READR: A RST Driven Text Processing Platform for Reading Comprehension

READR: A RST Driven Text Processing Platform for Reading Comprehension

A THESIS PRESENTED
BY
JAKE CUI
TO
THE DEPARTMENT OF COMPUTER SCIENCE AND
THE DEPARTMENT OF LINGUISTICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF ARTS (HONORS)
IN THE SUBJECTS OF
COMPUTER SCIENCE AND LINGUISTICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS
MARCH 2020

Thesis advisor: Elena Glassman Jake Cui

READR: A RST Driven Text Processing Platform for Reading Comprehension

Abstract

Human language is built upon underlying structure, yet the text we are used to reading is displayed in a flat body with minimal context. I present *READR*, a tool that can identify and highlight hidden structures in text (like examples and elaboration) to augment reading comprehension in realtime. *READR* classifies sentences based on the linguistic Rhetorical Structure Theory (RST) by deep learning a neural embedding across >10000 labeled corpora. Given any target text document, whether it be a research paper, Wikipedia article, or textbook, *READR* presents an interactive interface in which users can quickly skim and process text based on rhetorical function. I show that *READR* has the potential to improve comprehension and reduce reading time in a preliminary user study with five participants.

Contents

Listing of figures

This thesis is dedicated to my parents.

Acknowledgments

Thank you so much to my wonderful thesis advisor, Professor Elena Glassman, for her endless mentorship and support throughout this thesis. Elena shared not only her expertise and research intuition, but also her love for HCI. Because of her, this thesis was highly instructional and more importantly, a joy to conduct and write. I would also like to thank Professor Diti Bhadra, who fostered my love of linguistics and first introduced me to semantics and discourse theory. I also would like to thank Professor Shafiq Joty for his help in the RST parsing component of this project. I also owe a thanks to thank my loyal friends, who supported and encouraged me throughout this project.

"The world was hers for the reading."

Betty Smith

1 Introduction

In an increasingly digital era, most text we consume is via screen. From social media posts to research journals, information is more frequently being posted online rather than in print. The prevalence of online text media has resulted in an emphasis on rapid comprehension of complex information - users want to be able to read fast [1].

Past research has shown that online readers have learned to optimize their reading behavior; users tend to direct their attentions to the most important areas in text. They continuously read through until the rate of information gain drops below a certain threshold, at which point they skip to the following section [2]. Faster, selective, reading strategies are widespread and highlight a user need centered around identifying the most important and relevant areas in text.

All meaningful discourse has underlying structure that readers will try to recreate in their own mental models. For example, paragraphs are a simple way to introduce a basic structure to text. Sentences that are relevant to one another can be grouped together in paragraphs. Readers know to expect some degree of topic shift when moving from one paragraph to another. Generally, if information is presented in a way that is more similar to the way we store information cognitively, it will be easier to read and comprehend [3].

In this thesis, I present an intervention in the form of a text reading platform that automatically processes and displays text based on its underlying linguistic structure. By presenting text in this structured manner, I hypothesized that users would be able to internalize a mental model of the discourse more intuitively. The goal of this thesis was to explore techniques to augment the online reading experience; I wanted to increase reading comprehension while simultaneously reducing reading time.

1.1 Motivation

The motivation for this thesis is rooted in Rhetorical Structure Theory (RST), a newly developed linguistic theory of modelling discourse. RST was created in 1988 by computational linguists for text generation applications. However, a plethora of additional use cases have been identified, including but not limited to text summarization, information theory, and coherence [4]. RST arranges discourse in a hierarchical tree; this rigidly defined structure is what allows it to be useful in the generation of one's mental model while reading. I will discuss RST in greater detail in Chapter 2.

RST is particularly useful for this thesis because of the amount of computational research surrounding it. There are an abundance of parsing algorithms that have been developed throughout the past decade [5]. These recent advances are what put a reading augmentation tool in the realm of possibility and may explain why such a tool does not currently exist. This thesis aims to use RST parsing as the foundation of an intuitive and usable text reading system.

1.2 Overview of Contributions

My primary contribution is *READR*, a functional, proof-of-concept, text reading application that is able to dynamically process any raw text body [1]. Using RST tree output generated by CODRA, an out of the box, intra-sentential parsing algorithm [6], I introduce a reading platform that automatically highlights and organizes text in a way to make it easier to read (Figure 1.2.1). To my knowledge, this is the first platform of its kind to leverage discourse structure to create a reading augmentation tool. Additionally, I performed five user studies on college students to evaluate the usefulness and usability of the platform. My findings suggest that a tool of this nature may be able to reduce reading time, increase reading comprehension, and minimize cognitive load. I encourage the reader to open up and play around with the demo link while reading this thesis.

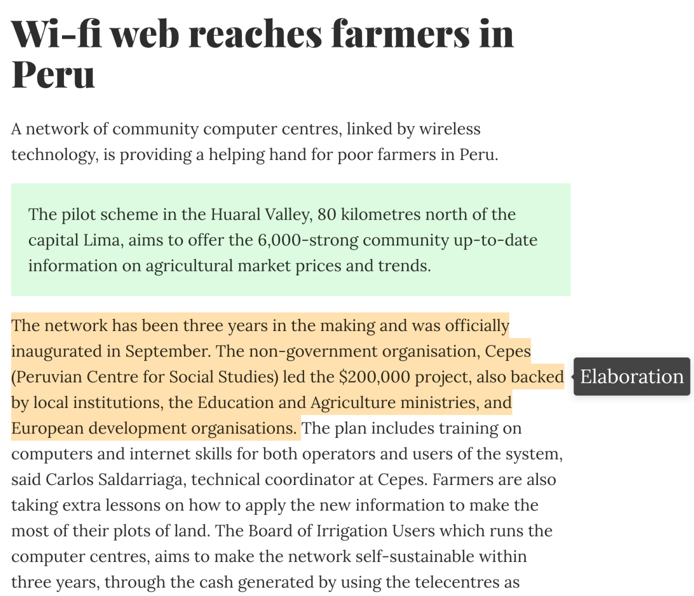

Figure 1.2.1: An example document automatically annotated and generated.

