The Wordometer – Estimating the Number of Words Read Using Document Image Retrieval and Mobile Eye Tracking

Kai Kunze, Hitoshi Kawaichi, Kazuyo Yoshimura, Koichi Kise
Osaka Prefecture University, Japan
kunze,kawaichi,yoshimura@m.cs.osakafu-u.ac.jp
kise@cs.osakafu-u.ac.jp

Abstract—We introduce the Wordometer, a novel method to estimate the number of words a user reads using a mobile eye tracker and document image retrieval. We present a reading detection algorithm which works with over 91% accuracy over 10 test subjects using 10-fold cross validation. We implement two algorithms to estimate the read words using a line break detector. A simple version gives an average error rate of 13.5% for 9 users over 10 documents. A more sophisticated word count algorithm based on support vector regression with an RBF kernel reaches an average error rate from only 8.2% (6.5% if one test subject with abnormal behavior is excluded). The achieved error rates are comparable to pedometers that count our steps in our daily life. Thus, we believe the Wordometer can be used as a step counter for the information we read to make our knowledge life healthier.

I. INTRODUCTION

To make their life healthier, more and more people are using pedometers. A survey of medical pedometer studies shows that monitoring something as simple as how many steps you take can have a huge impact in health. People wearing pedometers walk in average over 1.5 km more per day, substantially decreasing their risk for heart attacks, type 2 diabetes and other diseases related to obesity [1].

A pedometer is a very simple, yet effective tool to monitor and improve our physical fitness. We want to provide a similar tool for our cognitive fitness: the Wordometer, counting how many words a user reads per day. This paper presents our initial work towards this goal.

Of course, how much we read directly influences the size of our vocabulary and our language skills [2]. However, several studies also indicate that the more people read throughout the day the higher are their general knowledge and critical thinking skills [2], [3]. Interestingly, similar effects could so far not been shown for TV or other video/multimedia consumption (even when focusing on documentaries). Being able to just count the words we read each day could help to assess the general knowledge of a person, as there are strong correlations between the two [3].

Exploring the reading activities of people, in general, and a Wordometer in particular is also an interesting for document analysis, as we would be able to provide more insides into how/when documents are read (e.g. number of words read per page statistics or “this is the most read paragraph/page in this book”).

The main contributions we present in this paper are as follows:

1) We present an algorithm based on eye gazes obtained by a mobile eye tracker to segment reading from not reading.
2) We present a method to segment read lines using the user’s eye gaze and document image retrieval.
3) Based on the line estimates, we developed 2 methods to estimate the number of words a user reads using a mobile eye tracker and document image retrieval.
4) Our best algorithm to estimate word numbers has an error rate of only between 6–8% evaluated in a study with 10 users and 10 documents.

Achieving an error rate of 6-8% for our Wordometer is reasonable comparing it with the pedometer error rate which is between 3-10% [1]. The pedometer studies are by far more representative. Yet, this is the first time to our knowledge somebody tired to estimate the amount of words read and 6-8% seem to be a good error margin for a problem more difficult then counting steps. Section II is devoted to describe our approach to implement the Wordometer. Sections III and IV are for experimental results and discussions. Section V is to sum up what we have learnt from this research activity.

II. APPROACH

We are interested in the number of words read. We could use optical character recognition/ text detection or similar technologies to record all characters a user sees assuming that the amount of text surrounding people is indicative for how many words they read. However, this will only be a rough estimate at best. It is easy to think of scenarios where this utterly fails. Imagine an American tourist in a Japanese city with a lot of billboards, although he’s exposed to a lot of text, the words read are close to zero (assuming he does not understand Japanese). Therefore, we have to monitor the reading –the decoding of letters, words and assigning meaning to them.

As reading is a cognitive process, sensing brain activity seems to provide the most insight. However, directly sensing brain activity can only be done by more or less invasive methods (e.g. electroencephalography, functional magnetic resonance imaging, electrocorticography). On the other hands, eye movements and gaze are strongly correlated with reading and text comprehension [4], [5], [6] and it can be done by relatively unobtrusive already today.

Our approach has several discrete steps:

1) The raw eye gaze data is summarized into fixations and saccades according to Busher et al. [7] (see Figure 1).
Fig. 1. Eye fixations and saccade traces while reading a document.

Fig. 2. Eyetracking data for Reading (left) versus not Reading (right)

2) On basis of the optioned eye gaze data we calculate features and classify reading from not reading.

