5-1-2014

StyleCheck: An Automated Stylistic Analysis Tool

Alexander P. Welton
Dartmouth College

StyleCheck: An Automated Stylistic Analysis Tool

By Alex Welton — Department of Computer Science, Dartmouth College

May 2014

Abstract

StyleCheck is a user-friendly tool with multiple functions designed to aid in the production of quality writing. Its features include stylistic analysis (on both document-wide and individual-sentence scales) and spelling and grammar check, as well as generating suggested replacements for all types of errors. In addition, StyleCheck includes the capability to identify the famous author (out of a limited corpus) with the style most similar to the user's. The source code for StyleCheck is available online at:

`https://github.com/alexpwelton/StyleCheck`

Dartmouth Computer Science Technical Report TR2014-754

1 Introduction

A feature that has been standard on every major word processor for decades is spell check — some type of automated identification of the user's misspelled words and the methods necessary to suggest their most relevant possible fixes. In past years, Microsoft Word and its competitors have offered a "grammar check" that seeks to identify incorrect sentence structure and propose correct alternatives with the same meaning, although this task is performed far less reliably than the simpler spell check. StyleCheck attempts to extend this concept to writing style analysis by comparing the user's writing against established stylistic rules, and generating both document-wide feedback and specific replacement suggestions for problemaic sections.

StyleCheck is an automated stylistic analysis and improvement tool that traverses a writing sample, using a number of resources to identify problems and suggest fixes for them. Among these resources are frequency analysis, part-of-speech tagging, word relation databases, and large, varied corpora. By combining multiple data sources and analysis mechanisms, StyleCheck is able to make intelligent replacement suggestions for many common spelling, grammar, and stylistic errors. Because its intention is to be a user-friendly tool for those uncomfortable on the command line, all of the StyleCheck functionality, as well as basic word processing features, is available through an intuitive GUI.

2 Literature Review

The following sources were consulted when considering various aspects of StyleCheck, from the overall scope of the project to the specific implementation of each facet. Each source's relationship to StyleCheck is briefly outlined below, and the sources are divided into four sections: theory and rules for writing style, natural language processing techniques, corpora and queryable assets, and existing automated style programs.

2.a Stylistic Theory and Guidelines

The Elements of Style Strunck and White's *The Elements of Style* is perhaps the most famous book on writing style ever published, and has influenced stylistic trends for decades. The book includes a list of commonly misused words, various rules for punctuation and grammar, and, most importantly for StyleCheck, a list of prescriptive style rules. The style rules Strunck and White present are mainly concerned with the production of clear, concise writing, a philosophy that

StyleCheck mirrors (for the most part). Examples of their suggestions include "preferring the standard to the offbeat" in terms of phrasing, or the declaration that "vigorous writing is concise." *The Elements of Style* was integral to the development of StyleCheck's stylistic rules and analysis.

The Chicago Manual of Style As the most widely used style guide in the English-speaking world, *The Chicago Manual of Style* proved an essential text to consult while developing StyleCheck's stylistic analysis components. The first edition of the guide was published over a century prior to this writing, and the contents have been updated continuously since in order to remain timely. While *The Chicago Manual of Style* includes vast sections on document formatting and citation formatting, but the prescriptive grammar rules and style suggestions are what proved useful for StyleCheck. Information from this text was used in the creation of the grammar check functionality as well as the stylistic analysis.

On Writing Well Zinsser's simply-titled style manual is one of the most influential and highly regarded style guides in bookstores today. In addition, it focuses mainly on the proper style and technique for analytical writing, which is StyleCheck's specialization as well. The sentence structure and word choice suggestions from *On Writing Well* were taken into account when crafting StyleCheck's stylometric profiling features.

The Writer's Art Due to the classic, conservative nature of the above three style manuals, *The Writer's Art* was chosen to provide a lighthearted, modern balance to StyleCheck's stylistic rules and suggestions. In addition, this text has a highly detailed deconstruction of the mechanics behind a vividly-written sentence, which proved an invaluable resource for StyleCheck.

2.b Existing Automated Style Analysis

Grammarly Grammarly is perhaps the most full-featured automated writing analysis tool outside of commercial products (like those used by standardized testing companies). It identifies simple spelling and grammar errors, punctuation errors, and checks more advanced patterns such as colloquial speech. Grammarly also features plagiarism detection and notifies the user if it locates an unoriginal section. In testing, this application correctly identified spelling and grammar mistakes, but the suggsted replacements for these mistakes were only usable in around half of cases. In addition, the algorithm gave passing marks for sentence

structure to essays written by non-native speakers that had significant problems in that area, lowering the accuracy of its assessment.

The Hemingway App The Hemingway App is a stylistic analysis tool with a specific focus on brevity and a high level of user-experience polish, but the one-dimensional analysis it provides is often incapable of understanding the nuances of style. Hemingway highlights the longest and most meandering sentences and suggests that they be split into parts, as well as highlighting what it deems to be clunky or overly long words and suggesting that they be replaced by a shorter counterpart. Similarly, the application counts occurrences of both adverbs and the passive voice, and suggests their elimination. While in the general case these ideas do improve writing style, in many cases a well-placed adverb or descriptive longer word is beneficial to the flow of a piece, and in these cases the Hemingway App actually makes writing style worse.

