**LASI: A Written Summary**

**Table of Contents**

1. Introduction **-- Scott**
2. Product Description **-- Richard**
   1. Key Product Features and Capabilities **-- Brittany**
   2. Major Functional Components (Hardware and Software) **-- Aluan**
3. Identification of Case Study **-- Dustin**
4. Prototype Description **-- Dustin**
   1. Major Functional Components (Hardware and Software) - - **Richard**
   2. Features and Capabilities -- **Erik**

Appendix A. LASI User Stories and GUI Flow

**1. Introduction**

**LASI – Linguistic Analysis for Subject Identification**

For this semester and next the CS410 Red Group is working to develop a program we have titled LASI. LASI stands for Linguistic Analysis for Subject Identification. Linguistic Analysis, in this case, means the contextual study of written works and how the words combine to form and overall meaning. LASI will be a decision support tool that will take in multiple documents of varying file types and return a weighted list of themes common to all. Themes are important because they help the reader to comprehend what has just been read. Then, if they can comprehend what was read, they can also summarize the material. Comprehension and summarization are important because together they help the reader to communicate the content of the material with other people.

Our general societal problem is that it is difficult for people to identify a common theme over a large set of documents in a timely, consistent and objective manner. This societal problem comes from Dr. Patrick Hester who, as part of his position at NCSOSE (National Center for System of Systems Engineering), analyzes large sets of documents to identify common themes and ideas. This document analysis process, as it stands, is lengthy, requires validation from his associate Dr. Meyers and also requires multiple readings of the documents. LASI would assist him in this document analysis phase by cutting down on the time spent reading through the documents. LASI will provide him with a weighted list of potential themes from which he can choose the one he feels best fits his understanding of the material. For LASI to effectively resolve our societal problem, and more specifically assist Dr. Hester and Dr. Meyers, it will need to accurately find themes, be system efficient and provide consistent results.

**2. Product Description**

This product is designed as a self-contained, stand-alone application to be run on a local machine such as a desktop or laptop. It will assist users in determining themes, and general meaning across multiple documents. LASI will provide consistent, timely results using its engine of algorithms.

LASI will also have a friendly and intuitive user interface. Users should be able to use and navigate the program without much formal training. After an analysis of a set of documents, the resulting data should be immediately available in a friendly, usable format. This will allow any user to not require a degree in computer science and English to use the product.

**2.1 Key Product Features and Capabilities**

LASI will be able to take in multiple documents and figure out the common theme between all of the documents added to the project. A new document can be added to the project at almost any time, except when the project is compiling. This allows the user to tweak the documents that are in the project when needed. There are also a few advanced features that LASI will include that make it stand out from the crowd. Before running the project you can decide on the type of analysis that you would like LASI to perform whether it is scientific or literary.

LASI will have the ability for the user to input words as a dictionary. If there are any company specific jargon that would make LASI's ability to parse documents easier the user will be able to build a dictionary for that project. As another advanced feature, the user will be able to start the project off with assumptions about the content. If the user knows that certain words or phrases are equivalent, then those can be specified before the project is compiled. This would help lower the time that LASI takes to go through all of the documents.

One of the most important features of LASI is the ability to view the outputted results in an easy to view format. The results will be separated into three sections easily navigated by tabs. The first tab will be the Top Results. This will provide a graphical representation of the most likely themes for the documents analyzed. As shown in figure 1.1-A below the top results will be pictured in a tornado chart listing the top 10 themes from most likely to least likely with their associated weights.

Figure 1.1-A

The user also has the option to change the type of chart the information will be shown in.

The second type of results shown will be the word relationships in each individual document as shown in Figure 1.1-B. The word relationships results primarily deals with parts of speech. The individual words will be colorized based on their parts of speech as well as having a link between individual words to show the links between them.

Figure 1.1-B

The last type of results will show the word counting and weighting of each of the words. As shown in Figure 1.1-C below this will just be a simple table with the ability to change the order in which the results are displayed. The word count and weighting displayed in each individual document. This allows the user to look at each document separately to see they affected the overall results. Word counting and weighting will also be able to show the words that were common throughout the documents analyzed and the individual weights that were placed on those words.

Figure 1.1-C

Another important feature that LASI will provide is the ability to print and export the results. The results will be exportable in a multitude of formats including but not limited to: pdf, jpg, and ppt.

