remove the leading *c*, add an *a* between the *v* and *i* and substitute an *l* in place of the *t*.

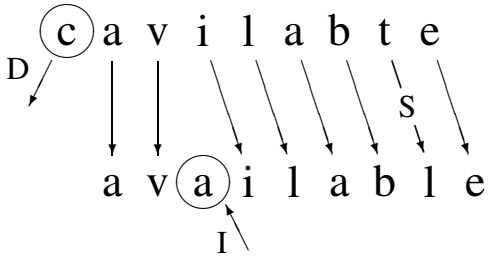

Figure 2: Edit Distance Example 2

The algorithm used to calculate the edit distance in our software was developed by Wagner and Fisher and is described in [5].

The word error rate (WER) is very similar to the character error rate (CER) defined above. The word error rate (WER) is a count based measure which penalizes insertions (I), deletions (D) and substitution (S) errors for the words in each frame. For a single frame t, we define the WER(t) below.

$$WER(t) = \frac{1}{NW(t)}(w_i I + w_s S + w_d D), \tag{2}$$

where NW(t) is the number of reference words in frame t. For the VACE evaluations, the weights w_i, w_s, and w_d are used to give more leniency to a certain class of error, or punish a class of error more harshly. The sum of the weights should be 3.

Since WER(t) is the error measure, we can calculate the word accuracy rate (WAR) as shown,

$$WAR(t) = 1 - WER(t) \tag{3}$$

We can compute the average recognition performance measure (ARPM) for the entire sequence as,

$$ARPM = \frac{1}{TNW(s)} \sum_{t=1}^{n} NW(t)WAR(t), \tag{4}$$

where NW(t) is the number of reference words in frame t and TNW(s) is the total number of reference words in the sequence.

For the VACE evaluations, system outputs will be scored using the average recognition performance measure (ARPM) as described above.

5.2 Mapping

The text recognition scoring software performs a one-to-one mapping on words from the system output to words in the ground truth annotation. The distance between the bounding box centroid for each ground truth and system output object is used to form the score matrix of the mapping algorithm. The mapping algorithm is then performed to find the one-to-one mapping which is used by the text recognition scoring software.

In some special cases, a new mapping is performed which includes not only the centroid distance, but also the character error rate of the system output word as compared to the ground truth word. The special cases where this combined measure is used are the following: when two system output words are at an equal minimum distance to a ground truth word and when the closest system output word from the ground truth is not selected by the mapping algorithm. The centroid distance and the character error rate are each normalized to the range from 0 to 1 inclusive, and then multiplied by the mapping weight assigned to that measure. Currently, equal weights are used, but a command line option has been implemented which allows the user to select different weights. The mapping algorithm is performed again with the new scores, and the resulting one-to-one mapping is used by the text recognition scoring software.

6 Candidate Tasks for Future Text Recognition Evaluations

In addition to the task described in this document, there are several other tasks of a more exploratory nature which are of interest and may be pursued in future text recognition evaluations. These are discussed in Section 6.1 through Section 6.1. These tasks are likely candidates for the VACE program to pursue in its future core evaluations. However, this list may be expanded as the community suggests additional such tasks of interest.

6.1 String of Words as a Text Object

This is similar to the basic text recognition task, but with the text object defined as a string of words. One could define word substitution, insertion, and deletion and use these in different evaluation measures which show system performance on entire blocks with semantic ordering.

7 Reference Annotations

The Video Performance Evaluation Resource (ViPER) [1] was developed as a tool for ground-truthing video sequences and will be used to create the reference annotations for this evaluation. Objects are marked by bounding box parameters. The objects are annotated in ViPER XML format. The ground truth annotation instructions for the text recognition task can be found in the companion annotation guidelines document [2].

8 System Input/Output

The system output is to be in ViPER XML format using the tags specified in the task definitions. Note that, the reference will be richly annotated with a variety of information some of which is intended for data selection and analysis only. Therefore, not all the annotated information will be used for evaluation. The proposed file naming conventions are as shown below,

FILENAME EXTENSION	DESCRIPTION
*.gtf	Ground Truth File
*.rdf	Result File
*.ndx	Index File
*.sysinfo	System Information File

Table 3: File naming conventions.