I Write Like I Write Like is an online tool that analyzes a writing sample using frequency analysis methods and determines which canonical writer from their list the user's style is closest to. While this is an intriguing tool, it simply provides context to the user for their choice of style rather than attempting to improve it.

3 Methodology and Results

The following sections detail the methodology used in each of StyleCheck's discrete components, and the results observed from each.

3.a Spelling and Grammar Check

3.a.i Dictionary Matching

A common task in StyleCheck, especially in spell checking, is determining whether a word is valid in American English, and this is accomplished through the use of a constant-time-queryable dictionary. The dictionary is built from the Linux Words List, a plain text list containing a comprehensive list of valid English words, which can be freely accessed at `http://www.cs.duke.edu/~ola/ap/linuxwords`. The dictionary stores all words in lowercase form, converting any input word to lowercase as well before checking it against the dictionary. In addition, StyleCheck stores a separate user dictionary for each session, which allows the user to add custom words (such as a proper noun they use frequently) to the dictionary.

3.a.ii Error Identification

To identify spelling errors, StyleCheck first traverses the document, parsing the text first into sentences and then into individual words by splitting on any non-alphanumeric delimiting characters. After removing any extraneous characters (e.g. quotation marks) from the word and converting it to lowercase, it is checked against both the user and the main English dictionary. If the word is not in the dictionary, it is considered an error.

While the approach above proved satisfactory for identifying the vast majority of spelling errors, there were several edge cases in which non-errors were mistakenly identified as errors. A common manifestation of this was the case of hyphenated words; a solution was implemented in which StyleCheck checks each word for the presence of an internal hyphen and splits the word on the hyphen or hyphens, evaluating each word individually. Another edge case is that of proper nouns; this was mostly addressed by ignoring any spelling errors that are capitalized in the middle of a sentence. Finally, typographical errors in which the user combined two words were returning poor replacement suggestions because the algorithm was evaluating the combined word as a whole, a problem alleviated by splitting each word into every possible combination of two words and checking them individually against the dictionaries.

To identify grammar errors, StyleCheck parses the document into individual sentences using the Stanford Parser's top-level tokenization function and checks the grammatical validity of each against multiple sets of rules. The Stanford Parser works by determining probabilistically the most likely grammatical structure for a sentence, from which it builds a complete sentence grammar tree and assigns the determined part of speech to each word (Socher et. al., 2013). The first is LanguageTool's grammar checking API, which proved to be a good source for grammar error identification, especially in the cases of tense and plurality disagreement (the LanguageTool documentation can be viewed at https://languagetool.org/development/api/org/languagetool/JLanguageTool.html. However, several problematic rules needed to be ignored (for example, correctly quoted text would throw an error), and LanguageTool proved a mediocre source at best for grammar replacements, often generating no replacements at all. The second is the Wikipedia corpus of commonly misused words (a list of commonly misused words and their corresponding corrections), which is queried for misused words in the sentence context (Wikipedia, 2014).

Both the spelling and the grammar errors are filtered against StyleCheck's error-ignore list before being presented to the user. The user can opt to ignore individual instances of an error, or ignore all instances of an error by electing to ignore either the spelling of the word or the grammatical rule being violated, depending on the error type. Errors that are ignored are still tracked in case they are un-ignored later, but are invisible to the user.

3.a.iii Replacement Generation

After identifying errors for both spelling and grammar, StyleCheck generates what it determines to be the most fitting replacements for the error, and displays up to three for the user to select from. Spelling error replacements are generated and ranked according to the process described in the sections below, then combined with any common misspellings obtained from the Wikipedia Corpus of Commonly Misspelled Words or the Birkbeck Misspellings Corpus (two lists of common misspellings and their corresponding correct words). Grammar suggestions are compiled from LanguageTool's suggestions, the Wikipedia corpus of commonly misused words, as well as StyleCheck's own grammar rules, and then displayed for the user, with Wikipedia-generated replacements given the most prominence.

3.a.iv Finding Closest Words

To generate replacement suggestions for the StyleCheck spell checking functionality, the program required a mechanism for locating the most similar words to a misspelled word in a sentence. If a spelling error is identified (meaning that a word is found that does not exist in either the user or the main English dictionaries), StyleCheck will first check the misspelled word against the corpora of common misspellings. If one is found, that replacement is assumed to be the best, and it is returned without further querying. If the misspelling is not in the corpora, StyleCheck iterates over all the words in both the user and main English dictionaries, calculating the Levenshtein distance between the misspelling and each word. The Levenshtein distance metric is a measure of how many theoretical edits are required to transform one word into another, where edits are defined as insertions, deletions, and substitutions. StyleCheck uses the dynamic programming Levenshtein algorithm (Wagner, Robert A. and Fischer, Michael J., 1974) in order to avoid a polynomial time complexity. After computing the edit distance for each word, it then returns the list of the closest few dozen possible replacements to the n-gram ranking functionality for final processing.