3) On reading data, we use the video recording from the eye tracker to apply a document image retrieval technique (this is used to filter some head movements and get the average word count by line for the document).

4) A line detection algorithm is applied on the rectified eye gaze data.

5) Using the numbers of lines detected we calculate the words per line using 2 methods: the first by multiplying it with the average words per line of the document, the second by using support vector regression.

A. Reading Detection

There are already a couple of methods in the related work [8], [9]. However, they work with different sensing modalities (e.g. electrooculography) and eye tracking hardware. Therefore we decided to implement our own algorithm.

The process is straight forward: we calculate eye gaze features given in Table I over a 3sec. frame sliding window and apply a Support Vector Machine classifier with a radial basis function on the resulting feature vector.

B. Applying document image retrieval

With the help of an eyetracker and document image retrieval, we are able to obtain information on the document the user is reading as well as his/her eye gaze information.

We use a document image retrieval method based on “Locally Likely Arrangement Hashing” (LLAH) to associate the paper with the digital document [10]. We use the video feed from the eye tracker as input to LLAH. LLAH retrieves the corresponding page from a document image database by comparing feature points. The eye gaze data obtained by the eye tracker is represented in the coordinate system of a scene camera of the eyetracker. We use LLAH to convert the eye gaze coordinates to the corresponding coordinates of the document by using a homography estimate.

Figure 3 illustrates an example of an eye gaze converted from a camera captured image (left) to its corresponding retrieved document image (right). On the left, the rectangle represents the focused area in which the corresponding document is retrieved. With this method we avoid potential confusion caused by other documents also captured in the image. By using this capability, we can record eye gazes on the coordinate system of a retrieved document (see Figure 4 for an example).

C. Detection of Line Breaks

Figure 5 shows a typical eye movement when a person reads over line breaks. In our proposed method, we segment a sequence of eye movements into parts corresponding to text lines by analyzing them on the coordinate system of a retrieved document.

The eye movement for a line break is against the regular reading order from the end of the text line backwards to the start of the new line. We detect such a movement using the following line break detection.

Let \( (g_1, ..., g_l) \) represent a subsequence of gaze points. A current gaze \( g_l \) is recognized as a line-break if the following conditions are satisfied:

1) the area of a rectangle that circumscribes the \( l \) gazes \( g_1, ..., g_l \) is larger than a certain threshold,

2) the direction of eye movements from the \( l - k \) th to the \( l \) th gaze is opposite. Some succeeding gazes are also included as a part of a line break.

<table>
<thead>
<tr>
<th>Table I. Features for Reading Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>feature</td>
</tr>
<tr>
<td>fixation related features</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>saccade related features</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>wavelet</td>
</tr>
</tbody>
</table>

Fig. 3. Conversion of an eye gaze by using LLAH.
Fig. 4. Two examples of an eye gaze conversion by using LLAH. The right sample works very well, in the left sample an offset is introduced.

Fig. 5. Example of eye gaze data from a user reading over a line break

D. Wordometer

Next, we apply our line break detector to build the Wor-
dometer.

In general, there are many ways to implement a Wordome-
ter. If we are able to record all read words accurately, it is
trivial to realize the Wordometer based on the record. However,
it is not easy to achieve such an accuracy even with the help of
document image retrieval. If we are interested in just counting
the rough number of read words, it is not necessary to have
high accuracy; some errors are acceptable just as is the case
of pedometers.

In this research we implemented two methods for a Wor-
dometer based on the line break detector as follows.

First we build a very simple method. Let $N$ be the average
number of words in one line of a page the user is reading, and
$L$ be the number of estimated line breaks. Note that $N$ can be
obtained by using document image retrieval. The number of
read words is estimated as $N(L+1)$. The more errors we have
in the number of estimated line breaks, the estimated number
of read words gets more inaccurate.

The second method is more sophisticated. In order to
improve the accuracy, we employ a feature vector $x$ that
represents eye movement information and estimate the number
of read words by a function $f(N, L, x)$, which is learnt by
using the learning samples. As a learner, we employ the SVR
(support vector regression). In the following this method is
called the SVR-based method.

Table II shows features in the vector $x$ we obtain from eye
movement data.

