CLASH Product Description CS410 Lab 1

Blue/Purple Teams

Old Dominion University

Table of Contents

1 Intro (Cory)

CLASH, or Color Lexical Analysis algorithm and Slash Handler is a web interface that includes two modules, 'COLRS' and 'Slash'. The 'COLRS' module 'colorizes' each part of speech (P.O.S) in a text document with a particular color to help increase comprehension of sentence structure and grammar. The 'Slash' module parses text into lexical bundles (a group of words that occur repeatedly together, or present one single thought), or thought groups to help increase reading speed and comprehension. A lexical bundle is a group of words that occur repeatedly together within the same register. Though our program could be useful in a number of settings, we are primarily concerned with English as a second language (ESL) students.

Based on state-reported data, in 2004, it is estimated that 4,999,481 ESL students were enrolled in public schools. Despite those numbers 15% of these ESL students had no special resources or programs to help them learn the language. This problem is systemic. In 2001, in the states that tested ESL students in reading comprehension, only 18.7 percent of ESLs were assessed as being at or above the norm. That same year, 10 percent of ESLs in grades 7-12 were retained. In February 2001, it was reported that ESLs had dropout rates up to four times that of their native English-speaking peers. Here at ODU, we have an entire department focused on addressing this problem, but while they work diligently to help ESL students, the processes of teaching reading and grammar are outdated. (McKeon)

Currently, the process is simple, for grammar, the professor writes a sentence on the board, then circles, or marks in some way each part of speech. Not only is this time consuming, it limits the amount and size of examples that can be given. Psychologically, it has been proven that color impacts learning, they relieve eye fatigue, increases information retention, increase productivity and accuracy, and support developmental processes. (HTTP://sdpl.coe.uga.edu/HTML/W305.pdf) Extrapolating from that information the 'COLRS' module should help students identify P.O.S. and increase their potential for learning grammar. For increasing reading speed and comprehension, reading assignments are given, or students are directed to sites like Spreeder(HTTP://www.**spreeder**.com). These sites don't help comprehension due to their focus on speed; they teach students to read word for word vs. in lexical bundles. It has been shown that those who learn lexical bundles read faster and perform better in word and sentence recall experiments.(Tremblay, Derwing, Libben, Westbury)

2 Product Description (Andrew, Justin)

CLASH is a solution that aims to help English as a Second Language student learn English easier and faster than traditional means. CLASH will be a web-based solution that will improve reading speed and comprehension. It will possess a Graphic User Interface for interaction with the student and the database. This tool will also allow instructors to monitor the usage of application by each student. The CLASH program consists of two modules Slash and COLRS.

Slash will improve reading speeds of ESL students. Slash will break up documents into lexical bundles for students to digest a passage easier. Slash will also greatly improve reading comprehension and will turn reader from word-to-word readers into lexical bundle readers. Slash will allow uploads of text documents for leisure reading. Slash possesses an easy to understand user interface for ESL students to utilize.

The COLRS module assist students in identifying parts of speech through the colorizing of text. The 8 traditional parts of speech, less interjection, that students learn will be the focus of the product with future updates that could add more nuanced parts of speech. Colorization of text will be done through JavaScript manipulation of HTML text from documents previously parsed and a text stream that was derived from said document.

2.1 Key Product Features and Capabilities (Fred) (Charles)

CLASH will be a web-based application that will allow users to improve their reading comprehension by breaking up sentences into lexical bundles and showing the parts of speech. CLASH features can be accessed using a standard web browser and internet connection. Through the application's user interface, users can control the speed at which lexical bundles are displayed. Parts of speech can be viewed in a separate part of the user interface.

The product features individual password-controlled user logins with three different types of user roles. These roles include Administrator, Instructor, and Student. The Student will be able to control their reading speed, type of view, and which available document to view. The Instructor will include the Student user capabilities plus more. The Instructor will be able to add/remove Students as users and select the documents available to be viewed by the Students. The application allows for the Instructor to view activity data for the Student users. This activity data will include the student's current reading speed, and the amount of time spent on the site. The

Instructor will be able to upload documents to the server to be parsed, edit files, and delete documents currently on the server. The Administrator has all the capabilities of the Instructor plus the ability to add/remove Instructors as users.

One of the features that make CLASH unique is that it both displays parts of speech and lexical bundles. This allows students to improve their reading comprehension by having a speed reading application that does not break up the individual lexical bundle. There is the ability to pause and change the display speed. CLASH will include a text parser that is able to automatically identify parts of speech, and color them accordingly, allowing for easy identification. It will also allow instructor review of the parsed text, to ensure accuracy(because no program is perfect). CLASH will be able to save usage data, allowing instructors to review student progress, and fine tune their lectures accordingly. CLASH is the first web-based speed reader specifically designed for use in ESL instruction. Instructors can have documents available to the students at their appropriate level. By focusing on ESL students, CLASH will give students and instructors a powerful tool to increase reading speed and comprehension.

