Electronic Preprint for Journal of Information Processing Vol.25

Technical Note

Vertical Error Correction Using Classification of Transitions between Sequential Reading Segments

Akito Yamaya[1,a) Goran Topić[2,b) Akiko Aizawa[2,1,c)

Received: March 4, 2016, Accepted: October 4, 2016

Abstract: In eye-tracking-based reading behavior research, gaze sampling errors often negatively affect gaze-to-word mapping. In this paper, we propose a method for more accurate mapping by first taking adjacent horizontally progressive fixations as segments, and then classifying the segments into six classes using a random forest classifier. The segments are then reconstructed based on the classification, and are associated with a document line using a dynamic programming algorithm. The combination of segment-to-line mapping and transition classification achieved 87% mapping accuracy. We also witnessed a reduction of manual annotation time when the mapping was used as an annotation guiding tool.

Keywords: eye-tracking, reading, error correction, fixation-to-word mapping

1. Introduction

Reading is an important language-based activity in our daily life. Eye movements when reading are useful in a variety of research fields such as prediction of language expertise or document layout optimization. Eye-trackers can visualize human eye movements as a consecutive stream data consisting of fixations and saccades. Fixations are points where readers concentrate their gaze to see a specific object. Saccades are quick synchronized eye movements between fixations.

To analyze the reading process using gaze data captured by eye-trackers, we require an accurate mapping between the fixation points and the corresponding words in the document. However, it is difficult to achieve such a mapping under the current tracking environment. One reason for these misplacements is the systematic error such as limitations in the accuracy of the tracking equipment, inaccurate and imprecise calibration, and the spontaneous head movements of readers. Another reason is complexity of human eye movements. Basically, most of the horizontally written text are read by moving eyes from left to right. Our eyes lie on the texts and move sequentially to progress the reading activity. However, eyes also have several other reading behavior patterns such as return sweep, re-reading, and skipping. Return sweep occurs when we transition from the end of the current line to the beginning of the next line. Re-reading occurs mainly when we can't understand the sentence. Skipping occurs when we wish to skip the current reading area or to search the target object from the text. We can go back if we understand the content, and skip if the contents is not for us, so that finding the appropriate fixation-to-word mapping is difficult. Human manual annotation guarantees a higher mapping quality, but requires considerable human resources and time.

In this paper, we propose an approach for vertical error correction as a preparatory step to automatic fixation-to-word mappings [15] [*1]. Our method considers consecutive horizontally progressive fixations as a sequential reading segment. We focus on the transitions between segments, and classify them into six classes. The machine-learning-based classifier is implemented with features including segment length, saccade angle, and transition direction. Segments are reconstructed based on the classification and are associated with the document line using dynamic programming algorithm.

The concept of sequential reading segments reflects real human reading behavior, and thus gives us useful high-level information applicable to pairing up individual fixations and corresponding words. The classification is used to adjust for the non-linear reading behaviors and to identify which segments correspond to the same line of text.

Our results show that we achieved 87% of fixation-to-word mapping accuracy when we combine the segment-to-line mapping method with the transition classification. We also confirmed significant annotation time reduction when our approach is used as a guiding tool.

2. Related Work

According to Martínez-Gómez et al. [9], natural reading that normally happens in uncontrolled environments is very hard to process with current equipment, particularly due to the vertical error. Increasing line spacing to avoid the vertical error dramati-

[1] The University of Tokyo, Bunkyo, Tokyo 113–8654, Japan
[2] National Institute of Informatics, Chiyoda, Tokyo 101–8430, Japan
[a)] yamaya@nii.ac.jp
[b)] goran_topic@nii.ac.jp
[c)] aizawa@nii.ac.jp

[*1] This work is based on an earlier work: Fixation-to-Word Mapping with Classification of Saccades, in *Proceedings of the ACM International Conference on Intelligent User Interfaces* (*IUI '16*), © ACM, 2016. http://dx.doi.org/10.1145/2876456.2879481

cally increases the accuracy of the measurement, but it forces the subjects to read unnaturally. There have been several recent attempts to address the alignment problem between gaze and text. Mishra et al. [10] heuristically used the first sequential fixations to determine the reading line. Although the results showed improvements in fixation re-mapping, this method does not overcome accumulated systematic errors. Martínez-Gómez et al. [9] proposed a feature-based image-registration method that spatially transforms the image representation of fixations to match the image representation of the text. This method uses several optimization strategies to search for the optimal scaling and translation parameters. Although this method achieved feasible accuracy, the precision of the word-level error correction was relatively less. Our previous study [16] considered sequential consecutive fixations, inspired by Carl's [3] dynamic programming-based approach, and associated sequential fixations with a text line to reduce the vertical misplacement of the gaze position. Although our previous study obtained better accuracy compared with manual annotation, it may not perform well in the presence of long-range regressions or skimming events.