[1]Demo available at https://jake-cui.github.io/thesis-demo/

2

Background and Related Work

2.1 Cognitive Load Theory

A key motivating factor for this project is the idea of cognitive load theory (CLT). Developed in the 1980s by John Sweller, the concept hypothesizes that humans are able to best process and understand information when it is presented in a way that is optimized for human cognitive structures [7]. Sweller initially introduced the idea while studying problem solving tasks, but additional research has explored how CLT can affect reading comprehension specifically. While attempting to read and understand text, humans are limited by their working memory capacity. All humans have a limited quantity of information they can store temporarily for real-time processing. To minimize the amount of effort and time required to understand a body of text, we should reduce the required working memory load [8].

First, it is important to understand how humans tend to perceive text and determine if it is possible to generalize a mental model of text comprehension. Past research has identified that humans will construct a mental model of not only the text and words they read in a sentence, but also of the situations being conveyed. Zwaan concluded that mental representations of text hinge upon three main categories: causality, spatiality, and temporality. It is easier for humans to store information relative to one another rather than as individual units (e.g. it is easier to remember a story with sequential events rather than memorizing a list of independent occurrences) [3].

To begin tackling the problem of minimizing cognitive load, researchers have identified three main types of mental load: intrinsic, extraneous and germane load. *Intrinsic mental load* is caused by complexity in the subject matter itself and it is impossible to reduce. *Extraneous load* is introduced by the way information is presented and can be reduced through organization and design. *Germane load* is introduced via the generation of mental models and cognitive processing of information. Germane load is actually seen as a positive influence for learning and it may not always be the best to minimize. Past research has indicated that mental restructuring and organization of information facilitates learning [9]. Cognitive overload occurs when the sum of these three forms of load is greater than the working memory of the reader [10]. In the context of this project, we will be experimenting with different methods to minimize extraneous load, which is most directly correlated to the way information is presented visually.

Information theory has identified two key characteristics of written text that affect reading comprehension. The first is information structure, which is the way that concepts are grouped and organized in text (for example, talking about a group of ideas that are similar before moving onto a different one). A readable information structure (e.g. a continuous story) allows readers to follow along easily and more intuitively generate an accurate mental model. The second critical component is information display, the way that information is arranged visually. Information display facilitates understanding and mental model generation by more explicitly highlighting characteristics of the text. For example, arranging discourse into paragraphs is a straightforward and common way to emphasize relationships between sentences.

Information display can highlight and intuitively emphasize important areas of text through visual cues (e.g. bullets, highlighting, font size). This is particularly relevant in use cases in which users want to read something quickly. Instead of reading linearly, users will typically skim around, observing headers and section labels to identify the topics they are most interested in [1]. Simplifying this process will be one of the key goals of this project.

2.2 Rhetorical Structure Theory

Rhetorical Structure Theory (RST) is a descriptive theory of the organization of natural text. Well written natural text is not a series of independent thoughts, rather a sequence of related and structured sentences. Natural language has a *coherence structure* in which sentences and clauses are bound to one another to express meaning as a whole [11]. Prior to the development of RST, linguists have taken several different approaches to explaining coherence. These theories followed the idea of dynamic semantics. Previous frameworks which treated discourse as following a set of rigid models [12]. Instead, dynamic semantics views discourse as the relations between contexts, known as *context change potential* [13]. Martin [14], Knott and Dale [15] proposed breaking down natural text by the use of connectives (such as: furthermore, but, hence, etc).

Another prevalent model of discourse coherence is Segmented Discourse Representation Theory (SDRT), proposed by Asher and Lascarides. Briefly, SDRT segments discourse into a set of labels, each one with a unique function classification under the framework. Then, each unit as assigned a rhetorical relation (e.g. *Explanation* or *Contrast*) between its adjacent labels [16]. This idea of segmenting text to basic units and generating relationships also is prevalent in RST. A key reason RST was selected for this project over SDRT is its use of nuclei and satellites, which I will explain below [17].

RST expresses discourse structures through discourse trees (DTs), hierarchical graphs in which each leaf represents a unit of discourse. Some say that RST was the most influential theory in computational linguistics. As mentioned, RST was initially developed for text generation but eventually was expanded to be a popular framework for parsing the structure of natural text [6].

2.2.1 Definition

RST was initially formulated by William Mann and rigorously defined by Mann and Thomson in 1988 [4]. They are credited to be the first to make RST explicit enough to be usable and open to examination. In order to effectively utilize RST to identify text importance, we first must understand each of the components of the theory.

RST provides a comprehensive analysis of text (all clauses are accounted for) rather than selective commentary. Mann and Thompson introduce the four key objects in RST:

1. Relations
2. Schemas
3. Schema applications
4. Structures

These objects serve as building blocks for one another, with Relations serving as the smallest unit (i.e. multiple relations form schemas, schemas form schema applications, etc).

Relations

To begin, all natural text in RST is broken up into a series of clause-like units that function as the building blocks of the entire discourse. Linguists have referred to these units as *Elementary Discourse Units (EDUs)*. Relations connect two non-overlapping EDUs and provide some information regarding how the two discourse units interact. The twelve categories of relations are shown in Table 2.2.1.

Circumstance	Antithesis and Concession
Solutionhood	Condition and Otherwise
Elaboration	Interpretation and Evaluation
Enablement and Motivation	Restatement and Summary
Evidence and Justify	Sequence
Relations of Cause	Contrast

Table 2.2.1: The twelve relations defined by Mann and Thompson.

