

Robust Annotation Positioning in Digital Documents

A.J. Bernheim Brush, David Barger, Anoop Gupta, and JJ Cadiz

September 22, 2000

Technical Report
MSR-TR-2000-95

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Robust Annotation Positioning in Digital Documents

A.J. Bernheim Brush¹, David Barger, Anoop Gupta, and JJ Cadiz

Collaboration and Multimedia Systems Group

Microsoft Research

One Microsoft Way

Redmond, WA 98052

ajb@cs.washington.edu, {davemb, anoop, jjcadiz}@microsoft.com

ABSTRACT

Increasingly, documents exist primarily in digital form. System designers have recently focused on making it easier to read digital documents, with annotation as an important new feature. But supporting annotation well is difficult because digital documents are frequently modified, making it challenging to correctly reposition annotations in modified versions. Few systems have addressed this issue, and even fewer have approached the problem from the users' point of view. This paper reports the results of two studies examining user expectations for "robust" annotation positioning in modified documents. We explore how users react to lost annotations, the relationship between types of document modifications and user expectations, and whether users pay attention to text surrounding their annotations. Our results could contribute substantially to effective digital document annotation systems.

Keywords

Annotation, robust, digital, documents, annotation system design.

1 INTRODUCTION

Four common activities surrounding documents are reading, annotating, collaborating, and authoring. Until recently, computer software vendors have primarily focused on the authoring task, with products like Microsoft Word. The industry is now realizing that, in fact, the most pervasive activity around documents is not authoring but reading, followed closely by annotating, then collaborating, and finally authoring. The majority of people read and annotate daily, but do not create new documents.

With this shift in perspective, there is an increased focus on software primarily targeting reading and annotating [10][11]. The reading-centric products are aware of the importance of pagination over scrolling, of side margins, and of the relationships between font size, line spacing, and line width. The annotation capabilities provided to date are, however, more primitive and currently being refined.

Cadiz et al [3] report on a recent study where they observed the use of electronic annotation by roughly 450 users over a 10-month period. While there were many observed benefits,

a key complaint was the *orphaning* of annotations. That is, when the online documents got changed, the annotations lost the link to their proper position within the document, and were presented at the bottom of the document.

The problem of orphaning is unique to annotations on digital/online documents, as paper-based documents do not change underneath the annotator. As more documents appear online and as other traditionally paper-based document processes become increasingly digital (such as editing and revision), "robust annotations" that remain associated with the correct portion of the document across modifications will become crucial.

But correctly positioning annotations in a revised document is a difficult problem. Some annotation systems work around the problem by limiting where an annotation can be placed [4][11], others silently orphan or drop annotations when documents change [5]. Researchers have begun to explore algorithms for robustly saving an annotation's position and finding it in a modified version of the document [6][15]. However, we believe focusing solely on algorithmic approaches to this problem neglects a crucial step. No one has asked *users* what they expect an annotation system to do when a document changes.

This paper's primary contribution is to take that step by reporting the results of two studies. Participants in the studies made annotations, transferred them to modified documents manually and also rated how well a simple algorithm positioned annotations in modified documents. Our belief was that observing the thought processes people use to place annotations in a modified document would help us create a robust positioning algorithm that does what people expect.

Some of the results were surprising. It was unexpectedly difficult for study participants to work with annotations that they had not made. Even when part of the original text associated with an annotation was found, in some cases it seemed participants would have preferred the system to orphan the annotation. Also, participants appeared to pay little attention to the text surrounding their annotations.

¹ Current Address: Computer Science and Engineering Department, University of Washington, Seattle, WA 98195

In the next section we review related work. Then Section 3 lays out a framework for annotation position information and types of document modifications. Sections 4 and 5 describe the methodology of our two studies and their results. Section 6 discusses how we can use these results to construct better robust positioning algorithms for annotations

2 RELATED WORK

Effectively positioning annotations in a digital document is a non-trivial problem. The exact document text related to an annotation is often ambiguous. For instance, Marshall [9] suggests that people frequently place their annotations carelessly. The underlines and highlights they create (on paper in this case) often follow document structure or typographical characteristics rather than content.

The positioning problem is even more difficult if the underlying document can be modified. Users are not forgiving when a system fails to correctly position their annotations in a modified document [3]. Thus, previous systems have taken a wide variety of approaches toward solving the positioning problem, as outlined below.