3.a.v N-Gram Ranking

Ranking replacements for misspelled words by Levenshtein distance alone proved to be ineffectual, as commonly more than a dozen possible replacements would exist with the same Levenshtein distance to the misspelling, many of which would make no sense in the sentence context. For example, the misspelling "caj" would generate replacements such as "cab," "cat," "car," "cad," etc. when based on Levenshtein distance alone, as all of those replacements have a distance of 1. To combat this, the replacements generated by the Levenshtein distance calculation were ranked by n-gram probability in the context of the sentence, using the 31 million n-grams (of lengths 1, 2, and 3) in the Google corpus. The BerkeleyLM toolkit and the ultra-memory-efficient n-gram model within are used to store and search the pre-computer map of n-grams.

Ranking was accomplished by selecting a "window" from the parsed sentence that contains the error and up to two of the surrounding words on each side (because the largest n-grams used were tri-grams). If the window would either extend beyond the confines of the current sentence or include another spelling error, the window boundaries are truncated such that only correctly spelled words from the current sentence are included. StyleCheck then iterates over the possible replacements generated by the Levenshtein distance calculation, calculating the probability of each window permutation (with each replacement), and ranking the replacements accordingly.

Ranking the replacements purely with n-gram probabilities commonly yielded words that work in context but are likely not the user's intended word choice, because the absolute probability of the sentence window was given total precedence over the Levenshtein distance. For example, it might have suggested "elegance" instead of "elegy," given the relative frequency of the former word in most contexts. To combat these misrankings, the replacement ranking from the n-grams is updated by the weighted addition of each replacement's Levenshtein distance from the misspelling. This added combination step ensures that the most likely replacement for the spelling error appears first.

The final step in the replacement ranking algorithm was created because of the observed tendency for StyleCheck to suggest a singular word when the plural was applicable and vice versa, and also because of observed suggestions with the incorrect tense. After ranking, the replacements are examined by morphological variant. If multiple words exist in the replacement list with the same base word

(e.g. examine, examines, examined...), the word with the variant most closely matching the misspelling's variant is given the best score of those words, a measure determined by determining the most similar suffix if two words have more than 50% of their length in common. This final processing elimates the last of the common errors from StyleCheck's spelling error replacement functionality, and overall reduces the number of errors, though it is much less accurate on shorter words where a common substring might be only two or three characters.

3.a.vi Results

The spell checking functionality of StyleCheck is, for the the most part, accurate. Errors are identified quickly and reliably, though observation of the results has yielded several valid English words that, for whatever reason, are not present in the Linux English Words list. These words have been manually added to the dictionary when noticed, but some undoubtedly remain. Additionally, StyleCheck is unable to identify whether the first word of a sentence is a proper noun (since it usually uses capitalization to identify proper nouns, and words that begin sentences are capitalized regardless), and so any word that is the first of a sentence is considered an error if it does not appear in the dictionary. The Stanford Americanizer tool is imperfect as well, and mis-translated or failed to translate several observed words from British to American English. This problem was partially addressed by replacing common spelling substitutions (e.g. "ou" $\rightarrow$ "o", "ise" $\rightarrow$ "ize", etc.), but some vestiges remain.

The spell check suggested replacements are almost always accurate, but some exceptions remain. While close enough for the vast majority of cases, the combined n-gram probability plus Levenshtein distance still occasionally mis-ranks words, especially when the n-gram in the sentence is not in the n-gram corpus, or when the n-gram window is truncated due to a sentence end or another error (as these scenarios lead to potentially inaccurate probability calculations). In general, the first suggested replacement is the desired one, and the following two are almost always reasonable substitutes.

The grammar checking component represents the least significant component of StyleCheck, as it relies in large part upon LanguageTool's Java API and the grammar checker accessible through its methods. The LanguageTool checker proved mostly accurate in identifying grammar errors, though it generated false positives on quoted text so often that StyleCheck filters those errors before displaying. A mechanism of ranking possible replacements through the grammar

scores generated by the Stanford Parser was investigated as a possible means of improving the suggestions, but this experiment proved unsuccessful (and, according to Internet research, cannot work). Though somewhat lackluster, the suggestions provided by LanguageTool for grammatical errors are a decent base from which to build a more comprehensive list.

3.b Stylistic Analysis

3.b.i Stylometric Profiling

In order to produce the necessary data to provide subjective stylistic analysis of user text and perform style similarity comparisons, StyleCheck includes a profiling function that examines a block of text for a total of 49 different metrics, 21 of which are used directly in the aforementioned stylistic analysis applications. These include statistics such as mean and variance, collected on part-of-speech usage, sentence length and variation, word length and variation, and vocabulary richness. To profile a document, StyleCheck counts each sentence and word individually, then calculates the statistical metrics after aggregating all necessary data. A document must be at least 250 words to use the stylistic analysis feature, an arbitrarily chosen threshold that suffices to prevent the inaccuracies inherent to stylometrically profiling a very short sample.

3.b.ii Overused Words

Once a document has been profiled, StyleCheck has enough information to make a variety of recommendations concerning style. One of these recommendations concerns the overuse of specific words, as undue repetition is inelegant. Overused words are highlighted for the user, and clicking on them results in suggested synonymous replacements in much the same way as spelling and grammar errors. Like those errors, the user may ignore either the specific instance of the error or the rule that identified it.