An essential characteristic of RST is the classification of all EDUs into a nucleus or satellite (N or S respectively). Relations will always attach a nucleus to a satellite. Each relation will specify a different set of judgements that the reader will be able to make regarding each of the EDUs. Additionally, the definition includes a set of constraints on the nucleus and satellite individually, which contextualize the content. These judgements are all semantically significant, and one of Mann and Thompson's example definitions is shown in Table 2.2.2

Relation name: EVIDENCE
Constraints on N: R might not believe N to a satisfactory degree
Constraints on S: R believes S or will find it credible
Constraints on the N + S combination: R's comprehending S increases R's belief of N
The Effect: R's belief of N is increased
Locus of Effect: N

Table 2.2.2: The relation definition for EVIDENCE.

Each relation is defined semantically as shown above. Each one of these definitions provides some insight on the function of each text snippet relative to the meaning of the entire discourse. These rigorous definitions also provide a simpler, checklist style way to classify different relations based off of raw text. It is also worth noting that Mann and Thompson intended their list of relations to be open to interpretation. Future RST studies often modify and expand on this list.

Schemas

Schemas specify a general format in which relations are allowed to occur. With the five organizational structures shown below, these schemas specify how nuclei can relate to their respective satellites. Figure 2.2.1 [18] illustrates how the different kinds of relations can occur with one another. Each of the curved lines represent a category of relation as shown in its label. The straight lines are identifying nuclear spans. Schema 1 in the Figure 2.2.1 is by far the most common structure of relations. Every other relation not shown in the figure follows this same structure in which a single satellite points to a nucleus.

Schema Applications

Schema applications allow for slight variations in the general schemas listed above.

1. **Unordered Spans:** These schemas do not restrict the order in which the nucleus and satellite must appear. For example, a satellite that is ELABORATING on a nucleus may come either before or after the nucleus.

2. **Optional Relations:** For schemas that may have multiple relations, all individual relations are optional. For example, a JOINT relation can relate 2+ units.

3. **Repeated Relations:** A relation in a schema can be applied repeatedly in the application of the schema. For example, ELABORATION can be applied repeatedly.

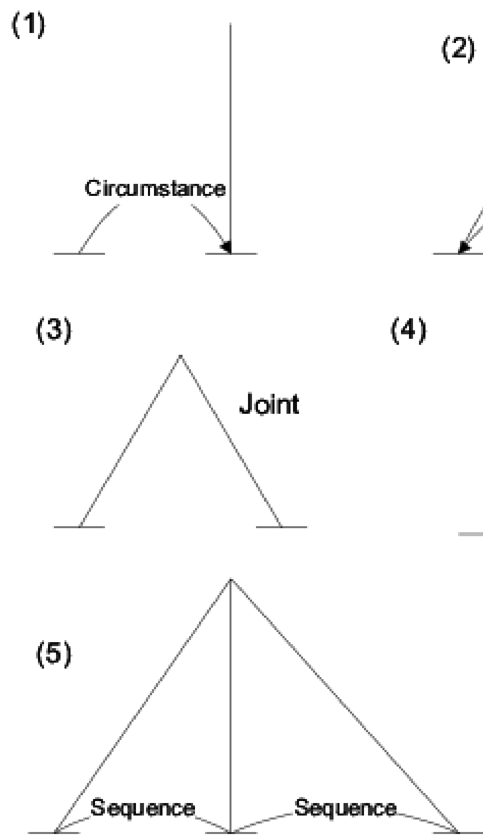

Figure 2.2.1: The five schema introduced by Mann and Thompson.

Structures

The final step in generating RST trees for a text body is applying schema applications to the discourse units to form a final structure that will meet each of the constraints listed below. It is worth noting that Mann and Thompson do not provide a formal definition for the discourse units. They claim that unit size is arbitrary and can be up to the discretion of the RST application. They advocate for a theory neutral classification of independent units. In their experience, Mann and Thompson essentially used clauses to represent discourse units.

1. Completeness: The schema applications must create a complete tree that spans the entire body of text
2. Connectedness: All text units must be part of the same tree structure. There will be one large text span that encompasses the entire input discourse.
3. Uniqueness: Each schema application cannot overlap with another. Each relation must apply to different text spans.
4. Adjacency: Schema applications can only be performed with text spans that are directly adjacent.

An example of a complete RST structure is shown in Figure 2.2.2 [19].

2.2.2 Parsing

Mann and Thompson's RST definition laid the foundation for more rigorous development of the theory. In particular, there has been strong research interest in attempting to create machine learning parsers that can automatically extract RST structure from raw text input. Several different algorithms have arisen with F_1 scores ranging from 50 - 80% accuracy [5].

Carlson and Marcu created a RST corpus that has served as the foundation for these RST parsing applications. The corpus contains 385 RST annotated articles

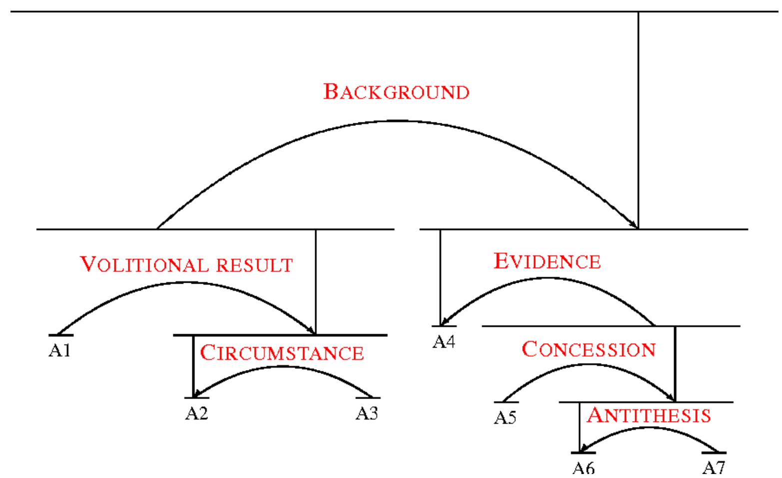

Figure 2.2.2: Example RST with seven discourse units.

in the Penn Tree Bank with a slightly modified definition inspired by Mann and Thompson [20].