8.1 System Input Data (Training/Testing)

The input data will be in MPEG-2 format as indicated earlier. The data will be presented to the research systems in multiple sequences varying in duration from 1–4 minutes. The video clips for each task will be present in a separate directory. An index file will exist for each task and will follow the naming conventions as explained below.

Year_Purpose_Domain_Task.ndx

where,

- *Year* specifies the year in which the evaluation would take place
- *Purpose* can be (Train, Test)
- *Domain* can be (BNews, Surveillance)
- *Task* can be (TREng, TRChin, TRArab)

Also, the index file will contain the following details.

- Sequence-ID
- Source Path/Filename
- Begin-frame
- End-frame
  where,
  - Sequence-ID is the input sequence ID which can take values (1 ... N_{seq})
  - Source Path/Filename is the path and filename of the original file from which the clip was extracted

- Begin-frame is the frame number in the original source file when the clip begins
- End-frame is the frame number in the original source file when the clip ends

Thus, together with the index filename and the information present in the file, we can uniquely identify a video clip and its original source file. Based on the *Sequence-ID*, we can map back to the original file with the details in the corresponding index file. The ground truth XML file will be present in the same directory, with the following naming convention.

Year_Purpose_Domain_Task_Sequence-ID.gtf

Each individual XML file will contain a header listing the tags used in the file and their possible values. For convenience a copy of the XML headers will be included in a separate *config* file.

An example config file and an associated example XML file is as shown below,

```
#BEGIN_CONFIG
FILE Information
      SOURCEDIR : svalue [static]
      SOURCEFILES : svalue [static]

OBJECT Text
      TYPE : lvalue [static] [ SCENE GRAPHIC ]
      READABILITY : lvalue [static] [ 1 2 3 ]
      BBOX : svalue [static]
      CONTENTS : svalue [static]
      NCHARS : dvalue [static]

#END_CONFIG
```

and an associated XML file,

```
/*----------------------XML File Begin----------------*/
<sourcedir="  "/>
<sourcefile="  "/>

<descriptor name="Text" type="SCENE" ID="1">
<attribute Contents="String Content"  readability="3"/>
       <data:lvalue ORIENT="x"/>
       <default>
           <data:lvalue ORIENT="0"/>
       </default>
       BBox="10 19 45 67"
       NChars=4
</attribute>
/*----------------------XML File End----------------*/
```

The format of these files are task dependent. For a face detection and tracking task, a config file and its corresponding example XML file will appear as shown below,

```
#BEGIN_CONFIG
FILE Information
      SOURCEDIR : svalue [static]
      SOURCEFILES : svalue [static]

OBJECT Face
      TYPE : lvalue [static] [ FULL PROFILE ]
      BBOX : svalue [static]
      DESCRIPTION : svalue [static]
#END_CONFIG
```

```
/*----------------------XML File Begin----------------*/
<sourcedir="  "/>
<sourcefile="  "/>

<descriptor name="Face" type="FULL" ID="11">
<attribute Description="Facing Camera directly"/>
       BBox="50 19 75 25"
</attribute>
/*----------------------XML File End----------------*/
```

8.2 System Output Data

The primary submission from each site should use the equal error rate operating point setting for each algorithm/task combination. Sites should also indicate primary submission versus contrast ones.

The system output will be an XML based file. For an input sequence *Year_Purpose_Domain_Task_Sequence-ID*, the corresponding XML based output file should be named as *Site_System_Year_Purpose_Domain_Task_Sequence-ID_Run-ID.rdf*

where,

- *Site* is a terse site ID
- *System* is a terse system name
- *Year* specifies the year in which the evaluation would take place
- *Purpose* can be (Train, Test)
- *Domain* can be (BNews, Surveillance)
- *Task* can be (TREng, TRChin, TRArab)
- *Sequence-ID* is the input sequence ID which can take values (01, 02, ... N_{seq})
- *Run-ID* can take values (01, 02, ... N_{run}) **(Also, Primary submission should always be 01)**

The description tags provided in this section are comprehensive to all tasks. However, only a subset of the tags relevant to each task are to be provided as specified in Section 2.