StyleCheck identifies overused words in four parts-of-speech: adverbs, verbs, adjectives, and nouns. Each individual word that falls under one of these part-of-speech categories has a separate count, and the ratio of the individual word count to the part-of-speech count as a whole is compared against the average value for the document. If this ratio is above a certain threshold of standard deviations over the mean (trained from Project Gutenberg data by famous authors, set individually by part-of-speech), it is considered an overused word, and all of its

occurrences in the document are highlighted. For example, by the statistics gathered from Project Gutenberg, approximately 12% of total words should be adverbs, and roughly 30% verbs. If the user replaces enough occurrences of the word that its usage falls below the aforementioned threshold, the word is removed from the style error list.

3.b.iii Synonym Searching

To generate suggested replacements for words that are not misspelled (as in the case when style analysis identifies a word as overused), StyleCheck uses the Princeton WordNet databases to identify likely synonyms of a given word. Since the database query returns synonyms in 117,000 synsets (which turned out to be far too many to process in real time, particularly when processing potentially dozens of words), the synsets are then sorted in decreasing order of frequency, where the frequency is the WordNet-defined frequency score for that context's usage of the given word with the given part of speech. The highest-scoring several dozen of these are retained, and the remainder discarded. Using the WordNet-generated synsets, scored by frequency, yielded far more accurate results than simple n-gram frequency; for example, the synonym replacement feature suggested "annihilated" as a replacement for "vanquished," where n-grams alone suggested the far less descriptive (but more common) "beat" and "destroyed."

Once the WordNet query is complete and a list of likely synonyms generated, the synonyms are then ranked using n-gram probabilities in the same manner as the spelling error replacements (without the Levenshtein distance weight, as word similarity is of no consequence for this application). The three synonyms with the highest probabilities are returned to the user in decreasing order as suggestions.

3.b.iv Number Processing

Numbers are handled separately in StyleCheck, with two formatting options for numerical strings. If a number is found in the document (defined as a word consisting only of digits or digits and punctuation), the format is checked to see whether it matches StyleCheck's numerical formatting rules. Numerical strings suspected of being a date or currency (or other special type) are excluded from this processing. If a "non-special" number is identified, StyleCheck generates the word form of the number, as well as a properly grouped numerical form. For example, "2000" would yield the replacement suggestions "two thousand" and "2,000."

3.b.v Document-Wide Feedback

In addition to identifying specifically overused words, StyleCheck provides a variety of document-wide stylistic feedback in a side panel next to the document. After profiling the document and analyzing the resulting statistics, StyleCheck compares the various stylometrics between the document and the trained "ideal" profile (from Project Gutenberg data). Using thresholds of number of standard deviations under or over the ideal values, StyleCheck provides written feedback on multiple stylistic topics to the user.

The first piece of document-wide feedback offered to the user is an evaluation of their vocabulary usage in the document. This score is calculated by examining the ratios of hapax legomena and dis legomena (the number of words used exactly once and exactly twice, respectively) to the total number of words, as well as the ratio between these two ratios. After comparing with the corresponding ideal vocabulary statistics, StyleCheck assigns a grade to the document's vocabulary usage and displays prewritten suggestions for improving the vocabulary score.

After the vocabulary score comes individually computed scores for part-of-speech usage over the entirety of the document. StyleCheck maintains scores for adverbs, verbs, adjectives, and nouns, and calculated their respective usage scores by calculating the ratio of each part of speech to overall word usage and comparing that ratio with the corresponding ideal profile values. Using individually defined thresholds (trained from Project Gutenberg data, with adverbs being the most severe), a grade is assigned for the usage of each part-of-speech. These grades are displayed along with prewritten suggestions for improving the scores.

The final pieces of document-wide feedback concern sentence length and variation. StyleCheck counts the length of each sentence, maintaining an average, a histogram of the lengths, and the variance of the sentence lengths. The first value is used to score the average sentence length on a scale from brief to wordy compared to the ideal profile data. The latter two values are used to generate a sentence rhythm and variation score when compared to the corresponding ideal profile values. These two grades are displayed to the user along with prewritten suggestions for improving the scores.

3.b.vi Results

Like the spelling and grammar check functionality, the stylistic analysis component of StyleCheck accomplished the majority of its goals, with a few persistent edge cases and errors that pollute the otherwise helpful results. For example, the overused words function works as intended, but StyleCheck is unable to differentiate between common connection words (e.g. more) that fall under the tracked parts-of-speech and the truly overused words. As a result, words that are common simply because of their role in the English language are commonly highlighted as overused, despite not necessarily deserving that designation. When an overused word is identified, however, the suggested synonyms are almost invariably suitable replacements that preserve the intent of the sentence.

The document-wide stylistic feedback works nearly as intended, with the sole caveat that the thresholds for what constitutes a given "grade" with respect to the ideal profile (number of standard deviations from the ideal) need further adjustment, since they are manually set. Depending on the specific document style, grades are sometimes assigned that make little sense because of suboptimal threshold values, leading to some cases where substandard writing receives an "above average" grade (though the reverse does not appear to be true).