In this thesis, we worked with Shafiq Joty and used his highly successful RST parser: CODRA [6]. The application name stands for a 'COmplete Probabilistic Discriminative framework for for performing Rhetorical Analysis'. CODRA has two main components: a discourse segmenter and a discourse parser. The segmenter is based on a binary classifier that breaks up a body of text into elementary discourse units for RST analysis. The parser than infers probabilities for the likelihood of each type of relation and schema, ultimately generating a complete RST tree [1].

For more detailed information on the parser itself, I recommend reading the CODRA website and demo referenced above. There are a few key characteristics of CODRA to note with regards to my thesis. For one, the parser does not rigidly adhere to Mann and Thompson's definition of various relations. Instead of using the 25 introduced by the original definition, Joty used 16 coarse categories described by Carlson [20] in order to adhere to the structure of the RST

[1] A CODRA demo is available at `http://alt.qcri.org/demos/Discourse_Parser_Demo/`

treebank.

Another notable factor is that CODRA performs two levels of sentence parsing: intra-sentential (sentence level) and multi-sentential (document wide level) parsing. CODRA is one of the first to distinguish between these two levels and reported significant increases in prediction accuracy. I chose to utilize CODRA for this thesis because it significantly out-performed state of the art parsing algorithms by a wide margin.

2.3 Related Work

There have been several attempts in the past at software interventions to improve the readability of digital text. Academics have long debated what characteristics make text easier and more intuitive to read.

2.3.1 Typography

One of the most heavily researched aspects of text in the context of readability is typography: the art of arranging written language to make it more readable. In a 2016 study, Rello examined how different font sizes and line spacing on Wikipedia [2] affected reading comprehension in a over 100 participants. They concluded that text heavy sites should be using font sizes of 18 and above in order to maximize readability [21].

In a 2011 study, Banerjee investigated a series of different font types and sizes to determine which was best at minimizing mental workload when reading [22]. An interesting takeaway from this study was that the font that optimizes for reading speed is not necessarily the one that will minimize mental workload.

These typography studies are highly relevant to this thesis because they all focused on quantitatively evaluating readability and text comprehension of human subjects. I will be heavily leveraging the same user study methods to evaluate my intervention. More details on the actual study itself will be explained

[2]https://www.wikipedia.org/

in the following chapter.

2.3.2 Contract Readability

Contracts have historically been text heavy and extremely cumbersome to read. In 2015, Passera investigated how various layouts, typography, and visual cues could impact the readability and mental load of reading contracts [8]. Again, they evaluated how these visual interventions affected comprehension. This paper was particularly relevant to this thesis because it dove deeply into the information theory of text. More specifically, it went a step beyond only typography and considered visual layout and information organization. The techniques used to evaluate these specific features were also leveraged in this thesis.

2.3.3 Skim Reading

With the rise of the internet, recent studies have focused on 'skim reading' specifically, or the act of comprehending as much information as possible within a short time frame. This type of reading is most similar to how users tend to consume written information online.

In a 2017 study, Lee introduced a software intervention for skim reading by creating transparent overlays of important information while a user is scrolling at high speeds [23]. They created a novel web reading platform in which highly relevant figures and sentences would remain constantly overlayed on the screen while users continued to scroll. Comprehension rates remained relatively unaffected, but the intervention allowed users to read more quickly and significantly reduced look up performance (when users had a specific piece of information they were to find and recall). In another 2013 study, Duggan investigated how individuals tend to skim articles by analyzing an eye tracking dataset [2]. The goal was to better understand how individuals skim content online in order to display text in a more readable manner. This is how they arrived at the text reading pattern explained in Chapter 1. These insights and behaviors helped drive the initial needfinding and user interviews described in the methods.

3
Methods

3.1 Needfinding

At the start of this thesis, I went in with the broad goal of improving the user experience of reading text online. There are many factors to consider when optimizing readability, including but not limited to: reading time, cognitive load, and recall.

I decided to conduct a series of casual needfinding interviews. There were five participants, each of which were randomly selected undergraduates at Harvard. The interviews were framed as a general discussion in which participants were asked to describe both the type of content they typically read online (e.g. news articles, books, research papers, social media posts) as well as the general approach they take to reading on these mediums. Afterwards, as a part of the user study described below, users were observed performing a series of simple comprehension tasks.

3.1.1 Time

By far, the most important factor to consider while reading online is time, regardless of the nature of the text itself. Participants remarked that whether it was a formal article or a social media post, they would always check to see the length of the natural text. Regardless of whether or not the user was required to read the text [1], most begun by checking the length. This can be done explicitly by quickly scrolling through the entirety of the text or also by seeing how far the scroll bar moves on the side of the screen when scrolling. Before reading any words at all, users tended to estimate how long it would take to read through a natural text body and then decide whether or not to click off. In scenarios in which users have already decided that they intend to read a text body, they will check for length regardless. Readers can utilize the text length to then decide their approach to reading (e.g. deciding just to skim through a longer article).

3.1.2 Lookups

One of the most useful features of online reading is the ability to explicitly search for keywords using control/command-f. Readers who claimed to enjoy reading online rather than on paper cited this as one of their primary motivations. Oftentimes, when users have a specific question they are striving to answer while reading, a majority of text on articles will not be relevant at all. Thus, they will prefer to jump directly to a section of interest, effectively reducing the fraction of the text that they will have to actually read. In situations in which an explicit search function is not available, readers will still attempt to quickly skim the article while searching for keywords that ideally correspond to their section of interest.