The common and specific tags that should be provided by the systems are (note that some of these can be copied from the annotation),

1. Filename of the video sequence.
2. Object ID.
3. Obox/Bbox specification (Obox if the box is oriented).
   (a) rotation in degrees (if Obox specified).
4. Frame number/Framespan.
5. Text Object Contents

The algorithm is to output along with the frame numbers, the box left hand corner co-ordinate parameters, the height and width of the box similar to the annotation style of defining the bounding box parameters, and, most importantly, the recognized contents of the box. An example of the expected system output is as shown below. The example scenario is that there is a text block detected from frames 13 through 80. The box is given the id equal to 1 and the location parameters are defined as shown in the attribute details.

```
<?xml version="1.0" encoding="UTF-8"?>
<viper xmlns="http://lamp.cfar.umd.edu/viper" xmlns:data="http://lamp.cfar.umd.edu/viperdata">
        <config>
                <descriptor name="Text" type="OBJECT">
                        <attribute name="LOCATION" dynamic="true" type="bbox"/>
                        <attribute name="KEY" dynamic="false" type="svalue"/>
            <attribute name="ID" dynamic="true" type="value"/>
                </descriptor>

        </config>
        <data>
```

```
        <sourcefile>
                <object name="Text" id="1" framespan="13:80">
                        <attribute name="LOCATION">
                                <data:bbox x="20" y="112" width="64" height="20" />
                                <data:bbox x="20" y="112" width="68" height="20" />
                                <data:bbox x="28" y="112" width="64" height="20" />
                                <data:bbox x="64" y="108" width="32" height="24" />
                                <data:bbox x="64" y="104" width="32" height="28" />
            .
            .
                        </attribute>
                        <attribute name="KEY">
                                <data:svalue value="A" />
                        </attribute>
                        </attribute name="CONTENTS">
                                <data:svalue framespan="13:80" value="ViPER" />
                        </attribute>
                </object>
    </data>
</xml>
```

9 Required System Information

For each test run, a brief description of the system (algorithms, data, configuration) used to produce the system output must be provided along with your system output.). The system description information is to be provided in a file named: *Site_System_Year_Purpose_Domain_Task_Sequence-ID_Run-ID.sysinfo* and placed in the directory alongside the similarly-named directories containing your system output. This file is to be formatted as follows:

1. Site name
2. System Identifier/Name and version
3. Submitter (contact Name and email)
4. System Description:
   (a) Overview (high-level overview of system approach and configuration)
   (b) Features (description of pertinent system features)
   (c) Relationship to other runs (if this a comparative experiment, what other runs are related)
   (d) Configuration (particular configuration for this run)
   (e) Training (what training data was used and how was it employed)
   (f) Source Data Processing (how was the test data processed)
   (g) Equipment (what hardware was used, # of processors, type of processor, real and virtual memory, OS)
   (h) Processing Speed (what is the Speed Factor for this run as defined in Section 9.1)
   (i) Notes (any other notes regarding this system/run)
5. References: [list pertinent references]

9.1 Processing Speed Computation

The processing speed for each system run should be calculated as specified below and cited in the System Information file for the experiment. These are compulsory details that have to be reported in the system description for each submitted run.

9.1.1 Total Processing Time (TPT)

The time to be calculated is the Total Processing Time (TPT) that it takes to process all parallel streams of recorded video provided (including ALL I/O) on a single CPU. TPT represents the time a system would take to process the recorded video input and produce the specified metadata output as measured by a stopwatch. So that research systems that aren't completely pipelined aren't penalized, the "stopwatch" may be stopped between (batch) processes.

Note that TPT may exclude time to "warm up" the system prior to loading the test recordings (e.g., loading models into memory.)

9.1.2 Source Signal Duration (SSD)

In order to calculate the realtime factor, the duration of the source signal recording must be determined. The source signal duration (SSD) is the actual recording time for the video audio used in the experiment. This time is stream-independent and should be calculated across all video streams for multi-view recordings. It is therefore the wall-clock duration of the period of recording (even if multiple simultaneous recordings were used).