As described in the previous section, eye movement when reading has various kinds of complexities. In order to analyze reading behavior, several classification techniques have been studied. Biedert et al. [2] proposed a robust approach that distinguishes whether the fixation is reading or skimming using gaze characteristics. They considered the forward speed and angularity as features for machine learning classification. The methodology employed in their study is helpful to us in using such effective computing features. Hara et al. [4] carried out a skipping word prediction task on the Dundee corpus [5], a large eye movement dataset. They choose a conditional random field model and extract a number of features from gaze and text data. Although the model is capable of estimating texts that are fixated, re-reading eye movement is removed and never used in their analysis. Nilsson and Nivre [11] tackled this problem using a machine-learning technique on the Dundee corpus. They used some of gaze features like length, frequency class and saccade distance. Kunze et al. [8] took into account what types of documents a user read solely from gaze data. They asked users to read several types of documents including comics, textbooks, magazines, novels, and newspapers. They selected angle, direction, and slope as the features and modeled each eye movement. This work was done for the application of a reading life log which is how many words people read and what is read by a user in their life. Based on these studies, we implemented the classifier that classifies the transitions between consecutive reading segments.

3. Proposed Method

Figure 1 illustrates the overview of our approach. We identify the text lines by simply extracting the set of Y-coordinates of each line from the text data. Our method processes gaze and text data into segments and text lines, and subsequently applies dynamic programming to determine the optimal alignment between the segments and the lines.

Raw gaze data represents the time stream sequences based on the capturing timing of eye-tracker. Because such gaze data is too complicated to use directly, we first detect fixations from the raw gaze stream. We used Dispersion-Threshold Identification (I-DT) algorithm [13] for detecting fixations from the raw input.

3.1 Segmentation

We convert a series of valid fixation points into sequential reading segments. Although the classification of the reading and skimming scanpath has been studied in the past [1], these methods are not directly applicable here, because our objective is to identify a scanpath that belongs to a single line of text, with backward saccades allowed if they occur on the same line. In order to satisfy such a requirement, we implement a segment detection scheme based on the Kimura's return sweep detection algorithm [6]. Segments are detected when the size of the bounding box of consecutive translations is above a threshold.

3.2 Classification of Segment Transitions

If a user reads a document from top to bottom with no re-reading or skipping, the number of sequential reading segments should agree exactly with the number of lines of text. However, this will not match the actual number of eye movements, because re-readings and skippings are bound to occur. To solve this issue,