3.c Style Similarity Comparisons

3.c.i Language Model Profiling

To accurately score the similarity between two pieces of writing requires examining a variety of metrics about both pieces, then weighting and comparing each of those until a composite score is formed. StyleCheck uses the stylometric profiling described in the previous section alongside a language model profile, which is created by tracking the frequency of each unigram and bigram that appears in the document. A similarity score is generated from the combination of a stylometric distance score (detailed below) and a probability score calculated using each language model on the opposite work. When identifying the closest author by writing style similarity, the similarity score for each of the authors in the Project Gutenberg corpus is calculated with respect to the document, and the author with the highest score displayed to the user.

3.c.ii Author Models

In order to both train the thresholds for the stylistic analysis feedback and perform author style similarity queries, StyleCheck includes precomputed models of each of eleven canonical Western authors from Kafka to Shakespeare, using four to five works each as training data (all works were obtained from the "Top 100 Authors" page of the Project Gutenberg public domain e-book repository). For each of the authors, StyleCheck traversed their works and created both stylometric and language model profiles of each work, saving them to file. These precomputed profiles were then analyzed to create combined author profiles, which were also saved to file. By precomputing these profiles, StyleCheck is able to load the data from analyzing a large amount of text near-instantaneously, allowing far more accurate comparison than would otherwise be possible in real time.

3.c.iii Stylometric Distance Calculation

While the language model similarity score is calculated in a straightforward manner using smoothed and weighted unigram and bigram probabilities, the stylometric model utilizes more subjective metrics, and necessitated the development of a proprietary heuristic to score the stylometric distance between two works. This heuristic is comprised of more than a dozen of the profile stylometrics, which are weighted and combined into a single Euclidean distance metric between the two works. The metrics used for this calculation include part-of-speech usage statistics, sentence length and variation ratios, and word choice and vocabulary data — all of the stylometric statistics that function independently of document length.

3.d Results

The style similarity functionality of StyleCheck works as intended, identifying the author of a test work (that was not used for training) correctly 7 out of 7 times (where there are 11 possible authors). This authorship identification feature also yielded interesting results when applied to the TOEFL test corpus of non-native essays used for spell and grammar check debugging. By a wide margin, the most similar authors to the non-native speakers' styles were James Joyce and William Shakespeare, probably owing to the wordiness and inverted sentence structure common among the non-native essays.

3.e User Interface

Since the goal of the StyleCheck program is to provide a user-friendly automated style tool that is accessible to even the relatively computer-illiterate, the raw back-end functionality is wrapped in a familiar word processor-style GUI with menus, buttons, and keyboard shortcuts to access both the word processing functionality and the more advanced StyleCheck analyses. Screenshots from various portions of the GUI are included at the end of this document.

3.e.i Word Processing Functionality

The StyleCheck program is on the surface a basic word processor, with a scrollable pane for text and a familiar feature set. The GUI allows users (with both menu items and keyboard shortcuts) to create a new document, open an existing document, save a document, and even open one of the eight most recently opened documents with a dedicated submenu (documents must be plain text with a ".txt" extension). In addition, it provides functions and keyboard shortcuts to select all, cut, copy, and paste. For easier viewing, users may also increase or decrease the font size, or maximize or minimize the window. Finally, the StyleCheck functionality is accessible through an "Analyze" menu, which allows the user to word count the document, spell and grammar check the document, analyze the document style, provide both the prior analyses concurrently, or identify the author with the most similar writing style. The side panel displays any results from those analysis functions that are not displayed directly on the document text.