2.1 Annotating “Frozen” Documents

Many systems simply assume that annotated digital documents will never change. Adobe Acrobat Reader [1], Aladdin Ghostview [2], and Microsoft eBook Reader [10], all take this approach. Other systems have augmented traditional annotation of paper documents (that don't change) with computer support [12][19]. In both types of systems, annotations are typically positioned using very simple means, such as character offsets, or page number plus an (x,y) position. The underlying document is never modified, so annotations never have to be repositioned.

Other systems do not explicitly require documents to remain unchanged, but work best when there are no modifications. In these systems annotations are created on any web page, stored separately on a central server, and visible to everyone with access to the server. Annotations are typically positioned by calculating a signature from the content of the page to which the annotation belongs. E-Quill [5], Third Voice [18], and Microsoft Office Web Discussions [11] are commercial systems that have taken this approach; public web-scale architectures such as OSF [17] and NCSA [7] do as well.

In many important scenarios such as the Web, however, it is unrealistic to assume that documents will never change. If a document does change, these systems fail to properly position some annotations, and the annotations either silently disappear or are displayed in a separate window as orphans. Not surprisingly, this problem has been found to be particularly debilitating. In a study of the large-scale use of Microsoft Office 2000 Web Discussions, lost annotations was cited as the primary reason people stopped using the system [3]. Our work is aimed at accommodating documents that may get modified.

2.2 Annotating Predefined Positions

Some systems attempt to compensate for potential modifications in web pages by only allowing users to annotate predefined positions. CoNotes [4] requires inserting special HTML-like markup tags into a document before it can be annotated. Microsoft Office Web Discussions [11] only allows users to attach annotations to a small selection of HTML tags. By limiting the places where annotations can be placed, these systems can better control how the annotations are positioned when the underlying page gets modified. Our goal is to allow users to position their annotations anywhere on a digital document.

2.3 More Complex Positioning Algorithms

A number of systems implement more sophisticated positioning algorithms that make very few assumptions about the documents. Annotator [13], ComMentor [16], Webvise [6], and Robust Locations [15] part of Multivalent Annotations [14], are examples of systems that take this approach.

These systems allow annotations to be positioned anywhere within a web page. They all store a combination of annotated text and surrounding text so that the annotation may be repositioned later. ComMentor stores key words that attempt to uniquely identify annotated text. Annotator calculates a hash signature from the annotated text. Webvise stores a “locSpec” for each annotation that includes a bookmark or HTML target name, annotated text, surrounding text and a character count of the start position. Robust Locations stores the position of the annotation in the document tree as well as surrounding text.

The annotations created by these systems are robust to varying degrees. Each system can fail to correctly position an annotation in a modified document and orphan it. The systems have varying strategies for presenting orphans to the user, from separate popup windows [15] to placing them at the end of the document [16].

While we build on the work of these systems, taking a user-centric approach to the problem of robustly positioning algorithms will help us determine the appropriate annotation position information to store and how to design a positioning algorithm that meets users' expectations.

3 FRAMEWORK

Approaching the annotation positioning problem requires understanding two key components: How digital annotations work, and how documents may be modified.

3.1 Annotation Definitions

An *annotation* is a marking made on a document at a particular place. Each digital annotation is composed of two items: Some *content* (for example, a user comment or highlighter ink) and an *anchor* (the information used to position an annotation in the document).

Marshall [8] has classified paper-based annotations into 4 groups based on whether the annotation content is *explicit* to another reader (e.g., a scribbled note) or *implicit* (e.g.,

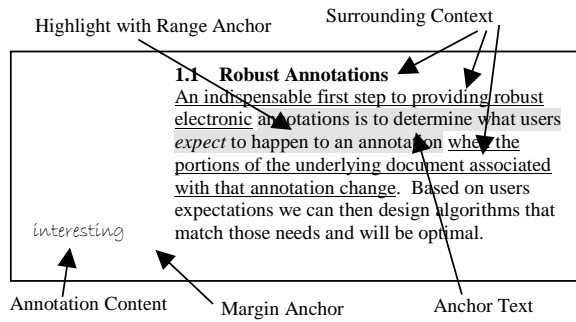


Figure 1: Annotation example showing anchor types and surrounding context. In our user studies we focused primarily on annotations with range anchors.

yellow highlighter ink implying importance) and whether the annotation’s anchor is a *margin* anchor (e.g., asterisks, a note scribbled to the side of a paragraph with no other marking) or a *range* anchor (e.g., highlighted text, circled word). Figure 1 illustrates the two anchor types. The highlight annotation has a range anchor and implicit content, and the asterisk annotation has a margin anchor and explicit content.