9.1.3 Speed Factor (SF) Computation

The speed factor (SF) (also known as "X" and "times-realtime") is calculated as follows:

$$SF = \frac{TPT}{SSD}$$

For example, a 1-hour news broadcast processed in 10 hours would have a SF of 10. And 5 minutes of surveillance video collected on 2 cameras simultaneously each processed in 30 minutes would have an SF of 12.

9.2 Reporting Your Processing Speed Information

Although we encourage you to break out your processing time components into as much detail as you like, you should minimally report the above information in the system description for each of your submitted experiments in the form:

- TPT = <FLOAT>
- SSD = <FLOAT>
- SF = <FLOAT>

10 Submission Instructions

The system output XML files along with the corresponding System Information Files are to be *tar-ed* and then *gzipped*. For example, if the input sequences considered are 2005_Test_BNews_TDEng_1, 2005_Test_BNews_TDEng_2 and 2005_Test_BNews_TDEng_3, then

- The algorithm will use these sequences and output its results into a single XML file for each sequence in the same corresponding directory. Output file name should follow the file naming protocol presented in Section 8.2.
- Assume that the current working directory has all the sequences that the algorithm output is expected, all the XML files can then be compressed into a single file for submission by using the command,
  *tar -cvf Site_System_2005_Test_BNews_TDEng_Run-ID.tar *.rdf* then,
  *tar -rvf Site_System_2005_Test_BNews_TDEng_Run-ID.tar *.sysinfo* followed by
  gzip -9 Site_System_2005_Test_BNews_TDEng_Run-ID.tar which results in the file *Site_System_2005_Test_BNews_TDEng_Run-ID.tar.gz*.

11 Schedule

The following is a draft schedule working backward from a Sep 2005 workshop report.

Event	Date
Receive Micro Corpus Data	February 25, 2005
Release Draft Protocol	April 1, 2005
Release Micro-corpus	April 11, 2005
Comments from Participants	April 16, 2005
Release Final Protocol, Release Revised Microcorpus	April 26, 2005
Release Scoring Software	May 26, 2005
Release Annotated Training Data	June 5, 2005
Begin Dry Run Evaluation	June 5, 2005
Dry Run Results Due	June 15, 2005
USF Releases Dry Run Scores	June 17, 2005
Release Test Data	August 16, 2005
Results Due from Participants, Complete Annotation of Evaluation Data	August 31, 2005
Release Scores and Reference Annotations to Participants	September 15, 2005
Release Preliminary Report, Workshop Presentation	October 16, 2005

Table 4: Event Schedule (2004–2005).