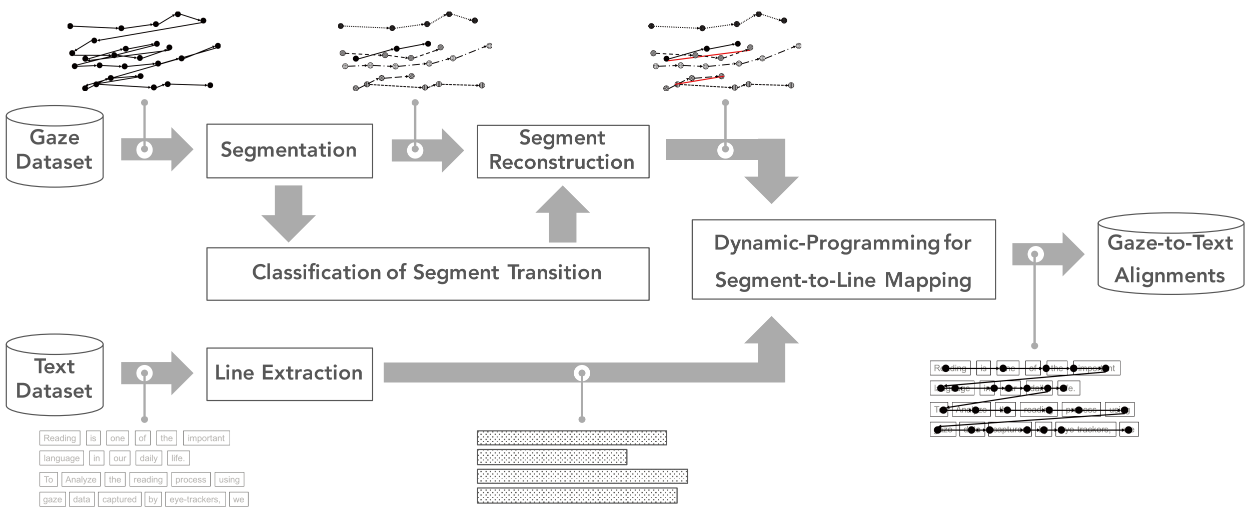

Fig. 1 Structural overview of gaze-to-text alignment.

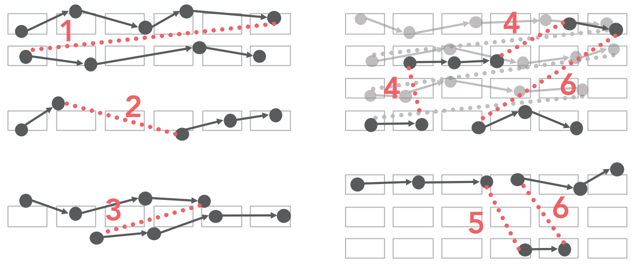

Fig. 2 Transition types between segments.

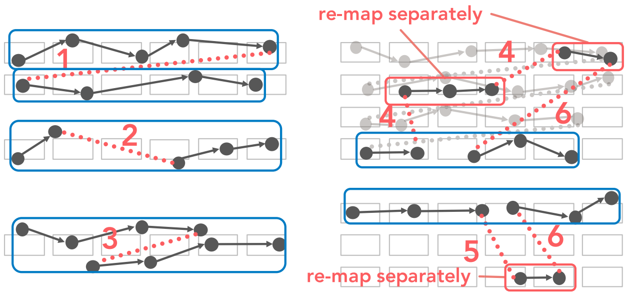

Fig. 3 Segment re-construction based on transition types.

we focus on the transition between consecutive segments.

We first categorized the transitions into the six types described in **Fig. 2**, which we defined empirically based on the observed gaze patterns, and then constructed a classifier that labels the transitions. The features we employed for modeling the classifier are based on previous studies [2], [7]. The features are mainly gaze-, segment-, and transition-oriented geometrical characteristics, and not only extracted from a transition that is focused on, but also from two transitions before and after the current one.

1. Return Sweep

Return Sweep is the right-to-left horizontal eye movement that moves from the end of one line to the beginning of the next.

2. Progressing

Progressing is the left-to-right horizontal eye movement used to read the next area of interest. This occurs when segments are divided on one line.

3. Short Re-reading

Short Re-reading refers to re-reading on the same line. This occurs when the user cannot adequately understand the sentence, and so reads again from the beginning or halfway along the same line, or when the user's saccade exceeds its expected position and goes backward.

4. Long Re-reading

Long Re-reading refers to multi-line re-reading. This is characterized by upward eye movement with the possibility of going back continuously. *Long Re-reading* occurs when the reader wishes to refer to a sentence, phrase, or word that has already been read.

5. Skipping

Skipping is when the next landing target is ignored and the gaze jumps to a new position. In particular, we observe this when users change the line they are reading, even though they have not read all of the current line.

6. Resuming

Resuming occurs when the user returns to the latest reading point following a period of Long Re-reading or Skipping.

3.3 Segment Reconstruction

After the classification of transitions, the segments are reanalyzed. Each transition represents either a breaking point or a concatenation point of the two segments, as shown in **Fig. 3**. The general procedure of the algorithm consists of two steps: 1) divide or concatenate segments based on the transition label, and 2) separately re-map the re-reading or skipping segments. If the transition is a *Return Sweep*, the segments are divided. If the transition indicates *Progressing* or *Short Re-reading*, then the segments are concatenated. In the case of *Long Re-reading* or *Skipping*, the following segment is temporarily stored. If *Resuming* is detected before the next *Return Sweep*, the preceding segment is concatenated with the following segments of *Resuming*, otherwise the previous segment is not concatenated to any other segment. After the division or concatenation, stored segments are mapped to their spatially closest preceding segment.