[1] For example, users may be required to read a paper for a homework assignment. In these scenarios, readers will not have the option to click away

3.1.3 Cognitive Overload

Although this is bound to vary depending on the reading task and the user, participants claimed to be generally averse towards heavy cognitive overload. Especially online in which so many resources are available, users have an extremely high sensitivity towards cumbersome reading. For example, a user remarked that when skimming through Google search results, if they encounter a link that is particularly difficult to read, they will not hesitate to click out and explore a different link. This shows that users are not interested in only optimizing for time, but also to reduce necessary mental workload.

3.2 Problem Definition

The goal of this project was to create an online platform that allows users to read digital text more quickly while also enabling them to understand important material more comprehensively. The target stakeholder is any consumer of online media. I intentionally kept the user group broad because I wanted to create a versatile intervention that would be potentially applicable in several different use cases.

The core tension that I observed was that online readers want to go through text as quickly as possible without sacrificing comprehension. However, there are no significant structures or interfaces in place that facilitate intuitive reading and understanding.

3.2.1 Existing Solutions

A majority of online readers will just read raw text from a screen (e.g. a New York Times article on a website). There are a few common tools used to increase readability of such texts, such as leveraging typography to highlight important information and headlines. Additionally, users can use search functionality in their browsers and devices to quickly find keywords and relevant sections. While this kind of online reading is satisfactory, especially considering how widespread it has become, users still find it to be time consuming and cumbersome.

3.2.2 Obstacles

I consulted existing literature and my needfinding interviews to determine where the major gaps were in the current online-reading user experience. There were a few key obstacles identified that would guide the general design of this platform:

1. Readers are naturally impatient. It is important that they are able to extract relevant information as quickly as possible.

2. Readers are often interested in only a small fraction of text. The rest of the material is just distracting and time consuming to skim through.

3. Comprehending text requires a heavy mental workload. Depending on the context, users can get frustrated and/or tired while reading.

4. Paragraphs are usually structured around key topic sentences which helps readers create a mental model of the text. However, the structure of these sentences varies and requires cognitive load from users to decipher.

To me, these areas are most heavily impacting how users are reading online today. Any potential solutions that address these obstacles will ultimately lead to a more streamlined reading user experience.

3.2.3 Design Axioms

When considering potential solutions, I started by orienting myself around a set of key constraints in order to create an application that would actually be useful.

For one, I wanted to minimize the learning curve for new users on the platform. Given that one of the key obstacles in this project is time constraints, it is important that users can ramp up and understand the interface as quickly as possible. Given the short attention spans of users while reading online, I wanted to make the features of the interface intuitive and easy to pickup immediately.

Secondly, it was important that the interface was relatively lightweight and unobtrusive. I did not want to attempt to redefine how people are used to reading. The goal here is to create a tool that will augment how users are currently reading rather than creating an entirely new protocol.

Additionally, I wanted to create a flexible platform that users could interact with to best meet their unique needs. I hypothesized that different users and text documents will necessitate various types of text processing for optimal reading. Therefore, I wanted to address these varying use cases by giving users control over how text is displayed on the platform.

Finally, it was important that the interface designed would be versatile. I wanted to be able to experiment with various types of digital texts, ranging from easy to read to highly dense (e.g. news articles and research papers). This thesis is meant to be exploratory in nature and preliminary user studies will hopefully guide future continuation projects.

3.3 READR

In this thesis, I present *READR* (RST Enhanced Assisted Dynamic Reader), a functional, proof-of-concept, text reading application that is able to dynamically process any raw text body to promote readability. If you have not yet, I highly recommend trying out the demo available here[2].

READR uses RST to parse and automatically extract key structures out of raw text. RST has a rich underlying structure that is even more detailed than our internal mental models. RST structure can be applied and modified to create structures that will directly map to how we internally understand text. I hypothesized that this would reduce cognitive workload of users by more easily allowing readers to generate a mental model of the text. The platform was built out in React.js with a Python backend[3]. I decided on three main interventions, sentence grouping, topic sentence callout, and RST function identification, to

[2]Demo available at `https://jake-cui.github.io/thesis-demo/`

[3]Code available at `https://github.com/jake-cui/thesis-demo`

format natural text in a way that is easier to digest. These features aim to circumvent the obstacles described above by reducing the required cognitive workload for readers and minimizing the amount of irrelevant text that is consumed. I will describe each of the three features in greater detail below.

3.3.1 Sentence Grouping

Writing already indicates grouping of text through paragraph structure. When readers move from paragraph to paragraph, they typically will anticipate a general switch in topic matter. *READR* takes this one step further by automatically grouping together sentences that are relevant to one another, even if they are in the same paragraph. It is a more granular way to indicate to users what clauses are related to one another. When users hover over text with a mouse, the sentence and other related clauses will be highlighted.

The intended use case is for users who are skimming over sections and quickly want to find information relevant to certain keywords. Additionally, I hypothesized that it would be helpful just to have more indicators of when text is shifting over from one topic to another.

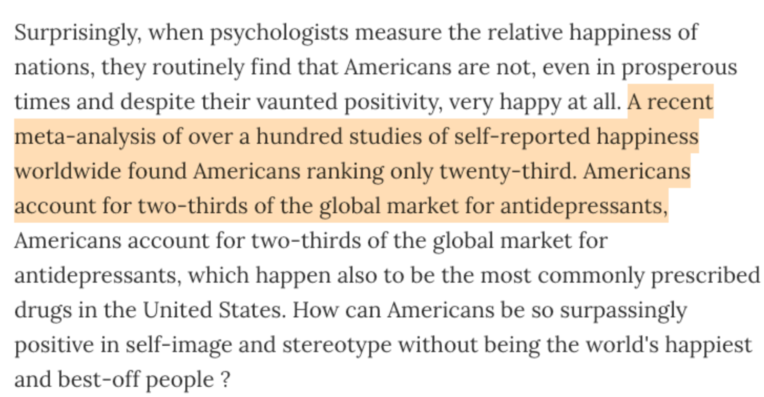

Figure 3.3.1: Example group highlighting when a user hovers over a sentence.