A APPENDIX: Sample Annotation XML file

```
/*----------------------------------------BEGIN_OF_CONFIG----------------------------------------*/
<?xml version="1.0" encoding="UTF-8"?> <viper
xmlns="http://lamp.cfar.umd.edu/viper#"
xmlns:data="http://lamp.cfar.umd.edu/viperdata#">
    <config>
        <descriptor name="Information" type="FILE">
            <attribute dynamic="false" name="SOURCETYPE" type="http://lamp.cfar.umd.edu/viperdata#lvalue">
                <data:lvalue-possibles>
                    <data:lvalue-enum value="SEQUENCE"/>
                    <data:lvalue-enum value="FRAMES"/>
                </data:lvalue-possibles>
            </attribute>
            <attribute dynamic="false" name="NUMFRAMES" type="http://lamp.cfar.umd.edu/viperdata#dvalue"/>
            <attribute dynamic="false" name="FRAMERATE" type="http://lamp.cfar.umd.edu/viperdata#fvalue"/>
            <attribute dynamic="false" name="H-FRAME-SIZE" type="http://lamp.cfar.umd.edu/viperdata#dvalue"/>
            <attribute dynamic="false" name="V_FRAME_SIZE" type="http://lamp.cfar.umd.edu/viperdata#dvalue"/>
            <attribute dynamic="false" name="V-FRAME-SIZE" type="http://lamp.cfar.umd.edu/viperdata#dvalue"/>
        </descriptor>
        <descriptor name="Frame" type="OBJECT">
            <attribute dynamic="true" name="Evaluate" type="http://lamp.cfar.umd.edu/viperdata#bvalue"/>
            <attribute dynamic="true" name="Crowd" type="http://lamp.cfar.umd.edu/viperdata#bvalue"/>
            <attribute dynamic="true" name="Multiple text" type="http://lamp.cfar.umd.edu/viperdata#bvalue"/>
        </descriptor>
        <descriptor name="Face" type="OBJECT">
            <attribute dynamic="true" name="Location" type="http://lamp.cfar.umd.edu/viperdata#obox"/>
            <attribute dynamic="false" name="Synthetic" type="http://lamp.cfar.umd.edu/viperdata#bvalue"/>
            <attribute dynamic="true" name="Visible" type="http://lamp.cfar.umd.edu/viperdata#bvalue"/>
            <attribute dynamic="true" name="Headgear" type="http://lamp.cfar.umd.edu/viperdata#bvalue">
                <default>
```

```
                <data:bvalue value="false"/>
            </default>
        </attribute>
    </descriptor>
    <descriptor name="Text" type="OBJECT">
        <attribute dynamic="true" name="location" type="http://lamp.cfar.umd.edu/viperdata#obox"/>
        <attribute dynamic="true" name="Readability" type="http://lamp.cfar.umd.edu/viperdata#dvalue">
            <default>
                <data:dvalue value="2"/>
            </default>
        </attribute>
        <attribute dynamic="true" name="Occlusion" type="http://lamp.cfar.umd.edu/viperdata#bvalue">
            <default>
                <data:bvalue value="false"/>
            </default>
        </attribute>
        <attribute dynamic="true" name="Content" type="http://lamp.cfar.umd.edu/viperdata#svalue"/>
        <attribute dynamic="false" name="Type" type="http://lamp.cfar.umd.edu/viperdata#dvalue">
            <default>
                <data:dvalue value="0"/>
            </default>
        </attribute>
        <attribute dynamic="true" name="DCR" type="http://lamp.cfar.umd.edu/viperdata#bvalue">
            <default>
                <data:bvalue value="false"/>
            </default>
        </attribute>
        <attribute dynamic="false" name="Logo" type="http://lamp.cfar.umd.edu/viperdata#bvalue">
            <default>
                <data:bvalue value="false"/>
            </default>
        </attribute>
    </descriptor>
    <descriptor name="I-Frames" type="OBJECT"/>
</config>
/*----------------------------------------END_OF_CONFIG----------------------------------------*/


/*----------------------------------------BEGIN_OF_DATA----------------------------------------*/
<data>
    <sourcefile filename="file:/19980209_1830_1900_CNN_HDL.mpg">
        <file id="0" name="Information">
            <attribute name="SOURCETYPE"/>
            <attribute name="NUMFRAMES">
                <data:dvalue value="49096"/>
            </attribute>
            <attribute name="FRAMERATE">
                <data:fvalue value="1.0"/>
            </attribute>
            <attribute name="H-FRAME-SIZE"/>
            <attribute name="V_FRAME_SIZE"/>
            <attribute name="V-FRAME-SIZE"/>
        </file>
        <object
            framespan="1:1 13:13 25:25 37:37 49:49 61:61 .. .. 49057:49057 49069:49069 49081:49081 49093:49093"
            id="0" name="I-Frames"/>
        <object framespan="5521:10307" id="0" name="Frame">
            <attribute name="Evaluate">
                <data:bvalue framespan="5521:5604" value="true"/>
                <data:bvalue framespan="5605:5640" value="false"/>
                <data:bvalue framespan="5641:5688" value="true"/>
                <data:bvalue framespan="5689:5689" value="false"/>
                <data:bvalue framespan="5690:7332" value="true"/>
                <data:bvalue framespan="7333:7333" value="false"/>
                <data:bvalue framespan="7334:7548" value="true"/>
```

```
        <data:bvalue framespan="7549:7549" value="false"/>
        <data:bvalue framespan="7550:8208" value="true"/>
        <data:bvalue framespan="8209:8209" value="false"/>
        <data:bvalue framespan="8210:8448" value="true"/>
        <data:bvalue framespan="8449:8460" value="false"/>