3.4 Segment-to-line Mapping

After reconstruction, the segments are used to find appropriate lines in the document. We extended both the similarity function and the penalty score of our previous study [16] to enhance the accuracy and robustness.

Let $S = \{s_1, s_2, \cdots, s_N\}$ be a sequence of reading segments, and $L = \{l_1, l_2, \cdots, l_M\}$ be the lines of text. Suppose the mapping function σ assigns a text line to the reading segment: i.e., $\forall s_i \in S$, $\exists l_j \in L$ such that $\sigma(s_i) = l_j$. Our segment-to-line alignment problem can then be formulated as the problem of finding the optimal σ between S and L.

The optimal alignment is found by completing a scoring matrix with $N+1$ columns and $M+1$ rows. To find the maximum score of each cell $D(i, j)$, we select the maximum score from the assumed values from adding the match or mismatch score to the diagonal value $D(i-1, j-1)$, or the insertion or deletion gap score to the left-hand cell $D(i, j-1)$ and top cell $D(i-1, j)$. These values are formulated as follows:

$$D(i, j) = \max \begin{cases} D(i-1, j-1) + Sim(s_i, l_j) \\ D(i, j-1) + inPen(j) \\ D(i-1, j) + delPen \end{cases} \tag{1}$$

where $Sim(s_n, l_m)$ represents a similarity score that measures the matching degree between the segment and the line, and *inPen*, *delPen* are gap penalties for insertion (adding a gap between lines) and deletion (adding a gap between segments), respectively. After filling the matrix, the optimal alignment is obtained by following the immediate predecessors from the last cell (N, M) to the original cell (0, 0).

Whether a segment represents a match or mismatch is determined by comparing the similarity between a segment and a line. Given a reading segment s_i, the set of neighboring text lines, denoted as $Neighbor(s_i)$, is defined based on the average Y-coordinate of that segment. In our case, we consider three neighbors: 1) the nearest line to s_i; 2) one line above the nearest line; and 3) one line below the nearest line.

$$(s_i, l_j) = \begin{cases} \text{match} & abs\left(1 - \frac{Len(s_i)}{Len(l_j)}\right) < \alpha \\ & \wedge l_j \in Neighbor(s_i) \\ \text{mismatch} & \text{otherwise} \end{cases} \tag{2}$$

We assume that the segment length should be roughly equal to the corresponding line length. The similarity function $Sim(s_i, l_j)$ is then formulated as:

$$Sim(s_i, l_j) = z \cdot \frac{Len(s_i)}{Len(l_j)} \cdot \frac{MedLen(L)}{Len(l_j)}, \tag{3}$$

where $z = 1$ for a match and $z = -1$ for a mismatch. $Len(s_i)$ and $Len(l_j)$ represent the length of a segment and text line, respectively. $MedLen(L)$ is the median of the entire text line, and the parameter α determines whether the segment-to-line alignment is a match or mismatch. In our experiments, we set $\alpha = 0.4$. $MedLen(L)$ is used to place extra weight on shorter lines at the end of a paragraph. This is motivated by our observation that such lines often serve as strong alignment clues.

We set $inPen(j) = 0$ if the segment represents a match with the line, and set $inPen(j)$ to the segment id j if it represents a mismatch. This is mainly because we consider segment insertions to be less important if we have a match, whereas insertions may be critical if there is a mismatch. That heuristic using the segment id is based on the consideration that the more users proceed to read, the more segment insertions, such as re-readings and skippings, are likely to occur. We empirically set *delPen* to a high value of $\beta = 100$, because skipping lines is unnatural during sequential reading.

4. Experiments

We used a part of the dataset in Ref. [9] because it features natural reading activity. Eleven participants read three English documents each, for the total of 33 gaze datasets. The participants were a mix of native and non-native speakers. The Tobii TX300, a screen-based remote eye-tracker, was used with default settings throughout the experiments. According to Ref. [9], participants read the text naturally, and did not use any equipment to fix their head position.