3.2 Robust Anchor Representation

The content and anchor information for digital annotations is often stored separately from the annotated document. This strategy allows people to annotate documents even if they don’t have permission to modify them. However, this also requires high quality anchor information. Without a good anchor, a system can’t position annotations correctly in a document for display to users.

To insure correct annotation positioning in a document even when the document changes, a system needs to use *robust anchors*. Robust anchors could potentially use two types of information to identify an annotation’s location:

- *Anchor text information*: E.g., Text under the highlight.
- *Surrounding context information*: Text in the document near the annotation, but not explicitly selected by the user (see Figure 1).

One goal of our studies was to determine the relative value of both types of information to users when trying to position annotations in a modified document.

3.2.1 Anchor Text Information

The key role of anchor text information is to uniquely and efficiently identify the annotation’s position in a document. As was discussed earlier, numerous strategies exist to address this problem, storing simple character offsets, keywords, or the entire text string selected by the user. These methods only work when a user explicitly marks text. Margin annotations don’t mark any text explicitly. For example, does the asterisk in Figure 1 relate to just the last few words, the last sentence, or the complete paragraph?

3.2.2 Surrounding Context

The surrounding context is the text that is near the annotation, but not explicitly selected by a user. For example, the underlined text in Figure 1 can be considered part of the surrounding context for the highlight annotation. More generally, we can think of the surrounding paragraph, subsection, section, and so on as part of the surrounding context. Meta-information, such as HTML markup tags, can also be used as part of surrounding context.

Surrounding context is important for several reasons. First, it is the only way to identify where margin annotations should be positioned. Second, surrounding context can be used, as in Robust Locations [15], to verify that the correct position for the annotation anchor has been located. Third, the range of text specified by the reader may not be carefully chosen [9]. For digital annotations, this may mean that people expect annotations to remain intact if the surrounding context remains, even if large changes occur in the anchor text information.

3.3 Document Modifications

Documents may be modified for different reasons and in a variety of ways. It is important to differentiate between modifications made to address annotations and modifications made independently of annotations.

A modification may be made in response to an annotation. For example, a sentence may be highlighted with “please

Modification Type	Modification	Description
Delete	Minor Delete	Between 1 character and half of the anchor is deleted.
	Medium Delete	More than half of the anchor is deleted.
	Total Delete	Entire anchor is deleted.
Reword	Minor Reword	Between 1 character and half the anchor is reworded.
	Medium Reword	More than half the anchor is reworded, reorganized, or split into multiple pieces.
	Total Reword	Complete anchor is reorganized. Typically only a few key words remain.
Move Anchor Text	Anchor Text Indirect	Anchor text itself doesn’t change, but the text around it does.
	Anchor Text Direct	Anchor text moves within the paragraph or changes paragraphs.
Move Paragraph	Paragraph Indirect	The paragraph in front of the annotation’s paragraph changes.
	Paragraph Direct	The paragraph containing the annotation moves forward or backward.

Table 1: Annotation Anchor Modification Types. The table presents different types of modifications that an annotation’s anchor text may undergo in the document modification process. We use this classification in our study to understand users’ expectations for robust annotation positions.

reword” written in the margin next to it. If the author rewords the sentence, it is difficult to know whether a system should try to position and show the annotation in the modified document. We do not focus on robust positioning of these editing annotations in this paper. A solution based on a “resolve button” is discussed in [3].

Modifications may also be made independently of any annotation. For example, an author may generate a new draft of a document while a colleague marks up a previous draft. This is the case we focus on here.

Our modification classification scheme is shown in Table 1. A piece of text can experience three main types of modifications: Deletes, rewords and moves. Note that a single piece of text may undergo several of these modifications at once.

Although delete and reword modifications are easy to see, move modifications are more complicated. For example, if the paragraph prior to the annotation is deleted, the surrounding context of the annotation changes without any change to the actual text that the annotation is anchored to.