        <data:bvalue framespan="8461:8696" value="true"/>
        <data:bvalue framespan="8697:8708" value="false"/>
        <data:bvalue framespan="8709:10307" value="true"/>
    </attribute>
    <attribute name="Crowd">
        <data:bvalue framespan="5521:10307" value="false"/>
    </attribute>
    <attribute name="Multiple text">
        <data:bvalue framespan="5521:8461" value="false"/>
        <data:bvalue framespan="8462:8712" value="true"/>
        <data:bvalue framespan="8713:9156" value="false"/>
        <data:bvalue framespan="9157:9852" value="true"/>
        <data:bvalue framespan="9853:10307" value="false"/>
    </attribute>
</object>
<object framespan="5542:5604" id="0" name="Face">
    <attribute name="Location">
        <data:obox framespan="5542:5544" height="32"
            rotation="3" width="37" x="137" y="117"/>
        <data:obox framespan="5545:5556" height="32"
            rotation="3" width="37" x="134" y="116"/>
        <data:obox framespan="5557:5568" height="34"
            rotation="3" width="37" x="134" y="116"/>
        <data:obox framespan="5569:5580" height="34"
            rotation="3" width="37" x="133" y="114"/>
        <data:obox framespan="5581:5592" height="34"
            rotation="3" width="34" x="133" y="109"/>
        <data:obox framespan="5593:5605" height="34"
            rotation="3" width="34" x="127" y="111"/>
    </attribute>
    <attribute name="Synthetic">
        <data:bvalue value="false"/>
    </attribute>
    <attribute name="Visible">
        <data:bvalue framespan="5542:5605" value="true"/>
    </attribute>
    <attribute name="Headgear">
        <data:bvalue framespan="4000:15000" value="false"/>
    </attribute>
</object>
<object framespan="5542:5604" id="1" name="Face">
    <attribute name="Location">
        <data:obox framespan="5542:5544" height="38"
            rotation="21" width="34" x="340" y="127"/>
        <data:obox framespan="5545:5556" height="38"
            rotation="21" width="35" x="340" y="127"/>
        <data:obox framespan="5557:5568" height="38"
            rotation="15" width="37" x="359" y="122"/>
        <data:obox framespan="5569:5592" height="38"
            rotation="5" width="37" x="377" y="119"/>
        <data:obox framespan="5593:5605" height="36"
            rotation="5" width="37" x="366" y="121"/>
    </attribute>
    <attribute name="Synthetic">
        <data:bvalue value="false"/>
    </attribute>
    <attribute name="Visible">
        <data:bvalue framespan="5542:5605" value="true"/>
    </attribute>
    <attribute name="Headgear">
```

```
            <data:bvalue framespan="4000:15000" value="false"/>
        </attribute>
    </object>
    <object framespan="5542:5604 5641:5688" id="2" name="Face">
        <attribute name="Location">
            <data:obox framespan="5542:5544" height="38"
                rotation="0" width="37" x="524" y="149"/>
            <data:obox framespan="5545:5556" height="38"
                rotation="0" width="37" x="516" y="148"/>
            <data:obox framespan="5557:5568" height="43"
                rotation="0" width="26" x="516" y="148"/>
            <data:obox framespan="5569:5580" height="43"
                rotation="0" width="26" x="520" y="147"/>
            <data:obox framespan="5581:5592" height="43"
                rotation="0" width="26" x="524" y="140"/>
            <data:obox framespan="5593:5605" height="40"
                rotation="0" width="26" x="515" y="139"/>
            <data:obox framespan="5641:5652" height="42"
                rotation="0" width="27" x="6" y="137"/>
            <data:obox framespan="5653:5664" height="40"
                rotation="0" width="27" x="14" y="140"/>
            <data:obox framespan="5665:5676" height="40"
                rotation="0" width="27" x="27" y="145"/>
            <data:obox framespan="5677:5689" height="39"
                rotation="0" width="26" x="22" y="146"/>
        </attribute>
        <attribute name="Synthetic">
            <data:bvalue value="false"/>
        </attribute>
        <attribute name="Visible">
            <data:bvalue framespan="5542:5605" value="true"/>
            <data:bvalue framespan="5641:5689" value="true"/>
        </attribute>
        <attribute name="Headgear">
            <data:bvalue framespan="4000:15000" value="false"/>
        </attribute>
    </object>
..
..
    <object framespan="5545:5652 5654:5664" id="2" name="Text">
        <attribute name="location">
            <data:obox framespan="5545:5653" height="21"
                rotation="0" width="46" x="398" y="403"/>
            <data:obox framespan="5654:5665" height="46"
                rotation="0" width="51" x="397" y="403"/>
        </attribute>
        <attribute name="Readability">
            <data:dvalue framespan="5545:5665" value="2"/>
        </attribute>
        <attribute name="Occlusion">
            <data:bvalue framespan="5545:5665" value="false"/>
        </attribute>
        <attribute name="Content">
            <data:svalue framespan="5545:5665" value="0.37 0.25"/>
        </attribute>
        <attribute name="Type"/>
        <attribute name="DCR">
            <data:bvalue framespan="0:10000" value="false"/>
        </attribute>
        <attribute name="Logo"/>
    </object>
    <object framespan="5545:5652 5654:5664" id="3" name="Text">