4.1 Experiment 1: Classification Performance

To assess the classification performance, a human annotator determined the label for each transition. The experiment was conducted under leave-one-out cross-validation. A random forest classifier was adopted, with optimal parameters found by a grid search.

Table 1 represents the result of the manual labeling, showing ratios of six transitions. There are 1,606 transitions in the 33 gaze dataset. We confirmed that the *Return Sweep* has about 58.5% of all transitions. The *Progressing* and *Skipping* barely appear because *Progressing* is false positive activity of the segmentation, it should be a piece of the sequential reading segment, and *Skipping* is a rare event when people naturally read a document from top to bottom. The *Short Re-reading* and *Long Re-reading* transitions accounted for a combined 19%, which is slightly higher than the 10–15% suggested by Rayner's report [12] due to frequent re-reading by some of our non-native participants.

Table 1 Transition ratio and classification performance.

Label	Num.	Ratio	Precision	Recall	F1-score
Return Sweep	940	0.585	0.85	0.97	0.91
Progressing	99	0.062	0.67	0.66	0.66
Short Re-reading	133	0.083	0.68	0.35	0.47
Long Re-reading	180	0.112	0.77	0.85	0.81
Skipping	52	0.032	0.46	0.23	0.31
Resuming	202	0.126	0.64	0.46	0.53
	1,606	1	0.78	0.80	0.78

Table 1 also presents the classification report in terms of the precision, recall, and F1-score of each label. An average F1-score of 0.78 (and 84% accuracy) was achieved by the classifier, with the *Return Sweep* and *Long Re-reading* transitions particularly well characterized. Although some of the labels showed higher error rates, the impact to the subsequent mapping step is limited, as these easily misclassified transition categories are significantly outnumbered by the more clearly recognizable ones. However, it should be emphasized that our classifier still has room for improvement.

Figure 4 shows the confusion matrix of our classifier. As shown by the red dot circle in Fig. 4, *Short Re-reading* and *Resuming* have a higher tendency of wrong prediction. They are mostly mislabeled as *Return Sweeps*. These results show that *Short Re-reading* and *Resuming* are geometrically similar to a *Return Sweep*. Indeed, eye movements of these reading behaviors include long right to left movement. Further investigation is needed for more robust and accurate classification.

Table 2 lists the 10 features with largest impact to our classifier. According to Refs. [2] and [7], saccade angularity and its direction are useful features for forecasting reading behavior. Our results confirmed similar trends. Additionally, our results show that the transition distance of X coordinate is the most character-

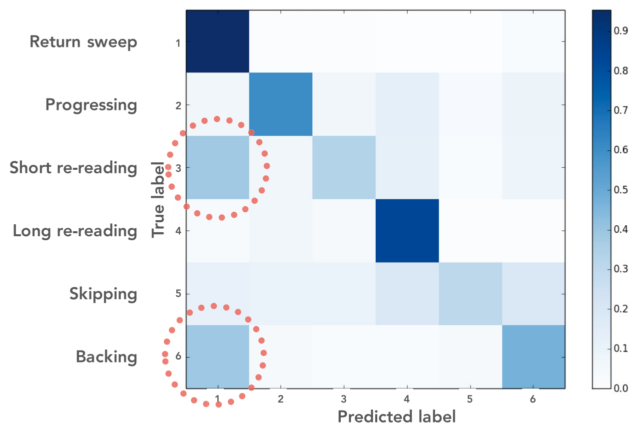

Fig. 4 Actual and predicted classifications of transitions.

Table 2 Contributing features.

Rank	Feature Name
1	X-distance of Transition
2	Saccade Angle
3	Direction of Transition
4	Short Reread Sign
5	Y-distance of Re-reading
6	Length of Previous Segment
7	Difference of Nearest Text Line Between Segments
8	Euclidean Distance of Transition
9	Length of Previous Segment[-1:]
10	X-distance Between Segments

Table 3 Performance of fixation-to-word mapping.