3.4 Study Focus

We chose to focus on a limited number of common annotation and modification types in this paper. First, because the majority of digital annotations use range anchors, not margin anchors – it is easier to highlight text with a mouse than it is to draw an asterisk in a margin – we focused on annotations with range anchors.

Second, we focused on annotations that were made during active reading of text documents, similar to those studied by Marshall [8], instead of examining editing annotations. Annotations made during active reading are often meant to persist for future reference, thus they are precisely the type of annotation that must survive document modifications.

4 PILOT STUDY: ANNOTATIONS ON PAPER

To examine user expectations for robust annotation positions, we conducted two user studies. The main goal of the pilot study was to explore what users perceive as annotation context. We did this by isolating the task from user interface design concerns and having participants perform the task for which we were trying to design an algorithm. We had participants transfer annotations from an original document to a modified version (on paper). Our hypothesis was that observing the thought processes people use to identify the context of an annotation and place it in a modified document would help us create a software algorithm that does what people expect.

4.1 Experimental Method

We recruited 8 participants who had at least basic computer knowledge. All were either college educated or college students and all read for at least 30 minutes on average every day. Participants received a gratuity.

Participants performed three main tasks. First, they looked at a pre-annotated document and told us what they thought

the context for each annotation was. The document was a news article with a variety of types of annotations on it (a selection of highlights, underlines, margin notes and symbols created by four coworkers).

Second, we had participants transfer the annotations from the original document to a version modified by a technical writer.

Third, we had participants compare the original annotated document with a modified version in which “a computer” had positioned the annotations. The annotations on the modified version were actually placed there by a person using an algorithm similar to the method reported in [15]. Participants rated how well the computer did using a 7-point Likert scale.

4.2 Lessons Learned

Instead of obtaining data about the cognitive processes people use to transfer annotations, we learned that making explicit the context of annotations and then transferring them is a difficult task. Problems seemed to stem from the fact that people were asked to work with annotations that they did not make. We consciously designed the task this way so that we could control what type of modifications each annotation was subjected to in the altered version of the document. However, if a participant did not understand (or agree) with an annotation, it negatively affected their ability to specify its context and to transfer it. One participant quipped, “Again we have another star here. That’s a horrible annotation right there.” Another said “I don’t see how it [another annotation] applies, but I guess it does.” One participant even refused to transfer annotations that were “someone’s opinion that I didn’t agree with. Why should I promote their cause?”

Rating the computer’s transfer of the annotations was also difficult because participants were not working with annotations that they had made. Instead of rating the new position of the annotation in the modified version, several participants rated how valuable they thought the annotation was. Also, because the task was done on paper (where it was clear that a person had marked up the document) people had a difficult time understanding that we were pretending a computer had positioned the annotations.

5 SECOND STUDY: DIGITAL ANNOTATIONS

Based on our experience from the pilot study, we conducted a second study where participants created their own annotations on a digital document using software we designed. We narrowed our focus to examine user ratings of annotation positioning done by a computer. Our primary goal for this study was to gauge users’ reactions to a relatively simple repositioning algorithm, especially when it failed.

5.1 Annotation Software

For this study, we extended Microsoft Internet Explorer as shown in Figure 2, to allow people to highlight and make notes on web pages.

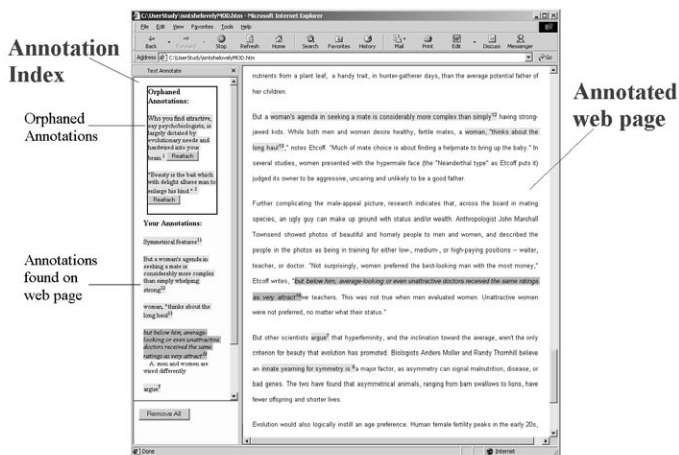


Figure 2: Text annotation software used by participants in the second study to create notes and highlights on web page. The annotation index lists the annotations for the current page, including the orphaned annotations that could not be placed on the page.