2.2 Major Hardware and Software Components (Ming)

Overview. The CLASH will be a web-based application that hosted on the server, and can be easily access through web browser. There is no special hardware requirement other than an active server and a database on the server end, and an internet-enabled device for the client to access the application. There are three major software components for CLASH, COLRS Module, Lexical Bundle Module, and Client-side Reader.

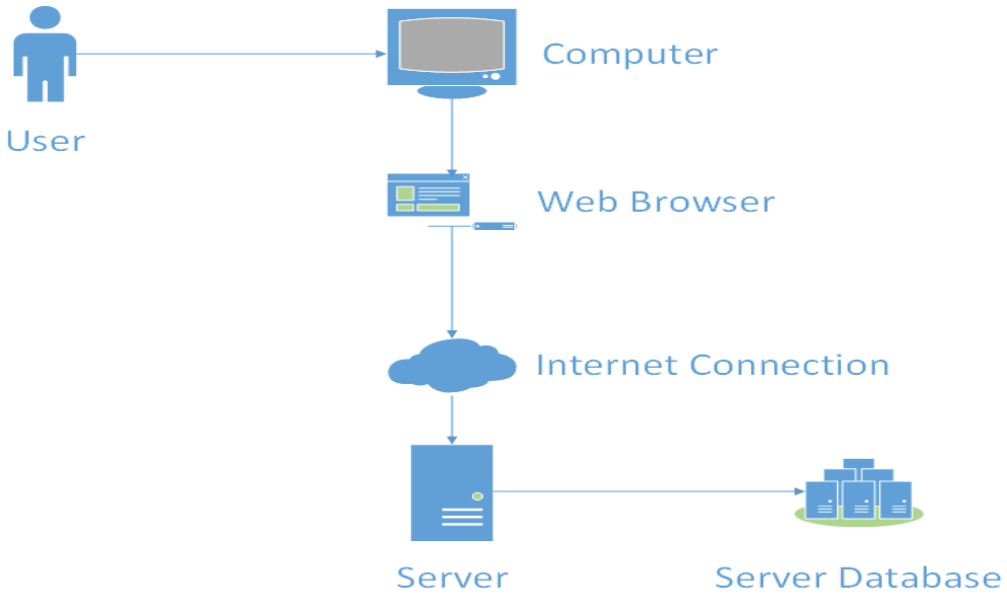
COLRS module. COLRS Module use open source Natural Language Processing (NLP) to tokenize and parse the input document, and it will generalize the P.O.S. tags in more readable format. The output will be a tokenized stream with token and tag pair.

Lexical bundle module. Lexical Bundle Module will accept tokenized stream from COLRS Module and decide how to break sentences into digestible lexical bundles. There will be a sub-component to deal with the exception list that improve the accuracy of the final output.

Client-side reader. The reader program will receive data from the server and parse the text according to their P.O.S. tag. There will be two display modes, color mode and speed mode. Under color mode, user can choose what P.O.S. tag to be colorize, and choose whether the Slash

need to be displayed. User can choose the reading speed under the speed mode, and the reader program will display each lexical bundle according to the reading speed.

Hardware Requirements



3 IDENTIFICATION OF CASE STUDY(Mohammed, Ali)

Old Dominion University is famous for its international students excessive presence. A lot of students from all over the world come to this university to pursue degrees in many programs ranging from finance to engineering, nursing, and computer science. However, for these students to start their academic studies, they will need to prove that their English is good enough for them to get along and get good grades just like their native English classmates. In order for them to prove that, they need to take plenty of tests such as the TOEFL test which is taken online and called IBT. If the students don't have enough knowledge or don't feel confident enough to take the test, then they apply for the English Language Center at ODU.

Greg Raver-Lampman; an instructor at the Old Dominion University English Language Center that teaches English to students who speak English as a second or third language. These students rely on him and other ELC professors to get as much knowledge of the English language before they start their academic classes. These professors do their best to help the

students grasp the language for use in future classes. They help them to understand grammar, write paragraphs that later become papers, read articles and books, listen to videos and learn how to listen to discussions, and other techniques to assist in the learning of the language. Often, however, students do not learn as quick as they want to and tend to be very slow readers. Because of this Professor Raver-Lampman requested help from the Computer Science Senior Design class, and a software solution to help students identify all the different parts and lexical bundles of a sentence was designed.