Figure 1: Splash Screen

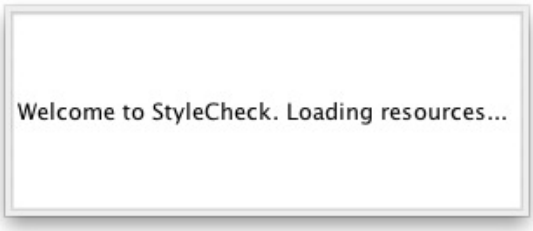


Figure 2: New Document

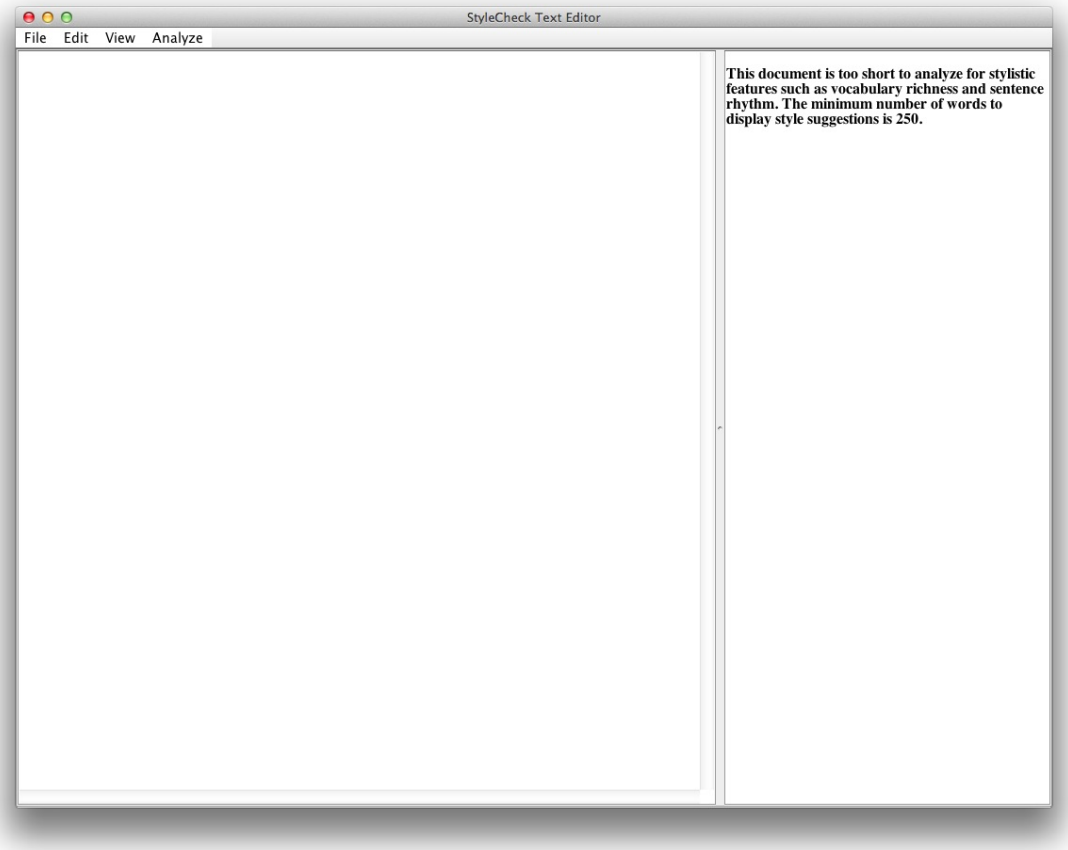


Figure 3: Open Recent Menu