	Document 1			Document 2			Document 3		
User	Baseline	DP	DP + Classifier	Baseline	DP	DP + Classifier	Baseline	DP	DP + Classifier
A	55.2	61.3	92.4	75.9	94.6	96.8	50.8	95.5	95.8
B	69.3	65.8	91.7	57.2	60.8	86.9	58.3	83.5	84.8
C	88.1	19.7	85.2	79.2	90.1	92.9	76.3	57.2	92.4
D	75.1	44.6	73.1	84.4	74.4	93.8	53.4	41.0	68.2
E	60.3	84.3	86.9	56.9	64.9	78.7	57.6	64.4	91.7
F	77.0	63.0	80.2	78.7	93.4	94.0	63.3	90.6	94.9
G	62.1	96.7	85.1	71.8	25.1	92.9	78.0	94.6	94.9
H	56.5	90.9	90.6	59.5	91.1	92.4	77.4	95.5	96.0
I	78.4	86.2	84.7	85.2	96.5	92.4	68.3	93.9	94.6
J	63.7	96.2	96.2	60.4	11.8	44.4	71.9	55.3	90.7
K	80.5	57.1	70.4	72.8	85.5	86.1	74.2	62.2	80.8
average							69.0	72.3	87.0

istic feature. Further improvement may be possible if we use not only geometrical features, but also linguistic or temporal features.

4.2 Experiment 2: Accuracy of Fixation-to-Word Mapping

The golden fixation-to-word mapping dataset is created using FixFix [14]. We employed four annotators, one to map all of gaze data and the other three to map gaze data corresponding to each document. We confirmed that 84.4% of mappings agreed on average: evaluation measure of its agreement is ratio of exactly matching of annotators' fixation-to-word mappings in the total number of fixations.

As mentioned above, manually creating the fixation-to-word mapping guarantees high quality at the expense of the human annotation load. To assess the reliability of the mapping method, we compared the differences in offset word position with the manual mapping. As an evaluation index, *Accuracy*(%) was calculated as $\mathit{Correctly\ Mapped\ Fixations}/\mathit{Total\ Fixations} \times 100$.

We compared three mapping methods. *Baseline* is the naïve mapping produced by each fixation immediately shifting to the closest word. *DP* uses the dynamic programming-based segment-to-line mapping (described in our previous work [16]) as a pre-step to the naïve mapping. *DP+Classifier* uses our proposed mapping technique with transition classification as a pre-step to naïve mapping.

Table 3 compares the accuracy of each mapping method. To assess the reliability of the automatic mapping method, we compared the differences in offset word position with the manual mapping. While the mapping accuracy of DP only, without transition classification, is 72.3%, our method *DP+Classifier* exhibits a significant improvement, with an average accuracy of 87.0%. These results confirmed the effectiveness of using transition classification. Although we achieved reasonably high mapping accuracy, in several cases our method was less effective than the naïve method (C, D and K in document 1 and J in document 2). A detailed examination suggests these errors mainly occur when readers re-read a large portion of text or do not read the whole document.

Figure 5 shows an example of the captured gaze and reading text data. It has been observed that the complexity of fixation-to-word mapping is increased by noisy tracking environments such as narrow line spacing, low-precision sampling, or re-reading/skipping activities. **Figure 6** illustrates one of the

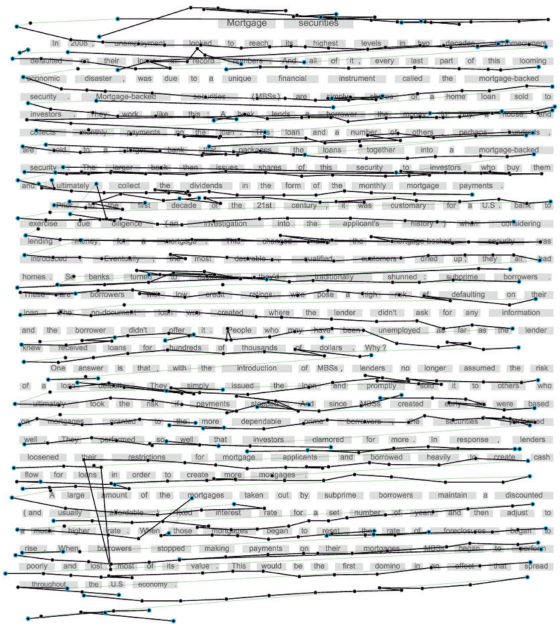

Fig. 5 Example of raw fixation data (Doc 2, User H).

successful mapping cases. The *DP+Classifier* fixation-to-word mapping performance is 92.4% (see Table 3), a significant improvement over 59.5% of *Baseline* mapping. Our method seems to give a reliable estimation compared with the original position, because the shorter reading segments shift to the shorter reading line.