**2.2 Major Functional Components (Hardware and Software)**

***Hardware Platform***

LASI has been designed to run as a standalone, text analysis engine. As such, it will be able to run on business grade laptop and desktop computers as well as high grade consumer PCs. Roughly speaking, a quad-core processor, 8 gigabytes of low latency DDR3 SRDAM, and a solid state based storage medium comprise the hardware requirements.

***Linguistic Analysis Algorithm***

***Primary Phase***

During the initial phase of analysis, the primary focus will be to build a textually agnostic set of associations based on the determined part of speech for each token in a document. At this relatively low level of abstraction, the document will be represented by a text-wise-linear collection of role specialized word objects where each corresponds to a text token and an inferred part of speech. Through this, a frequency assessment of the words will be used to determine an initial, naïve significance metric, for each usage of each word.

***Secondary Phase***

After the initial assessment, this naïve metric will be refined through two syntactic association procedures. Firstly, binding instances of pronouns with to the nouns they refer to will significantly increase the accuracy of the noun significances determined during the initial phase. Further, by binding adjectives to the nouns they describe, we begin to associate adjective counts to specific noun instances. Both of these procedures are significant in that they begin the process of associating together the linear text into constructs, thus they begin raising the level of abstraction from individual words to inter-word relationships.

***Tertiary Phase***

From here, we will begin to bind subject and object entities to each other by way of the verbs which associate, through this, we can identify and correlate the ways entities are related to one another and the significance of these associations. These linkages form the basis for statistical synonym association and, most significantly, form the basis for the themes the LASI analysis engine will identify for the user.

***Progression and Refinement***

As more and more constructs are linked together the relationships between them begin to complexify into abstract semantic associations allowing for continuing refinement of the results via multi-pass, iterative refinements and branching assessment of parallel possible implications. This forms a basis for future higher order algorithm phases to be develop as abstractions on top of the analysis provided by the three outlined here.

**3. Identification of Case Study**

**Dr. Patrick Hester & Dr. Tom Meyers Work for an organization called NCSOSE. At NCSOSE, they currently utilize a process known as the AID process to identify problem statements from groups of strategic documents. AID stands for Assessment Improvement Design. The Assessment phase of this process is currently what LASI will be improving upon. This phase of the process involves analyzing multiple strategic documents in a range of domains he may not be an expert in and then based on several criteria and the identification of key components. This involves him doing extensive research and is an incredibly time consuming and in-depth process.**

**Dr. Hester and Dr. Meyers then take the analyzed data from these documents and formulate a concise and accurate problem statement. This problem statement is used to identify organizational issues and then to offer ways to optimize the company who consults with Dr. Hester and Dr. Meyers and improve efficiency. This outlines the current AID process.**

**There is currently a bottleneck in the system at the Assessment part of the process which we believe we would be able help resolve. LASI by nature will eliminate much of the thorough analysis that Dr. Hester does of the documentation proved and output the same results to him quickly. All he will need to do is analyze the results of LASI and determine the most appropriate theme from the results. It will remove much of the guesswork because of the objective nature of LASI and it will be a great means of defending his findings. LASI will provide him with a group of themes in an objective, consistent, and timely manner. This will resolve the bottleneck in his system and allow him to interact with his clients while he is processing the “Assessment” step in AID.**

**4. Prototype Description**

Due to the complex nature of our algorithm and the simplistic nature of both our user interface, as well as the hierarchy of users, we have decided that the only area that needs to be scaled down is our algorithm. The algorithm’s output will be approximately the same. It will still accept multiple documents and search for common themes across them. It will still interact with the UI the same way. The results will still be displayed graphically on the results page. The differences will be that the algorithm will be a bit less versatile as it will make a few more assumptions about the documents through which it searches. The only two acceptable input types will be .doc and .docx files. It will also be forced to limit the number of documents that it can analyze. The prototype will limit interpretable documents to 10 pages or less. Lastly rather than focusing on mapping every part of speech to related parts of speech, we will focusing on subjects, verbs, direct objects, and indirect objects.

The decision to limit ourselves to strategic documents is due to the fact that it is all our case study required. It will also make our results more accurate for our case study to be specific with the nature of the documents we parse. For instance, we can search for keywords like “Mission,” “Vision” and “Goals” and then place increased emphasis on the content below those words to create a more accurate weighting algorithm. This also saves us some work because it leads to us having to make fewer passes of the document.