The groupings were very easy to extract from a RST tree structure. Currently, my algorithm is looking for a cluster of *n* nodes that have the same root, in which *n* is a range of size values that can be specified by the algorithm. For initial testing, I selected an arbitrary *n* range of 3 - 7 (in other words, clauses are grouped in sizes of between 3 - 7). This value can be tested and modified based on user feedback.

3.3.2 Topic Sentence Callout

In skim reading, there is a heavy emphasis on the identification of topic sentences. Users tend to look through paragraphs until they find a topic sentence that is relevant to their information of interest [2]. *READR* makes this easier for users by automatically identifying key topic sentences and highlighting them explicitly. Additionally, the platform will group all relevant following text under the topic sentence as shown in Figure 3.3.2. Users have the option to expand and collapse the auxiliary information by clicking on the header sentence.

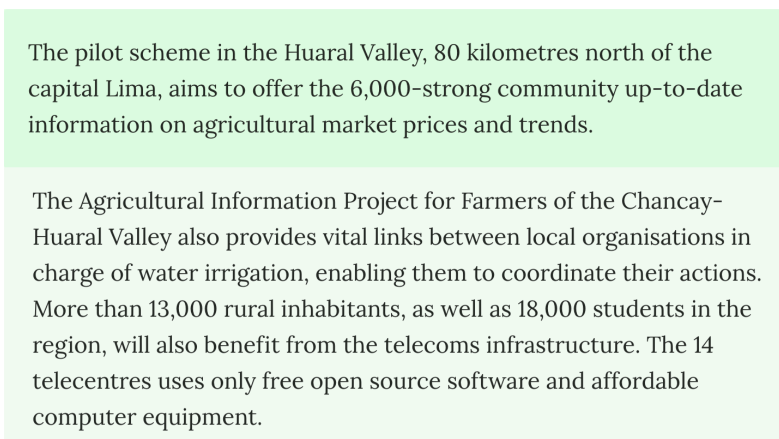

Figure 3.3.2: Example topic sentence highlight and grouping.

The motivation behind this feature is that users tend to rely heavily on topic sentences while skimming through text to find important information. While topic sentences frequently occur at the beginning of paragraphs, this is not always the case. By highlighting these key sentences explicitly, I aim to reduce the cognitive workload for readers. Users will not have to scan through paragraphs to find key sentences.

Algorithmically, it is fairly straightforward to identify topic sentences based on RST structure. In this thesis, I take a fairly naive approach as follows:

1. Identify all leaves (which are complete clauses in this case) that are nuclei.

2. Of the leaves identified, find each one that has a body of text consisting of greater than *x* discourse units branching off the original leaf.

In short, I am identifying all clauses that serve as the root of several other clauses. The *x* value in the algorithm can be specified by the user as well. In this thesis, I chose an arbitrary minimum of 0, but this can be iterated on with further experimentation.

3.3.3 RST Function

The final feature included was the identification of RST function. CODRA is able to identify every rhetorical relation between clauses and groups of clauses. For both the sentence groupings and topic sentence callouts, users will be able to see the rhetorical relation of each by hovering over the text with a mouse.

Additionally, note that the topic sentence callouts have different colors. They are color coded based on RST function as shown in Figure 3.3.3.

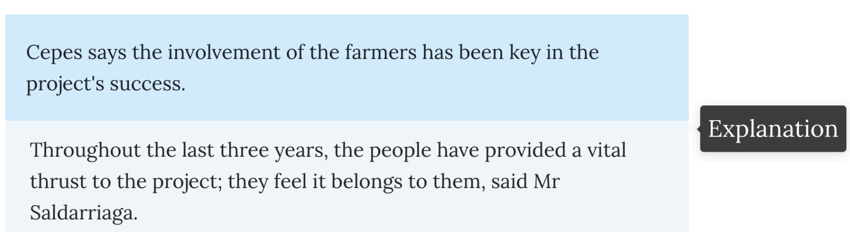

Figure 3.3.3: Example RST function being shown when a user hovers over text.

The use case for this feature is the least immediately obvious. In general, I hypothesized that by explicitly outlining the RST function of certain sentences, it would allow users to form a better mental understanding of the text structure. I planned on heavily observing and interviewing participants during user study to see how they were able to take advantage of these function classifications.

3.4 User Study

To evaluate *READR*, I designed a simple user study with a few goals in mind. To begin, this was intended to be more of a preliminary and exploratory study in which I could receive more qualitative feedback on the interface. I also wanted to evaluate the usability of my interface, focusing on how the platform was designed. Finally, I was interested in seeing if *READR* actually helped users read more effectively. The primary metrics I wanted to evaluate were reading comprehension and reading time. The experimental task was to complete a series of reading comprehension questions for three articles, similar to those on a standardized test. I conducted the user study described below over the span of two days.

3.4.1 Participants

Five Harvard undergraduate students (3 male, 2 female) were randomly recruited for this study. Each was a native English speaker with college level reading proficiency. There were three users in the control group, and two in the experimental group.

3.4.2 Experimental Design

Users were randomly split into a control and experimental group. The control group was asked to answer a series of reading comprehension questions based on the raw text of an article, similar to the way they would read online. The experimental group was asked to answer the same set of questions when the article was parsed with *READR*. All typography and text was exactly the same across the two groups to ensure consistency[4].

3.4.3 Task and Materials

Three different articles were selected for the user study that were meant to represent a range of reading difficulty. The first (Article A) was a BBC news article that was meant to be easiest to read. The second (Article B) was from a popular book that was intended to be more difficult. The final passage (Article C) was a section pulled from a philosophy paper which was intended to be the most difficult. Each paper had six, three, and five reading comprehension questions respectively that went along with them. The difficulty of each question was roughly the same. Questions were categorized into direct and indirect, in which direct questions asked for factual information straight from the passages, while indirect ones required readers to analyze and synthesize information they gathered.