This web application was primarily created to help students at the ELC learn how to read faster. This application can be a superior solution to websites that help other people learn English. This website can be used by not only students, but also professionals that travel to English speaking countries frequently and want to learn how to read and listen correctly. This is a solution that can be used by anyone that simply wants to learn the English Language and read it efficiently.

4 CLASH Product PROTOTYPE DESCRIPTION (Erich)

The CLASH prototype will be built as a modified version of the Single Page Application (SPA) architecture. A SPA is a highly responsive web application that fits on a single page and does not reload as the web page changes states. Traditionally, SPAs are built completely in JavaScript and each component of the SPA architecture is built in and utilize JavaScript as the primary programming language. For example, the traditional SPA stack includes a JavaScript built user-interface, a JavaScript friendly application and web server, such as Node.js, and a NoSQL database such as MongoDB. The reason SPAs use NoSQL databases is because the data is stored in JSON format, which stands for JavaScript Object Notation, and is native to the JavaScript language. Where C.L.A.S.H will differ from this traditional stack is that instead of using a NoSQL database, we will be using a traditional relational database, however, our user-interface will still be built in JavaScript and we will be using Node.js as our web and application server.

Benefits to the user. A SPA is an ideal platform for CLASH because it offers the look and feel of a desktop application, but the accessibility of a web application. This particularly benefits users who are not computer savvy for two main reasons. First, since it is a web-application, there is no new software to install and all users will simply need to login to a web

portal. Second, since the entire web application is built on a single page, the user interface will be clean and users will not get lost in a rabbit hole of web-pages. All functionality will be accessible through menus on the main page.

Overview of prototype vs real world product. As we designed our prototype, we wanted it to be as close to the real word product as possible. However, given the time-frame provided to build the tool, there are some of the concessions we had to make:

1. Database will be a traditional relational database rather than NoSQL.
2. The COLRS module will not have a homework mode for students.
3. The COLRS module part-of-speech tagging will be rudimentary and relying on pre-built natural language processing tools. There will not be any custom machine learning to improve the tool's accuracy over time.
4. Will not integrate with any ODU enrollment systems. All user accounts will be set up manually.
5. The organizational hierarchy will only include one institution. The real world product would need to be able to scale to include multiple institutions.

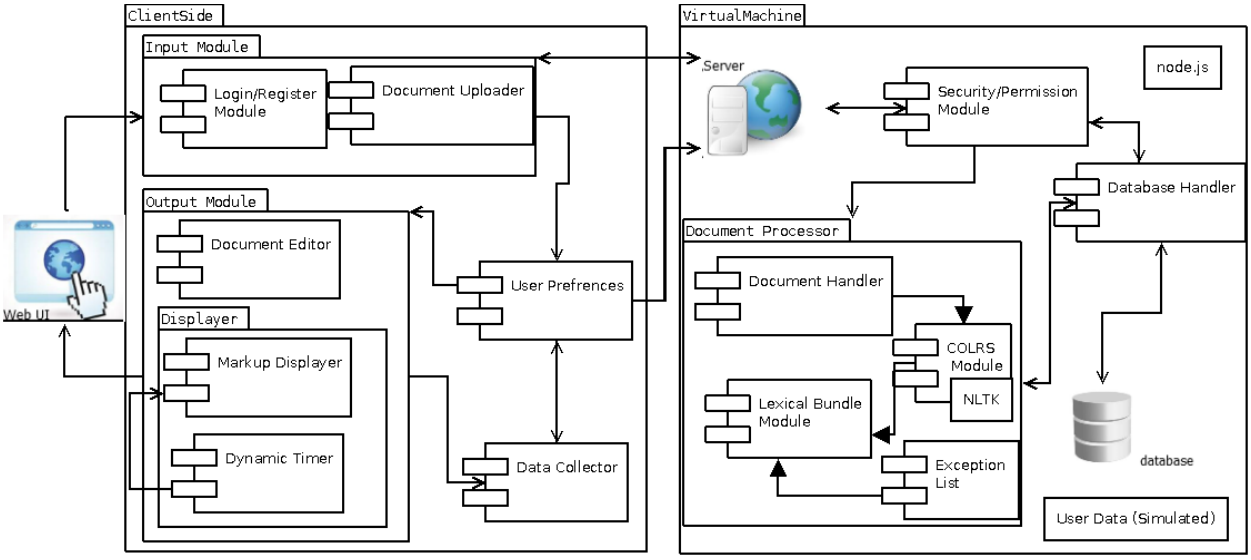
Overview of simulated areas and models. Since CLASH is not able to integrate with any ODU student enrollment systems, we will instead build in a user management module. This module will allow for instructors and administrators to add and remove users as needed. Instructors will be able to add student users on a one-by-one basis and they will also be able to add users in bulk with an ODU enrollment file. Administrators will also have access to add and remove students, but in addition they will have access to add and remove instructors as well.