A user makes an annotation by using the mouse to select a portion of text on a web page, and then left-clicking the selection. A menu pops up from which the user can choose to highlight or attach a note to the selected text. Highlighted text is displayed with a yellow background and text with a note attached is displayed with a blue one. A list of all annotations for the web page is shown in the annotation index window on the left. This index also displays the contents of any note annotations. All annotations are automatically numbered. Participants could delete annotations by left-clicking on an existing annotation and selecting “delete” from the menu.

5.1.1 Annotation Positioning Algorithm

We included a simple algorithm to reposition annotations if an annotated document was modified. The algorithm was similar to the context method reported in [15]. The algorithm saved the text selected by the participant as the anchor and then used text matching to find the anchor position in the modified version. If all the original text was not found, the algorithm alternated cutting words off the front and back of the anchor text while looking for the shorter text in the modified document until it found a partial match, or until the length of the anchor fell below 15 characters. If the algorithm could not find a location for the annotation, it *orphaned* the annotation. Orphaned annotations were displayed at the top of the annotation index (see Figure 2).

This algorithm is fairly simple. It does not take into account surrounding context or search for the anchor text in a more sophisticated manner, and it weighted the center words of anchor text more heavily than the words toward the beginning and the end. We decided to use this algorithm to gather observations of user expectations before developing a more complicated algorithm. We expected the algorithm to fail often so that we could be alerted to scenarios where

participants were most unhappy with the algorithm’s performance.

5.2 Experimental Method

For this study, 12 participants were recruited in the same manner as the first study. Participants were first given a brief training task to familiarize themselves with the system, and then given the task of annotating a document so that it “could be skimmed quickly by a busy executive.” The document was a general interest news article from the web. Next, participants were told that an updated version of the document was available, but that rather than repeating the task of annotating the document, they would have the computer transfer their annotations from the old to the new document. Participants then examined each annotation and rated its position in the new document on a 7-point scale where 7 was “perfect”, 4 was “ok”, and 1 was “terrible.”

In this study, because participants made their own annotations, we needed to create an updated version of the document before the study with modifications that would affect participant’s annotations. To do this we had a few pilot study participants annotate the document (on paper). Then we made changes in the original document in places where people tended to make annotations. A second updated version was created by a colleague unfamiliar with the annotation positioning algorithm. If participants quickly finished the rating task using the first updated version, we had them repeat the task for the second updated version.

5.3 Results

The main purpose of this study was to examine participant satisfaction with the algorithm’s attempt to reposition annotations in the updated document. The 12 participants made a total of 216 annotations and then rated their satisfaction with how each annotation was positioned in the first updated version. Half the participants also rated the positions of their annotations in the second updated version. A total of 302 position satisfaction ratings were collected.

We present participant position satisfaction ratings in the following sections by breaking down the set of 302 ratings into three logical groups based on the changes made to an annotation’s anchor text:

- **Same:** Annotations anchored to text that did not move or change.
- **Move:** Annotations anchored to text that was moved from one portion of the document to another, but that otherwise did not change.
- **Complex:** Annotations anchored to text that was changed and possibly moved.

We expected high satisfaction ratings for the transfer of annotations in the **Same** group because our algorithm finds all such annotations. For annotations in the **Move** group we still expected fairly high ratings, since the algorithm also finds these annotations. However, we believed that if the anchor text moved significantly in the document, this would

change its surrounding context, and perhaps render it irrelevant. In this case, participants might prefer the annotation to be orphaned.

For annotations in the **Complex** group we expected lower scores due to the simplicity of the algorithm. We expected instances where participants would be unsatisfied with how much of an annotation’s anchor text the algorithm found, or that an annotation had been orphaned. We also believed that participants would always rate annotations that were found higher than annotations that were orphaned, except when the orphan was caused by deletion of the entire anchor text.

5.3.1 Same: When Anchor Text Does Not Change

Although our algorithm is simple, it is guaranteed to find annotations attached to unique text that does not move or change. 47 out of 302 position ratings fell into this category. As we expected, the median participant rating for these annotation positions was a perfect 7.0. When the text doesn’t move or change and the system finds the entire annotation anchor text in the new document, participants are happy.