        <attribute name="location">
            <data:obox framespan="5545:5653" height="22"
                rotation="0" width="59" x="479" y="404"/>
```

```
        <data:obox framespan="5654:5665" height="45"
            rotation="0" width="58" x="480" y="404"/>
    </attribute>
    <attribute name="Readability">
        <data:dvalue framespan="5545:5665" value="2"/>
    </attribute>
    <attribute name="Occlusion">
        <data:bvalue framespan="5545:5665" value="false"/>
    </attribute>
    <attribute name="Content">
        <data:svalue framespan="5545:5665" value="70.38 63.25"/>
    </attribute>
    <attribute name="Type"/>
    <attribute name="DCR">
        <data:bvalue framespan="0:10000" value="false"/>
    </attribute>
    <attribute name="Logo"/>
</object>
<object framespan="5545:8448 8450:10345" id="4" name="Text">
    <attribute name="location">
        <data:obox framespan="5545:10381" height="17"
            rotation="0" width="84" x="575" y="406"/>
    </attribute>
    <attribute name="Readability">
        <data:dvalue framespan="5545:10381" value="2"/>
    </attribute>
    <attribute name="Occlusion">
        <data:bvalue framespan="5545:10381" value="false"/>
    </attribute>
    <attribute name="Content">
        <data:svalue framespan="5545:10381" value="Headline"/>
    </attribute>
    <attribute name="Type"/>
    <attribute name="DCR">
        <data:bvalue framespan="0:10345" value="false"/>
    </attribute>
    <attribute name="Logo"/>
</object>
<object framespan="5545:8448 8450:10345" id="5" name="Text">
    <attribute name="location">
        <data:obox framespan="5545:10345" height="15"
            rotation="0" width="35" x="576" y="424"/>
        <data:obox framespan="10346:10381" height="15"
            rotation="0" width="32" x="577" y="426"/>
    </attribute>
    <attribute name="Readability">
        <data:dvalue framespan="5545:10381" value="2"/>
    </attribute>
    <attribute name="Occlusion">
        <data:bvalue framespan="5545:10381" value="false"/>
    </attribute>
    <attribute name="Content">
        <data:svalue framespan="5545:10381" value="CNN"/>
    </attribute>
    <attribute name="Type"/>
    <attribute name="DCR">
        <data:bvalue framespan="0:10345" value="false"/>
    </attribute>
    <attribute name="Logo">
        <data:bvalue value="true"/>
    </attribute>
</object>
<object framespan="5545:8448 8450:10345" id="6" name="Text">
    <attribute name="location">
        <data:obox framespan="5545:10381" height="17"
```

```
                        rotation="0" width="47" x="611" y="424"/>
                </attribute>
                <attribute name="Readability">
                    <data:dvalue framespan="5545:10381" value="2"/>
                </attribute>
                <attribute name="Occlusion">
                    <data:bvalue framespan="5545:10381" value="false"/>
                </attribute>
                <attribute name="Content">
                    <data:svalue framespan="5545:10381" value="NEWS"/>
                </attribute>
                <attribute name="Type"/>
                <attribute name="DCR">
                    <data:bvalue framespan="0:10345" value="false"/>
                </attribute>
                <attribute name="Logo"/>
            </object>
..
..
        <object framespan="10141:10333" id="359" name="Text">
                <attribute name="location">
                    <data:obox framespan="10141:10333" height="16"
                        rotation="0" width="27" x="379" y="429"/>
                </attribute>
                <attribute name="Readability">
                    <data:dvalue framespan="10141:10333" value="2"/>
                </attribute>
                <attribute name="Occlusion">
                    <data:bvalue framespan="10141:10333" value="false"/>
                </attribute>
                <attribute name="Content">
                    <data:svalue framespan="10141:10333" value="for"/>
                </attribute>
                <attribute name="Type"/>
                <attribute name="DCR">
                    <data:bvalue framespan="10141:10333" value="false"/>
                </attribute>
                <attribute name="Logo"/>
            </object>
            <object framespan="10141:10333" id="360" name="Text">
                <attribute name="location">
                    <data:obox framespan="10141:10333" height="20"
                        rotation="0" width="91" x="417" y="431"/>
                </attribute>
                <attribute name="Readability">
                    <data:dvalue framespan="10141:10333" value="2"/>
                </attribute>
                <attribute name="Occlusion">
                    <data:bvalue framespan="10141:10333" value="false"/>
                </attribute>
                <attribute name="Content">
                    <data:svalue framespan="10141:10333" value="governor"/>
                </attribute>
                <attribute name="Type"/>
                <attribute name="DCR">
                    <data:bvalue framespan="10141:10333" value="false"/>
                </attribute>
                <attribute name="Logo"/>
            </object>
        </sourcefile>
    </data>
</viper>
/*----------------------------------------END_OF_DATA----------------------------------------*/
```