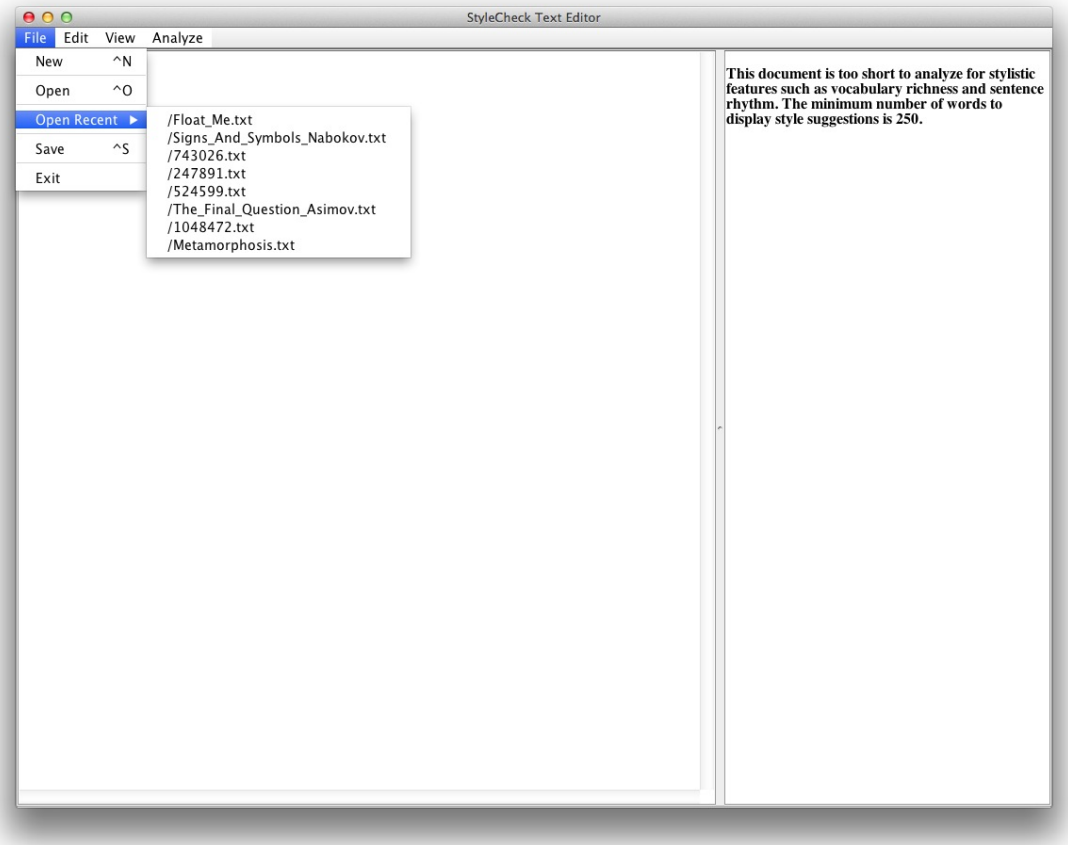


Figure 4: Newly Open Document

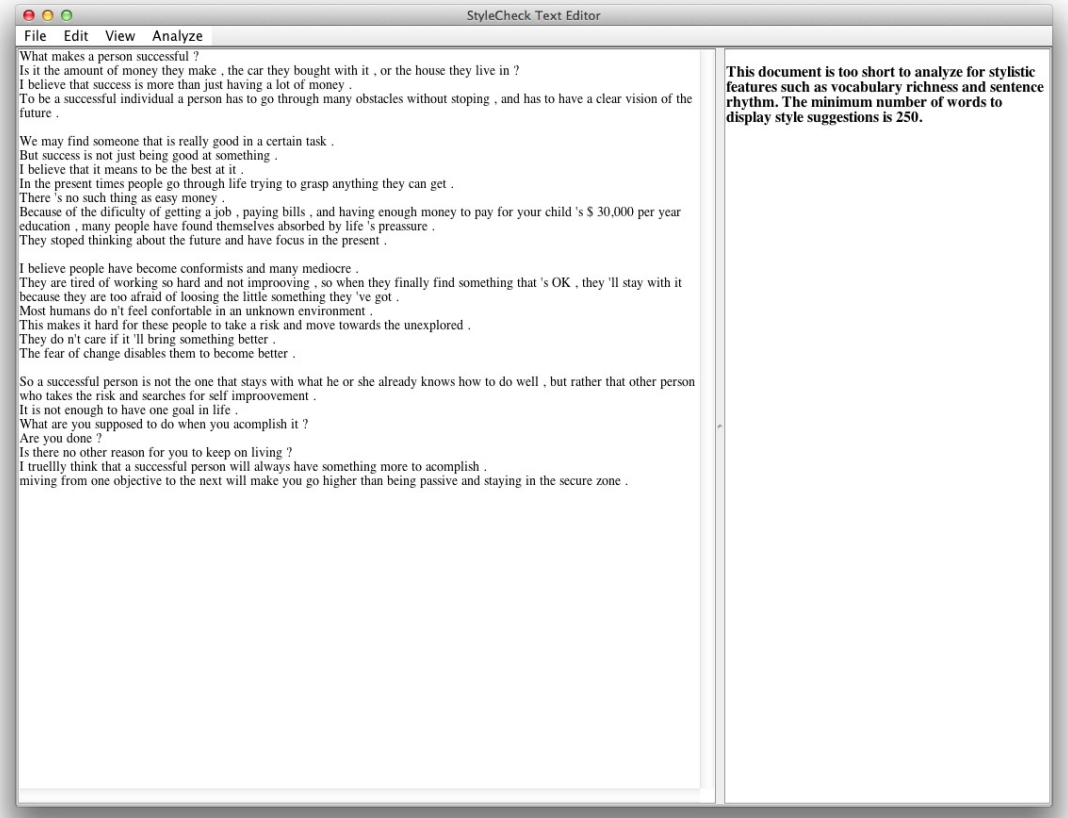
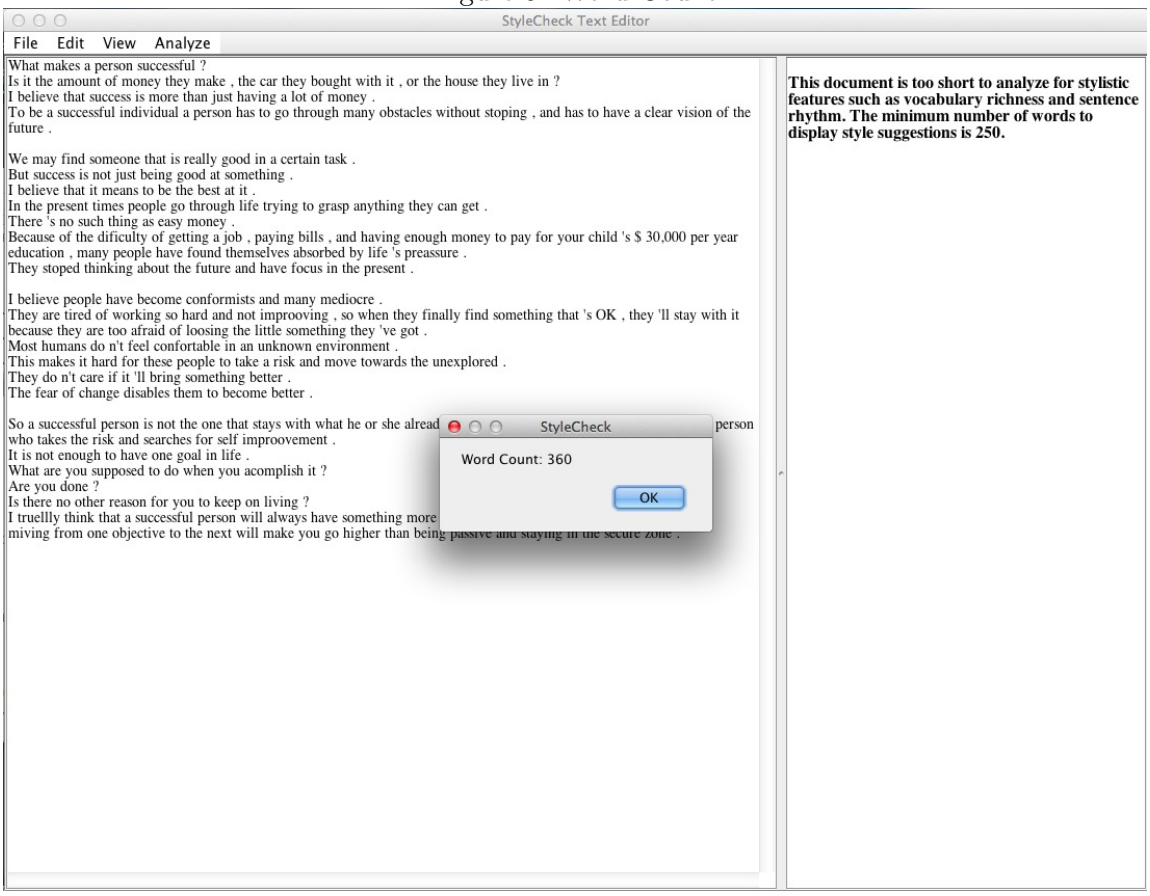