5.3.2 Move: When Anchor Text Moves

121 of the position ratings were for annotations attached to anchor text that was moved in the updated document, but not changed in any other way. We focused on move modifications that were noticeable to a human reader. For example, a paragraph might have been moved from one page to another. 100% of annotations attached to text that moved, but did not change, were found in the updated document. This was due to our algorithm’s use of simple text matching to find an annotation’s anchor text and the fact that participants attached their annotations to unique sections of text. The median participant rating for these annotation positions was 7.0.

The high ratings given for these annotation positions surprised us somewhat. We expected that if the text an annotation was attached to moved significantly, there would be times when an annotation would lose relevance and need to be orphaned. However, the data indicate that this is not the case. Thus, perhaps the surrounding context of an annotation is of lesser importance when considering factors that contribute to keeping participants satisfied with automated annotation positioning. It would be interesting to explore whether users feel the same way about the surrounding context for editing and margin annotations.

5.3.3 Complex: When Anchor Text is Modified

134 of the position ratings were for annotations attached to text that was changed in some way in the updated document. Of these annotations, our algorithm successfully transferred 71 and orphaned 63. Note that a piece of text may have been both changed and moved, but since data in the previous section indicate that ratings are independent of moves, we focus primarily on how the anchor text changed.

Modification Score	Rating (number of annotations)
1	1.50 (12)
2	3.0 (18)
3	3.0 (7)
4 (total delete)	7.0 (25)

Table 2: Median participant position satisfaction ratings, on a 1 to 7 Likert scale, for annotations where the anchor text changed and the annotations were not found (orphaned). As the amount of modification to the anchor text increased, participants were more satisfied that the annotation had been orphaned.

To analyze this set of annotations, we classified the changes that were made to an annotation’s anchor text. Sometimes just one word was changed, and sometimes the entire sentence was rewritten. Changes were coded using the six “delete” and “reword” categories outlined in Table 1, and these encodings were used to compute a modification score for each annotation. Minor rewords and minor deletes were given one point and medium rewords and medium deletes were given two points. Using this scheme, higher scores indicated more drastic changes, with a highest possible combined modification score of 3. Total deletes were treated as a separate category and automatically given a score of 4. Total rewords were eliminated from the analyses because only one such case occurred.

Reliability of these classifications were verified by having a colleague not involved with the research code a representative sample of the anchor text changes. Inter-rater reliability for the modification score was high ($\alpha = .90$).

5.3.4 When Annotations are Orphaned

Table 2 shows the median position ratings for annotations that were orphaned in cases where the text changed. The overall median score for this set of annotations was 5.0. As we expected, the table shows that participants gave the lowest ratings when little modification occurred to the text and the annotation was not found. In fact, participant ratings were significantly correlated at .72 ($p < .001$) with modification score. Thus, ratings increased as more modifications occurred, to the point where participants gave the highest ratings to orphaned annotations when all of the text the annotation had been attached to was deleted.

Comments that participants made while rating orphaned annotations also support the hypothesis that as the amount of text change increases, people are more satisfied when the annotation is not found. For one annotation, a participant told us that the document “changed around enough and the keywords left out of the second article, I could see it might not find that.” Of another annotation, a participant observed that the modifications “redid [it] entirely...makes sense they [the algorithm] didn’t find that one.”

Modification score	1 to 24% found	25 to 49% found	50 to 74% found	75 to 100% found	Overall
1	3.0 (3)	3.0 (13)	3.0 (19)	6.0 (18)	4.5 (53)
2	2.0 (9)	3.0 (6)	4.0 (1)	5.0 (1)	3.0 (17)
3	-	3.0 (1)	-	-	3.0 (1)
Overall	2.5 (12)	3.0 (20)	3.5 (20)	6.0 (19)	4.0 (71)

Table 3: Median participant position satisfaction ratings for annotations where the anchor text changed and some percentage of it was found. Participant satisfaction is directly correlated to the amount of anchor text found and inversely correlated to the amount of modification that occurred to the anchor text. Number of annotations in each case is in ()'s.

5.3.5 When Anchor Text Changes and Annotations are Found

Table 3 shows the median position ratings for annotations that were found in cases where the anchor text changed. The overall median score for this set of annotations was 4.0. Note that a successful annotation transfer includes cases where only part of an annotation could be transferred. For example, if a person made the following highlight annotation:

The quick brown fox jumped over the lazy dog.