III. EXPERIMENTS

We asked 10 subjects to read 10 pages of documents
written in English and record their eye movements by using the
SMI iViewX ETG (Eye Tracking Glasses), see Figure 6. Among
the subjects, 7 are with unaided vision, 2 are with contact lenses
on both eyes, and one with a contact lens on a single eye. Documents are from tests of PET (Preliminary En-
lish Test) in ESOL (English for Speakers of Other Languages)
by University of Cambridge. For each document we have a
time limit defined based on the number of words included in
it.

For the reading detection we designed another experiment
including also the 10 subjects reading 10 documents from a
job hunting test. Other than that the conditions are the same.
The not reading activity included looking around in the room,
playing with objects (e.g. cellphone etc.).

The following is the procedure of experiments. Each sub-
ject was requested to wear the eyetracker for calibration. Then
he/she is requested to read a document placed on a desk in
front of him/her and with a natural posture. The calibration of
the eyetracker is applied after reading each document. Then
we applied $k$-fold cross validation with $k$ subjects.

IV. RESULTS AND DISCUSSION

A. Reading Detection

Reading versus not reading can be easily distinguished just
by visualizing the the eye gaze data for the different classes, as
depicted in Figure 2. As expected the reading detection works
reliably well with an average of 91 % over the 10 subjects,
see Table III for details.

B. Line break estimation

One test subject with unaided vision often fell asleep during
the data recording and his recorded data showed a lot of
inconsistencies. Therefore, we decided to exclude him and the
data from the remaining 9 subjects are utilized for 9-fold cross
validation.

Table IV shows errors observed during the line break
detection. The average errors of the estimated number of lines
is 2.1 lines per document. Since the average number of lines
of documents is 17, we are able to estimate the number of
lines with 12.4% errors.

From the Table IV , we can observe that the results by the
subject $h$ are much worse than other results. His data includes
more frequent eye blinks than any other subjects. In addition,
he often re-read the end of the line just after the line break.
This behavior erroneously increased the number of detected line breaks.

Another source of errors are short lines. The proposed method tends not to detect such short lines so that the resultant estimated number is smaller than the groundtruth. The reason is two-fold:

1) The detection is based on the area of a bounding box of gazes. Since short lines only have small number of gazes and thus the area is sometimes too small.

2) For most subjects, the duration of gazing a short line is too small.

The results for document C are clearly worse compared to other documents. Subjects b and h often went back to previous lines and read them for this document again.

C. Wordometer

Table V shows the number of read words for each document estimated by the simple method. The average error is 47.95 words. Since the average number of words in a document is 279.3, the error is 17.2%. Table VI shows the number of read words for each document estimated by the SVR method. The average error is 40.5 words (14.5%) thus the SVR method is superior.

Table VII shows the number of estimated words for all documents. For the SVR method we achieve an average error of 8.2%, however we get a really high error rate from subject h due to him also re-reading several lines. If we remove subject h from the analysis we are can improve the error to just about 6.5%.

V. RELATED WORK

A good overview about controlled studies in psychology about eye movements in relation to reading is given by Rayner [11] Kligel et al. explore the relations of eye fixations and mental activities during reading [12].

The existing work so far focuses on controlled lab experiments on specific population groups—mostly elderly and children—trying to diagnose specific disease or mental states (attention deficit, alzheimer etc.) [5], [4]. There is some work showing the strong relation between how much a person reads and their language skills, general knowledge and critical thinking skills [11], [2], [4].

The closest to the work presented here is by Bulling et. al. They present a method to for reading detection using electrooculography [13]. Although their work looks at reading detection in a realistic environment, they do not try to estimate the word count.

Other research classifies the different types of reading, skimming etc. [14], [8]. Xu et al. use eye tracking in a controlled lab environment to create summaries of documents [15].

The Wordometer is complementary to our other efforts to monitor reading activities. We investigated the feasibility of using EEG and eye gaze to infer what document types a user is reading [16], [17]. In other related work, we look into whether language expertise can be inferred from eye gaze [18].

VI. CONCLUSION AND FUTURE WORK

We presented a reading segmentation algorithm, a line break detector and two methods to implement a Wordometer.

In conclusion, our error margin for the Wordometer is 8.2% (or 6.5% removing one test subject). Comparing it to the effectiveness of a pedometers in realistic scenarios (error rate of 3-10%), we believe 8.2% is a good baseline for counting words regarding the impact a Wordometer could have on quantifying language skills and general knowledge.