3.4.4 Procedure

Users were given five, five, and ten minutes to read Articles A, B, and C respectively. These times were meant to make the readers slightly rushed in order to encourage faster comprehension and skim reading practices. Additionally, some questions were given to users while they were reading, and others were asked as follow ups after they finished reading. This was meant to identify if there was a significant difference in the look up and recall performance of participants using *READR*. After each article, users were asked for qualitative feedback regarding both the article itself and the *READR* user interface. For the experimental group, users were given a quick tutorial regarding the basic functionality of the interface (under two minutes) but there was no interaction during the actual reading portion of the study. Finally, to measure usability, users

[4] An example of a raw text article is available on the demo link.

were asked to rate *READR* on the System Usability Scale[5] to evaluate general usability. Each user study took around 45 minutes.

[5] https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

4
Conclusion

4.1 Results

Due to the small size of the user study, it was difficult to perform rigorous statistical analysis on the data collected. Regardless, there are few quantitative results that we can use to get a general understanding of feedback on the system.

4.1.1 Understandability

To start, users were asked to evaluate the readability of each of the three articles on a 1-5 scale (where 1 is most difficult to read, and 5 is easiest). The results are shown in Table 4.1.1.

While we cannot make any statistically significant conclusions, it seems like that users found articles slightly easier to read when using *READR*. On average,

	Article A	Article B	Article C	Average
Control	3.33	4.0	1.5	2.9
Experimental	4.0	3.75	4.0	3.9

Table 4.1.1: Understandability ratings for the articles across experimental groups (on a scale of 1-5).

participants gave the articles a readability score of 2.94 without *READR* and 3.91 when using the platform. This effect was most notable in Article C, which was the most difficult and dense paper to read, in which users gave a readability score of 1.0 without *READR* and 4.0 with *READR*. I hypothesize that this is because the identification of topic sentences and collapsing of irrelevant text.

4.1.2 Reading Comprehension Accuracy

Additionally, we also have results from the participants' performance on the reading comprehension questions; results are shown in Table 4.1.2. Again, there appears to be a slight increase in reading comprehension performance, but this is hard to conclude based on small participant group size. It is also important to note that there is likely a large amount of variance in a user's ability at correctly answering the reading comprehension questions.

	Article A	Article B	Article C	Average
Control	83%	50%	67%	67%
Experimental	92%	67%	70%	76%

Table 4.1.2: Reading comprehension accuracy.

As shown above (Figure 4.1.2) there was a 9% increase in question accuracy for participants using *READR* (results are not significant). We also collected data on users question accuracy when asked questions with the article in front of them and afterwards. This was intended to distinguish between simply looking up answers in a text and recalling meaningful information after reading. The results are shown in Table 4.1.3. It appears that *READR* assisted users in recalling information rather than just searching for text relevant to a certain question.

	Lookup	Recall
Control	75%	58%
Experimental	75%	90%

Table 4.1.3: Reading comprehension accuracy for questions with and without the article.

The last set of quantitative data collected was to identify system usability. I employed the System Usability Scale (SUS) for users in the experimental group. Results are shown in Table 4.1.4.

Question	Rating (1-5)
I think that I would like to use this system frequently	2.5
I found the system unnecessarily complex	2.5
I thought the system was easy to use	4.0
I think I would need technical support to use the system	1.0
I found the functions in this system well integrated	4.5
I thought there was inconsistency in this system	2.5
Most people would learn this system quickly	4.5
I found the system very cumbersome to use	2.0
I felt very confident using the system	4.0
I needed to learn a lot before getting going with the system	2.0

Table 4.1.4: System Usability Survey results in the experimental group.

The ratings above correspond to a 1-5 scale in which 1 and 5 represent strongly disagree and strongly agree respectively. Due to study size, data was gathered from only two participants. Some notable points however, include 'I thought the system was easy to use' with a 4.0 score, and 'Most people would learn this system quickly' with a 4.5 score.

4.1.3 User Feedback

A bulk of the data collection in this user study was focused on user feedback. A majority of the study was devoted to observing and interviewing participants to get a better understanding of how *READR* could be most effectively used.

Usability

Users had very positive feedback for the actual interface. The reading layout was inspired by Medium's typography[1] which users felt to be clean and intuitive. Users did not struggle to navigate the interface and immediately began taking advantage of the highlighting and collapsibles. There were no criticisms in terms of the interface design and user interface. Moving forward, I believe that the general interface design can be maintained. No significant changes appear to be necessary.

Sentence Grouping

One participant really appreciated the sentence groupings, claiming that it helped them mentally group sentences and ideas together. The other felt that the highlighting was a bit distracting, especially when a single sentence was split up into separate groups. Moving forward, we may consider some hard coded conditions that will prevent situations like these.

Topic Sentence Callout

Users really appreciated the topic sentence callouts. Especially with the look up oriented questions, users liked how the highlighting immediately identified sentences of importance. Users were able to effectively use the topic headers to more quickly skim over sections that were not relevant to the information they were interested in. Participants also remarked that they appreciated how the

[1] https://medium.com/

expanded content was guaranteed to pertain to the topic header. In general, I would say that this feature was the one that participants felt was the most useful.

While the general topic sentence callouts were well received, there was more mixed feedback regarding a few other design decisions. For one, no participants felt that the color coding based on function was particularly useful. Additionally, there was a bit of confusion regarding the relation classification. Participants were a bit confused how the label applied to the header and body text. One participant felt that the highlighted callout was a bit distracting and disruptive to the natural flow of reading an article. Finally, another user had missed information that was in the body of a collapsible because they felt that the text would not be important due to the content in the header.

RST Function Identification

Users felt that the RST function identification was relatively unhelpful for the task provided. They did not feel like having the relations outlined promoted understanding, with exception of the occasional topic callout. Moving forward, it is important to think about whether or not these function callouts are actually necessary. Regardless, we will have to think of potential adjustments to take advantage of the RST classifications.