Below is an example of modified text and the partial anchor text our algorithm would have found:

The quick fox jumped away from the dog.

To take into account partially found annotations, we also examined this set of annotations by looking at what percentage of the annotation anchor text was found in the modified document. These percentages are listed in the columns of Table 3.

The data in Table 3 suggest two trends. First, not surprisingly, the greater the percentage of the annotation anchor text found, the more satisfied people are (bottom row of Table 3, read left to right). Percentage of annotation anchor text found was significantly positively correlated at .65 ($p < .001$) with participants' rating. Second, and somewhat counterintuitive, the more drastic the modifications to the anchor text, the less satisfied people were when the annotation anchor was found (right column of Table 3, read top to bottom). Modification score was significantly negatively correlated at $-.34$ ($p < .003$) with participant rating. This was unexpected. We thought that participants would be more impressed with the system when it was able to find annotations even when significant changes occur, but this was not the case.

Finally, somewhat surprising was the participant's median ratings of 3 for both found and orphaned annotations with modification scores of 2 & 3 (see Tables 2 & 3). We had expected found annotations to always be rated higher than orphans not caused by a total delete of the anchor text.

6 DISCUSSION

The results from our studies provide valuable insight for designers of annotation systems.

6.1 Surrounding Context is Less Important

As noted previously, robust anchors can be created by storing an annotation's surrounding context and anchor text information. We were surprised when our studies indicated that users may not consider surrounding context very important for annotations with range anchors (even though it may be crucial for annotations with margin anchors).

We observed rather casual text selection where annotation boundaries were influenced by document formatting (for example, ends of lines) similar to Marshall's observation for annotations on paper [9]. We thought this might cause participants to expect the annotation transfer algorithm to perform a more sophisticated search for the correct text when it was deleted, but the data do not support this. Participants gave very high position ratings for annotations attached to text that was significantly moved, and for annotations that were orphaned due to the original text being deleted.

This does not necessarily mean that robust positioning algorithms should not save surrounding context. Rather, users may not consider it very important, so it should perhaps be weighted less heavily in algorithms that do employ it. Future research should examine whether this finding was due to our focus on active reading annotations instead of other types of annotations, such as editing.

6.2 Focus on Keywords

When examining the particular cases where participant ratings were low, we found that participants often expected the system to do a better job locating key words or phrases. Comments included:

- "The key words are there, it should have been able to somehow connect that sentence [in the modified version] with the original"
- "Should have gotten that one, at least the quote."
- "Should have at least highlighted the name."
- "Doesn't pick up a change in wording that means essentially the same thing."

Thus, when designing robust positioning algorithms, it may be helpful to pay special attention to unique or "key" words in the anchor text, as the ComMentor [16] system does. Participants also appear to consider names, and quotations as particularly important. A simple thesaurus or grammar parser may additionally be useful to recognize when simple rewords have occurred that do not change the semantics of the sentence.

6.3 Orphan Tenuous Annotations

Based on Tables 2 and 3, two trends seem to emerge. First, if an annotation is found, users initially assign the highest rating and then move down the satisfaction scale based on

how much of the annotation anchor text the algorithm found and how many modifications occurred. For orphaned annotations the process works in reverse. Participants start with the lowest rating and then move up the scale as more modifications are noticed, or when they realized the entire anchor text has been deleted.

These trends suggest that there may be a point when, even though an algorithm may be able to find a highly likely location for an annotation in a modified document, the participant would be more satisfied if the annotation was orphaned. Further testing this hypothesis is a good area for future research.

6.4 Include user intervention

If indeed systems choose to orphan some annotations even when they have a relatively good guess as to where annotations should be positioned, it may be helpful to provide users with a “best guess” feature that shows them where orphaned annotations might be located. This feature may also be helpful for situations where users need to insure all annotations are moved to a modified version of the document. Some of the system’s “best guesses” may not be correct, but they may provide enough information for a user to easily reattach orphaned annotations.

7 CONCLUDING REMARKS

The primary contribution of this paper has been to explore what users expect when they make annotations on a digital document that is subsequently modified. The paper also presents a framework for building robust annotation systems that requires us to ascertain the relative importance of surrounding text vs. anchor text, as well as the kinds of anchor text information that is more important to users than others.