4.3 Experiment 3: Measurement of Annotation Time

As a supplementary experiment, we utilized our automatically created mapping results as a guiding tool for the manual annotation. We have implemented the guiding function on the web-based manual annotation tool FixFix [14], which provides an easy-to-use interface for adjusting the fixation positions. **Figure 7** shows the use case of the function. This function enables the linking of both original and aligned gaze fixations at the same time stamps. We asked six participants to annotate four gaze datasets, a smaller version of text in the interest of time, with and without the function. Each participant conducted eight annotations in total, then the annotation order is randomized. In order to assess the efficiency, we measured both annotation time reduction and annotation agreement.

Fig. 6 Example of mapped fixation data (Doc 2, User H).

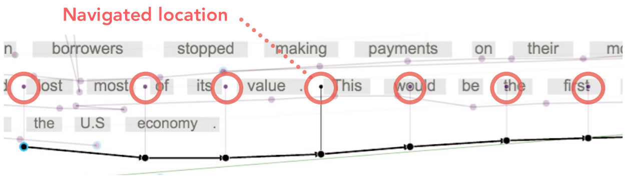

Fig. 7 Application use case: guiding tool for the manual annotation.

Table 4 Time reduction ratio of manual annotation.

Participant	a	b	c	d	e	f	average
Ratio	6.3	5.9	10.4	12.7	8.4	14.6	9.7

Table 5 Annotation agreement between annotators.

Doc : User	2 : G	2 : H	2 : I	2 : K	average
Without_guidance	84.2	82.1	89.7	89.7	86.4
With_guidance	90.5	90.1	92.6	90.9	91.2

Table 4 shows time reduction from six participants who conducted four gaze annotations. Time reduction ratio is calculated as $1 - \mathit{Time\ with\ guidance}/\mathit{Time\ without\ guidance}$. We observed an 9.7% reduction in annotation time. **Table 5** shows annotation agreement of the six participants with or without the function. We confirm an increase in agreement by using the guiding function. Since this experiment is still in its preliminary stage due to the lack of sufficient data quantity, it is difficult to state the significance of the result. However, it suggests the potential of using our mapping method to assist with the annotation task, and further discussion for future works.

5. Conclusion

In this paper, we have proposed an alignment methodology that reduces the vertical misplacement of noisy gaze data. Our method regards horizontally progressing temporally consecutive fixations as a sequential reading segment. We focus on the transition between the segments, and classify it into six classes. Machine-learning-based classifier characterizes the transition by both transition- and fixation- oriented features such as segment length, saccade angle, and transition direction. Segments are reconstructed based on classified labels, and associated with a document line. The dynamic programming is applied to determine the best global alignment across the entire stream of gaze data samples. The cost function is based on the length of reading segments and document text lines to determine whether a reading line matches a document line. Our experiments demonstrate that the proposed method, as a preparatory step, yields 87% accuracy in naïve distance-based fixation-to-word mapping (up from 69% of pure naïve method, and 72.3% of the previous method without transition classification [16]). Based on this result, we believe our method to be a potentially valuable instrument in the processing of gaze data from reading activities. We also confirmed that manual annotation time can be reduced by using our approach as a guiding tool. Future work for this topic involves refining our method to make it more robust to the presence of frequent and large regressions in the dataset. Moreover, we will consider horizontal errors, to produce better overall fixation-to-word mappings. It can be said that our method provides sufficiently good accuracy to warrant future analysis.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 24300062.

References

[1] Biedert, R., Dengel, A., Elshamy, M. and Buscher, G.: Towards Robust Gaze-based Objective Quality Measures for Text, *Proc. Symposium on Eye Tracking Research and Applications* (*ETRA '12*), pp.201–204 (2012).

[2] Biedert, R., Hees, J., Dengel, A. and Buscher, G.: A Robust Realtime Reading-skimming Classifier, *Proc. Symposium on Eye Tracking Research and Applications* (*ETRA '12*), pp.123–130 (2012).