4.1 Hardware and Software Prototype Architecture (James)

Hardware. The hardware required for the prototype system will initially be a laptop owned by the application developers. Upon receipt of a Virtual Machine (VM) (provided by the ODU Computer Science Department) the system will be migrated to the VM.

Software. The software required to prototype this system will be a collection of open source or liberally licensed software to permit distribution or potential commercialization of the software package. Our prototype will consist of the following packages installed under Ubuntu 14.04 LTS. The web and application server will be Node.js. This software will provide the HTTP server to allow incoming network connections from users (the students and instructors), as well as the application handling text processing of documents. The Node.js server will also interact with the MySQL database server running on the same machine, and Natural Language Toolkit. The configuration and integration of these separate components is the heart of this project. The SPA built into CLASH's Node.js server will create the web pages for the users by integrating custom programming which will parse out the input text via NLTK generate a markup stream return that stream to allow the users browser to render the text as described elsewhere in this document.

Prototype MFCD



4.2 Prototype Features and Capabilities (Justin)

The CLASH program prototype will demonstrate the ability to identify specific parts of speech by way of coloring the text and will also identify lexical bundles through the inserting of slashes. In addition the prototype will allow users to display bundles one at a time, increasing, decreasing or pausing the lexical bundle stream. Success of the prototype will be determined by

the accuracy of the tagged parts of speech and lexical bundles that will be verified by users via feedback.

Determined risks are to be mitigated through the involvement of the mentor throughout development of the product for ease of use, accuracy of core features and functions and the integration of teams building the modules.

Building a working prototype with these core features and functions will result in the customer being able to test the product in a real world situation against a control group, further assisting the development of the product. Completing these proofs of concepts the program prototype will demonstrate the feasibility of such a system and together with the prototype testing group, the utility of using the product in an academic environment thus giving the customer the desired outcomes.

4.3 Prototype Development Challenges (Francia & Fizz)

There are multiple challenges that CLASH may face on its developmental journey. One of them is computational time of the exception list. By adding more exceptions to the list, computational time increases due to the increase of size of the list itself.* Another challenge is missing attributes; meaning features are not working properly, not working at all, or missing completely. CLASH, being unique and very useful, will have minor hiccups that have to be taken care of when they show up. It is impossible to write a program without minor bugs, but those issues will be discovered by actual testing or as they occur.

English is a challenging language and with that comes errors. One particular error that will possibly arise is the incorrect identification of parts of speech and causing incorrect placement of slashes. CLASH aims to provide the user the ability to control the speed at which the lexical bundle will show with the “Slash” feature. There is a possibility of having difficulty getting the algorithm that will control the speed to do the proper task. CLASH is meant to be simple for use, meaning having a simple interface. This is to make the product accessible to users with various computer skills levels. There is that possibility that what we believe to be simple is not simple which will in the long run affect the appeal it has for certain users. Further more, another challenge that will arise with time is storage capacity. Users will populate our database

with their documents and 10GB will not be enough for long term use. Over all, our prototype is going to have a few challenges but CLASH will one of a kind.

Glossary

CLASH - Color Lexical Analysis algorithm and Slash Handler
COLRS – Colored Organized Lexical Recognition Software
ELC – English Learning Center

ESL – English as second language
IBT – International benchmark test
JSON – JavaScript Object Notation

Lexical Bundle – a group of words that occur repeatedly together within the same register

MFCD – Major Functional Component Diagram
NLTK – a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python programming language.

Node.js – an open source, cross-platform run-time environment for server-side and networking applications.
POS – Parts of Speech
SPA – single page application, is a highly responsive web application that fits on a single page and does not reload as the web page changes states.

TOEFL – Test of English as a Foreign Language

Ubuntu – a Debian-based Linux operating system.

VM – Virtual Machine

References

McKeon, D. (n.d.). Research Talking Points on English Language Learners. Retrieved December 11, 2014.

Tremblay, A., Derwing, B., Libben, G., & Westbury, C. (2011, January 15). Processing Advantages of Lexical Bundles: Evidence From Self-Paced Reading and Sentence Recall Tasks. Retrieved December 10, 2014.

Mikowski, M., & Powell, J. Single Page Applications. Manning Publications 2014.