For future work we want to overcome the limitations of document image retrieval (all documents the user reads need to be registered with the system). Currently our Wordometer methods still rely on document image retrieval for:

1) the correction of the eye gaze (translation to the document coordinate system)

2) the average word count per document.

For the eye gaze correction we can use optical flow. There is related work that shows the successful application of optical flow to filter out head movements during tomography [19]. The simple solution regarding the average word count per line is to apply a constant. Yet this will likely increase the error rate. We will also try an estimation of words per line using the eye fixations and saccades per line.

ACKNOWLEDGMENT

This work is supported in part by the CREST project from Japan Society for the Promotion of Science (JSPS).
TABLE V. Number of read words for each document estimated by the simple method.

<table>
<thead>
<tr>
<th>doc.</th>
<th># words</th>
<th>ave. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>238.0</td>
<td>333.7</td>
</tr>
<tr>
<td>B</td>
<td>203.9</td>
<td>233.4</td>
</tr>
<tr>
<td>C</td>
<td>210.5</td>
<td>324.4</td>
</tr>
<tr>
<td>D</td>
<td>296.1</td>
<td>312.4</td>
</tr>
<tr>
<td>E</td>
<td>276.9</td>
<td>324.6</td>
</tr>
<tr>
<td>F</td>
<td>276.9</td>
<td>286.9</td>
</tr>
<tr>
<td>G</td>
<td>281.3</td>
<td>312.2</td>
</tr>
<tr>
<td>H</td>
<td>298.1</td>
<td>333.2</td>
</tr>
<tr>
<td>I</td>
<td>255.0</td>
<td>272.0</td>
</tr>
<tr>
<td>J</td>
<td>417.1</td>
<td>251.3</td>
</tr>
<tr>
<td>aave.</td>
<td>35.4</td>
<td>19.1</td>
</tr>
</tbody>
</table>

TABLE VI. Number of read words for each document estimated by the SVR-based method.

<table>
<thead>
<tr>
<th>doc.</th>
<th># words</th>
<th>ave. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>255.0</td>
<td>253.2</td>
</tr>
<tr>
<td>B</td>
<td>203.9</td>
<td>225.2</td>
</tr>
<tr>
<td>C</td>
<td>210.5</td>
<td>252.8</td>
</tr>
<tr>
<td>D</td>
<td>296.1</td>
<td>258.0</td>
</tr>
<tr>
<td>E</td>
<td>276.9</td>
<td>266.9</td>
</tr>
<tr>
<td>F</td>
<td>276.9</td>
<td>266.6</td>
</tr>
<tr>
<td>G</td>
<td>281.3</td>
<td>252.8</td>
</tr>
<tr>
<td>H</td>
<td>298.1</td>
<td>324.4</td>
</tr>
<tr>
<td>I</td>
<td>255.0</td>
<td>286.9</td>
</tr>
<tr>
<td>J</td>
<td>417.1</td>
<td>333.2</td>
</tr>
<tr>
<td>aave.</td>
<td>65.6</td>
<td>38.4</td>
</tr>
</tbody>
</table>

TABLE VII. Estimated number of read words for all documents

<table>
<thead>
<tr>
<th>subject</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>ave. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple method</td>
<td>2628.8</td>
<td>3415.1</td>
<td>2784.6</td>
<td>3059.9</td>
<td>2950.5</td>
<td>2697.4</td>
<td>2070.2</td>
<td>4085.0</td>
<td>3459.0</td>
<td>375.1</td>
</tr>
<tr>
<td>errors by the simple method</td>
<td>1514.2</td>
<td>622.1</td>
<td>8.4</td>
<td>272.9</td>
<td>157.5</td>
<td>105.7</td>
<td>85.8</td>
<td>1312.5</td>
<td>666.0</td>
<td>13.5%</td>
</tr>
<tr>
<td>SVR-based method</td>
<td>2200.4</td>
<td>2897.5</td>
<td>2796.0</td>
<td>2758.8</td>
<td>2530.8</td>
<td>2672.9</td>
<td>267.4</td>
<td>3401.5</td>
<td>3048.5</td>
<td>229.8</td>
</tr>
<tr>
<td>errors by the SVR-based method</td>
<td>572.6</td>
<td>94.5</td>
<td>3.9</td>
<td>36.2</td>
<td>262.2</td>
<td>120.1</td>
<td>115.6</td>
<td>608.5</td>
<td>255.5</td>
<td>8.2%</td>
</tr>
</tbody>
</table>

REFERENCES