[3] Carl, M.: Dynamic programming for re-mapping noisy fixations in translation tasks, *Journal of Eye Movement Research*, Vol.6, No.2, pp.1–11 (2013).

[4] Hara, T., Mochihashi, D., Kano, Y. and Aizawa, A.: Predicting Word Fixations in Text with a CRF Model for Capturing General Reading Strategies among Readers, *Proc. 1st Workshop on Eye-tracking and Natural Language Processing*, pp.55–70 (2012).

[5] Kennedy, A. and Pynte, J.: Parafoveal-on-foveal Effects in Normal Reading, *Vision Research*, Vol.45, No.2, pp.153–168 (2005).

[6] Kimura, T., Huang, R., Uchida, S., Iwamura, M., Omachi, S. and Kise, K.: The Reading-Life Log – Technologies to Recognize Texts That We Read, *Proc. 12th International Conference on Document Analysis and Recognition* (*ICDAR '13*), pp.91–95 (2013).

[7] Kunze, K., Masai, K., Inami, M., Sacakli, O., Liwicki, M., Dengel, A., Ishimaru, S. and Kise, K.: Quantifying Reading Habits: Counting How Many Words You Read, *Proc. 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (*UbiComp '15*), pp.87–96 (2015).

[8] Kunze, K., Utsumi, Y., Shiga, Y., Kise, K. and Bulling, A.: I Know What You Are Reading: Recognition of Document Types Using Mobile Eye Tracking, *Proc. 2013 International Symposium on Wearable Computers* (*ISWC '13*), pp.113–116 (2013).

[9] Martínez-Gómez, P. and Aizawa, A.: Recognition of Understanding Level and Language Skill Using Measurements of Reading Behavior, *Proc. 19th International Conference on Intelligent User Interfaces* (*IUI '14*), pp.95–104 (2014).

[10] Mishra, A., Carl, M. and Bhattacharyya, P.: A Heuristic-based Approach for Systematic Error Correction of Gaze Data for Reading, *Proc. 1st Workshop on Eye-tracking and Natural Language Processing*, pp.71–80 (2012).

[11] Nilsson, M. and Nivre, J.: Learning Where to Look: Modeling Eye Movements in Reading, *Proc. 13th Conference on Computational Natural Language Learning* (*CoNLL '09*), pp.93–101 (2009).

[12] Rayner, K.: Eye Movements in Reading and Information Processing: 20 Years of Research., *Psychological Bulletin*, Vol.124, No.3, pp.371–422 (1998).

[13] Salvucci, D.D. and Goldberg, J.H.: Identifying Fixations and Saccades in Eye-tracking Protocols, *Proc. 2000 Symposium on Eye Tracking Research and Applications* (*ETRA '00*), pp.71–78 (2000).

[14] Topić, G., Yamaya, A., Aizawa, A. and Martínez-Gómez, P.: FixFix: Fixing the Fixations, *Proc. Symposium on Eye Tracking Research and Applications* (*ETRA '16*) (*accepted*) (2016).

[15] Yamaya, A., Topić, G. and Aizawa, A.: Fixation-to-Word Mapping with Classification of Saccades, *Proc. ACM International Conference on Intelligent User Interfaces* (*IUI '16*) (*accepted*) (2016).
[16] Yamaya, A., Topić, G., Martínez-Gómez, P. and Aizawa, A.: Dynamic-Programming–Based Method for Fixation-to-Word Mapping, *Proc. 7th KES International Conference on Intelligent Decision Technologies* (*KES-IDT '15*), pp.649–659 (2015).

Akito Yamaya received his master's degree from the University of Tokyo in 2016. He is a system development engineer at NTT DATA corporation. His current research interest is human eye movements and reading analysis.

Goran Topić received his master's degree from the University of Tokyo in 2012. He works as a programmer at KMCS-NII.

Akiko Aizawa received her M.E. and Ph.D. from the University of Tokyo in 1987 and 1990. She is a professor in Digital Content and Media Sciences Research Division at National Institute of Informatics. She is also an adjunct professor at the University of Tokyo. Her current research interests include natural language processing and text media. She is a senior member of the IPSJ and a member of the IEICE, JNLP, IEEE and ACM.