B APPENDIX: Matching Strategies

Assume that there are N ground truth objects and M detected objects. There needs to be a best possible match between these objects in a global sense. A brute force algorithm will have an exponential complexity, a result of having to try out all possible combination of matches (n!). However, this is a standard optimization problem and there are standard techniques to get the optimal match. The matching is generated with the constraint that the sum of the chosen function of the matched pairs is minimized or maximized as the case may be. In usual assignment problems, the number of objects in both cases are equal, i.e, when $N = M$. This is not a requirement and unequal number of objects can also be matched.

	DT_1	DT_2	$\ldots$	DT_M
GT_1	x			
GT_2				x
$\vdots$				
GT_N		x		

There are many variations of the basic Hungarian strategy [4] most of which exploit constraints from specific problem domains they deal with. The algorithm has a series of steps which is followed iteratively and has a polynomial time complexity, specifically some implementations have $O(N^3)$. Faster implementations have been known to exist and have the current best bound to be at $O(N^2 logN + NM)$ [3]. In our case, the matrix to be matched is most likely sparse and this fact could be taken advantage of by implementing a hash function for mapping sub-inputs from the whole set of inputs.

References

[1] D. Doermann and D. Mihalcik. Tools and techniques for video performance evaluation. In *ICPR*, volume 4, pages 167–170, 2000.

[2] Harish Raju et al. Viper annotation instructions for the text recognition task (to be written). VACE Text Recognition Annotation Document.

[3] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of ACM*, 34(3):596–615, Jul 1987.

[4] J. R. Munkres. Algorithms for the assignment and transportation problems. *J. SIAM*, 5:32–38, 1957.

[5] R.A. Wagner and M.J. Fisher. The string-to-string correction problem. *Journal of ACM*, 21:168–178, 1974.