The decision to limit the input types to either .doc files or .docx files is due to the fact that determining a suitable mechanism for parsing other data types requiring optical character recognition would require us to learn an entire API and would increase the likelihood of errors in parsing. We would need to make sure that the program doesn’t read the same word or phrase differently based on input file extension. A homogenous format for gathering parsing input would resolve that. Also, .docx files are incredibly simple to parse and doc files can be converted easily to .docx.

Limiting the number of documents in the prototype will result in a simplified output. If we allow for too many documents to be parsed, we will need to develop some sort of algorithm to remove irrelevant subject/object/verb associations. We won’t have time to do this, so we need to keep our input pool small because otherwise, the results will be confusing and difficult to read. Another reason to limit the number of input documents is that this prototype will be a multi-threaded application. The more documents that are allowed for input, the more memory will need to be devoted to this tool.

Limiting the number of pages in a document accomplishes essentially the same thing as limiting the number of documents. The program will need to rely heavily on its weighting algorithm to determine themes of these documents. Not assuming a fixed length increases the amount of testing we need to do exponentially and we simply won’t have the necessary time to debug the prototype in 18 weeks without limiting input.

Lastly, decision for limiting our prototype to identifying subjects, verbs, direct objects, and indirect objects was because at its core, themes can still be derived from these, but it produces fewer incorrect associations than going deeper and analyzing every single part of speech. If we focus on subject/object relationships, we can determine valid themes with an increased statistical likelihood of accuracy, but we won’t need to go back through and remove false associations. This change will also remove the need to search for synonyms individually as they can be determined not just by verb associations, but by the associations with the other parts of that sentence.

**4.1 Major Functional Components (Hardware and Software)**

The major functional components for the prototype will remain largely unchanged for the prototype. The hardware component of LASI does not change for the prototype. The LASI prototype will still be designed to run on a higher end laptop as described in the original major functional component diagram as shown above.

All of the major software functional components will still exist. Although, each component will be scoped down for the allotted amount of time to design and create the prototype. LASI will still be based largely on its algorithm. The complexity has been reduced from running through the 3 phases that were originally determined to run. The primary phase will still be performed during analysis, along with parts of the secondary and tertiary phases, including outputting a weighted data set of themes.

The user interface will still keep its multiple views to represent the resulting data. It will also display a user friendly list of weighted themes. The step-by-step justification and detailed breakdown displayed by the user interface will provide some additional information, and not as in-depth as originally planned. This is largely based on the parts of LASI’s algorithm that are completed. It is planned to develop intuitive ways of presenting this kind of information (i.e. pop-up bubbles, tornado diagram).

**4.2. Features and Capabilities**

Time constraints, on the following semester, have required for LASI’s list of features to be decreased in size. Our team has looked over all of LASI’s possible capabilities and removed those seemingly unnecessary for our first implementation, in order to have a working, real-world product by the end of Spring. We have decided to limit the types, lengths, and number of input documents. In addition to these limits, we have agreed upon implementing Verb Binding solely, as opposed to Pronoun *AND* Verb Binding, as well as eliminating Synonym Identification and reducing the Number of Passes required to parse each collection of text. Some user-input elements have been removed and consist of the removal of user Word Dictionaries, Keywords, Content Assumptions, and Specified Document Types. The final reduction of LASI’s full prototype to LASI’s real-world prototype is the removal of the visual representation of LASI’s Reasoning Process.

|  |  |  |
| --- | --- | --- |
|  | **“Full” Prototype** | **"Real World" Prototype** |
| **Input Document Types** | All Unstructured Text | .DOC/.DOCX Only |
| **Number of Input Documents** | Arbitrary | 3 to 5 |
| **Input Document Length** | Arbitrary | 10 pages |
| **Addition of Input Documents** | Anytime | Prior to Parsing Only |
| **Parts of Speech Binding** | Pronoun and Verb Binding | Verb Binding Only |
| **Number of Parses/Passes** | Number Required For Accuracy | Fewer Than Optimal |
| **Word Dictionaries** | User-input + LASI's | LASI's Only |
| **Synonym Identification** | Enabled | Disabled |
| **User-input Keywords/Content Assumption** | Enabled | Disabled |
| **User Specified Types** | Enabled | Disabled |
| **Represents Reasoning Process** | Enabled | Disabled |

**Appendix A. LASI User Stories and GUI Flow**