Learning Curve

Although users claimed that the interface was easy to use and understand, it took the participants quite a while to read through with *READR* smoothly. The interactive platform still introduces a bit of a learning curve, as it requires users to go through text in a different way than they are used to. All participants felt that they were able to leverage the tool better as the study went on, as they became more comfortable interacting with the system. This shows that there definitely is an initial learning curve, but it does not seem too difficult to overcome. Even after reading one or two articles on the platform, users felt much more confident navigating the tool.

4.2 Discussion and Future Work

All in all, users felt that the platform was interesting and unanimously agreed that it could be useful for certain use cases. Users seemed to feel that the interface would be best for use cases in which readers were searching for specific information in an article. They appreciated the topic sentence callouts especially, given that they were under a time crunch during the experimental task. All users took up a majority of their allotted time, indicating that they could have performed better if given more time. Users also felt that the tool was excellent for helping readers get a quick, general understanding of a text. In theory, by skimming through the topic headers and corresponding sentences, a reader can grasp the general gist of the paper. *READR* seems to have a lot of potential for reading difficult and dense text in which users cannot skim as easily. It is in these use cases in which we may be able to take the most mental workload off users' shoulders.

Based off feedback from the user study, there are a set of next steps and adjustments that I would like to propose.

1. Additional user studies are necessary. In this thesis, I was able to gather a good amount of qualitative feedback from five participants, but ideally I would like more quantitative results from the comprehension tasks.

2. Slight modifications need to be made to the user study. As shown in the results of this experiment, there is a slight learning curve for users using *READR* for the first time. This is bound to skew the comprehension results if not somehow addressed.

3. The main critique from users was that the *READR* interface felt a bit distracting at times. It also appears that all the features included may not be absolutely necessary. In the future, I may consider removing either the highlighting or the RST function identification, as these features did not appear to have a significant impact on reading comprehension.

4. I would like to continue experimenting with how *READR* is used with different kinds of text. The results of this study show that the nature of the text greatly affects how users will benefit from the tool. In the future, I would like to further investigate these differences with a wider variety of text documents.

4.3 Conclusion

READR and the results of this thesis indicate that RST has great potential in the preprocessing of text to improve readability. The three features implemented in this proof-of-concept helped users get a better understanding of raw text while under time pressure. Furthermore, users felt that they had a better overall reading experience with the tool, claiming that the same articles were easier to read with *READR*. The tool seemed to excel at helping users find important areas in text related to key topics of interest. Although there are several areas of improvement and future expansion, the results of this preliminary study are promising. It is very straightforward to convert RST data into a readable and dynamic text body. I am excited to see how this technology may revolutionize the way interact with our digital devices.

References

[1] Fernando Loizides and George Buchanan. An empirical study of user navigation during document triage. pages 138–149, 09 2009.

[2] Geoffrey Duggan and Stephen Payne. Skim reading by satisficing: Evidence from eye tracking. pages 1141–1150, 05 2011.

[3] Rolf Zwaan, Joe Magliano, and Arthur Graesser. Dimensions of situation model construction in narrative comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21:386–397, 03 1995.

[4] William Mann and Sandra Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8:243–281, 01 1988.

[5] Mathieu Morey, Philippe Muller, and Nicholas Asher. How much progress have we made on RST discourse parsing? a replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1319–1324, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[6] Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. CODRA: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435, September 2015.

[7] John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285, 1988.

[8] Stefania Passera. Beyond the wall of text: How information design can make contracts user-friendly. 2015.

[9] Paul Chandler and John Sweller. Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8(4):293–332, 1991.

[10] Detlev Leutner, Claudia Fohry, and Elke Sumfleth. Cognitive load and science text comprehension: Effects of drawing and mentally imagining text content. *Comput. Hum. Behav.*, 25:284–289, 2009.

[11] M.A.K. Halliday and R. Hasan. *Cohesion in English*. English Language Series. Taylor & Francis, 2014.

[12] Richard Montague. *The Proper Treatment of Quantification in Ordinary English*, pages 221–242. Springer Netherlands, Dordrecht, 1973.

[13] Hans Kamp. *A Theory of Truth and Semantic Representation*, pages 329 – 369. Brill, Leiden, The Netherlands, 2013.

[14] J.R. Martin. *English Text: System and Structure*. John Benjamins Publishing Company, 1992.

[15] Alistair Knott. Using linguistic phenomena to motivate a set of coherence relations. *Discourse Processes - DISCOURSE PROCESS*, 18, 04 1993.

[16] Alex Lascarides and Nicholas Asher. Segmented discourse representation theory: Dynamic semantics with discourse structure. In H. Bunt and R. Muskens, editors, *Computing Meaning: Volume 3*, pages 87–124. Kluwer Academic Publishers, 2007.

[17] Laurence Danlos. Comparing rst and sdrt discourse structures through dependency graphs. 01 2005.

[18] Kamel-Eddine Haouam, Touir Ameur, and Farhi Marir. Towards a framework design of a retrieval document system based on rhetorical structure theory and cue phrases. pages 139–148, 01 2003.

[19] Eva Forsbom. Rhetorical structure theory in natural language generation. 2005.

[20] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. *Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory*, pages 85–112. Springer Netherlands, Dordrecht, 2003.

[21] Luz Rello, Martin Pielot, and Mari-Carmen Marcos. Make it big! the effect of font size and line spacing on online readability. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 3637–3648, New York, NY, USA, 2016. Association for Computing Machinery.

[22] Jayeeta Banerjee, Deepti Majumdar, and Dhurjati Majumdar. Readability, subjective preference and mental workload studies on young indian adults for selection of optimum font type and size during onscreen reading. *Al Ameen Journal of Medical Sciences*, 4, 01 2011.

[23] Byungjoo Lee, Olli Savisaari, and Antti Oulasvirta. Spotlights: Attention-optimised highlights for skim reading. 05 2016.