For the types of annotations studied, our results suggest that participants paid little attention to the surrounding context of an annotation, and algorithms may want to give the surrounding context relatively little weight when determining an annotation’s position. As for anchor text information, participants’ comments stressed the importance of key words, proper names and quotations. We also found in certain cases, even when part of the annotation’s anchor text is found, users may prefer that the positioning algorithm does not place it in the modified document. The detailed data we collected are useful for determining potential thresholds for orphaning annotations.

While our results have revealed valuable information about user expectations and will help us design more robust annotation positioning algorithms, much work remains. Future studies should explore how our results apply to other types of annotations (such as editing annotations and margin annotations) and to other types of documents.

ACKNOWLEDGMENTS

We thank Molly Brown, Mike Brush, Duncan Davenport, and Scott LeeTeiernan for their assistance.

REFERENCES

- [1] Adobe Acrobat, <http://www.adobe.com/products/acrobat/main.html>
- [2] Aladdin Ghostview, <http://www.cs.wisc.edu/~ghost/>
- [3] Cadiz, J., Gupta, A., Grudin, J. Using Web Annotations for Asynchronous Collaboration Around Documents. To appear in Proceedings of CSCW 2000 (Philadelphia, PA, Dec. 2000)
- [4] Davis, and Huttonlocker. CoNote System Overview. (1995) Available at <http://www.cs.cornell.edu/home/dph/annotation/annotations.html>.
- [5] E-Quill. <http://www.e-quill.com/>
- [6] Grønbaek, K., Sloth, L., Ørbæk, P. Webvise: Browser and Proxy Support for Open Hypermedia Structuring Mechanisms on The WWW, Proceedings of the Fifth International World Web Conference, (Toronto, May 1999), Available at <http://www8.org/w8-papers/3a-search-query/webvise/webvise.html>.
- [7] Laliberte, D., and Braverman, A. A Protocol for Scalable Group and Public Annotations. 1997 NCSA Technical Proposal, available at <http://union.ncsa.uiuc.edu/~liberte/www/scalable-annotations.html>.
- [8] Marshall, C.C. Annotation: from paper books to the digital library, Proceedings of Digital Libraries '97 (Philadelphia, PA, July 1997), ACM Press 131-140
- [9] Marshall, C.C. Toward an ecology of hypertext annotation, Proceedings of HyperText '98 (Pittsburgh, PA, June 1998), ACM Press, 40-48.
- [10] Microsoft ebook Reader, <http://www.microsoft.com/reader/>
- [11] Microsoft Office 2000 Web Discussions, <http://officeupdate.microsoft.com/2000/focus/articles/wWebDiscussions.htm>
- [12] NotePals, <http://guir.berkeley.edu/projects/notepals/>.
- [13] Ovsiannikov, I., Arbib, M., McNeill, T. Annotation Technology, Int. J. Human Computer Studies, (1999) 50, 329-362.
- [14] Phelps, T., Wilensky R. Multivalent Annotations, Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries (Pisa Italy, Sept. 1997) Available at <http://www.cs.berkeley.edu/~phelps/papers/ed97-abstract.html>
- [15] Phelps, T., Wilensky R. Robust Intra-document Locations, Proceedings of the 9th World Wide Web Conference, (Amsterdam, May 2000) Available at <http://www.cs.berkeley.edu/~phelps/Robust/Locations/RobustLocations.html>.
- [16] Roscheisen, M., Mogensen, C., Winograd, T. Shared Web Annotations as a Platform for Third-Party Value-Added, Information Providers: Architecture, Protocols, and Usage Examples, Technical Report CSDTR/DLTR (1997), Stanford University. Available at <http://www-diglib.stanford.edu/rmr/TR/TR.html>.
- [17] Schickler, M.A., Mazer, M.S., and Brooks, C., Pan-Browser Support for Annotations and Other Meta Information on the World Wide Web. Proceedings of the Fifth International World Wide Web Conference (Paris, May 1996), available at http://www5conf.inria.fr/fich_html/papers/P15/Overview.html.
- [18] ThirdVoice web annotations. <http://www.thirdvoice.com/>
- [19] Wilcox, L.D., Schilit, B.N., and Sawhney, N. Dynamite: A Dynamically Organized Ink and Audio Notebook, Proceedings of CHI '97 (Atlanta, Georgia, USA, March 1997), <http://www.acm.org/sigchi/chi97/proceedings/paper/ldw.htm>.