Figure 5: Word Count



3.e.ii Error Processing

Error Dialogs To allow fast and intuitive fixing of errors, StyleCheck includes GUI functionality to highlight the spelling errors (in red), grammar errors (in green), and style errors (in blue) in the document. These highlights appear automatically on each of the errors after running the appropriate analysis function, and all three types of highlights may appear simultaneously. Once the highlighted errors are displayed, the user may click on them. When an error is clicked on, a dialog box appears at the position of the error with several options. The suggested replacements (up to 3) are displayed in descending order of estimated relevance, and buttons are displayed for each that allow the user to perform the replacement with a single click. The other options presented are a button to ignore the specific instance of the error in the future, a button to ignore the general error case in the future (i.e. the spelling of the word or the grammar/style rule that was violated), and a button to return to the document without implementing a change. Choosing all but the final option removes the error from the list of errors, and removes the highlight from the error.

The Feedback Pane While spelling, grammar, and overused word errors are displayed via highlights directly on the document, the nuanced stylistic analysis of document-wide trends necessitates a lengthier explanation than is possible inline or in a dialog box, and so document-wide feedback is displayed to the user via the side panel attached to the main document text viewer. If a spelling and grammar analysis is run, the number of each type of error is displayed, along with instructions for addressing them. If a stylistic analysis is run, the document-wide feedback described above is displayed, with interspersed instructions for improvement.

Figure 6: Spelling + Grammar Check

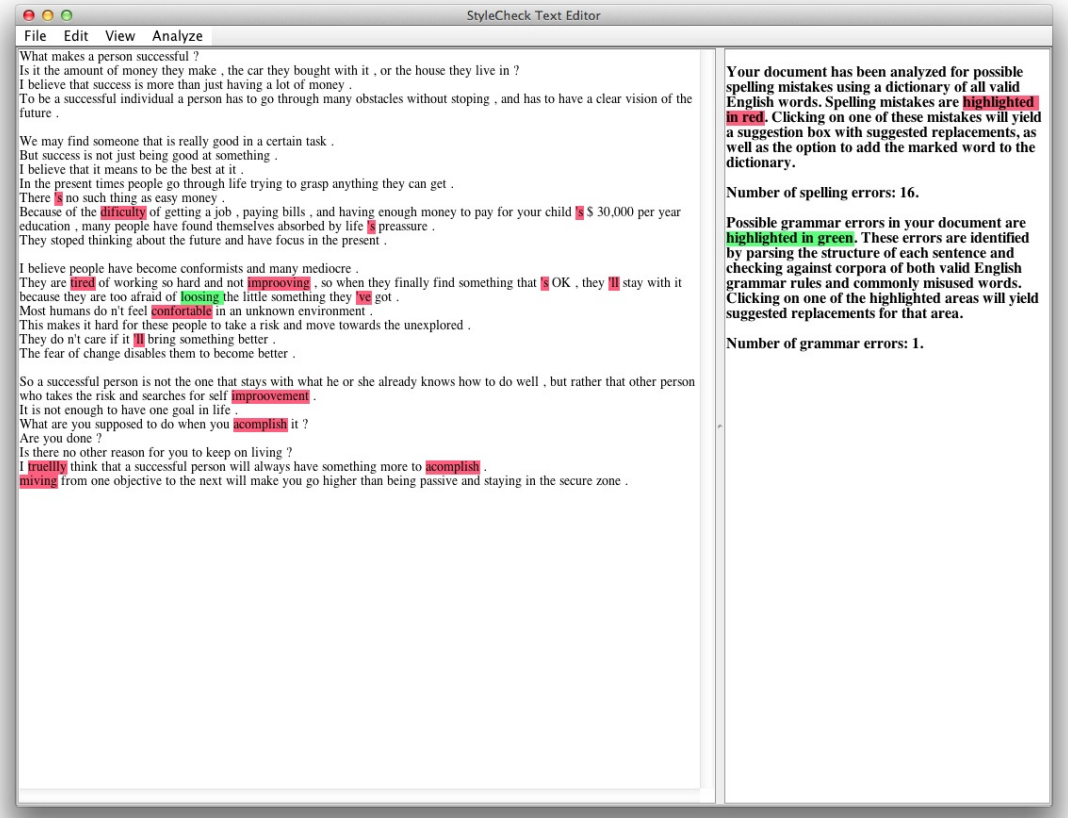


Figure 7: Spelling Error Dialog

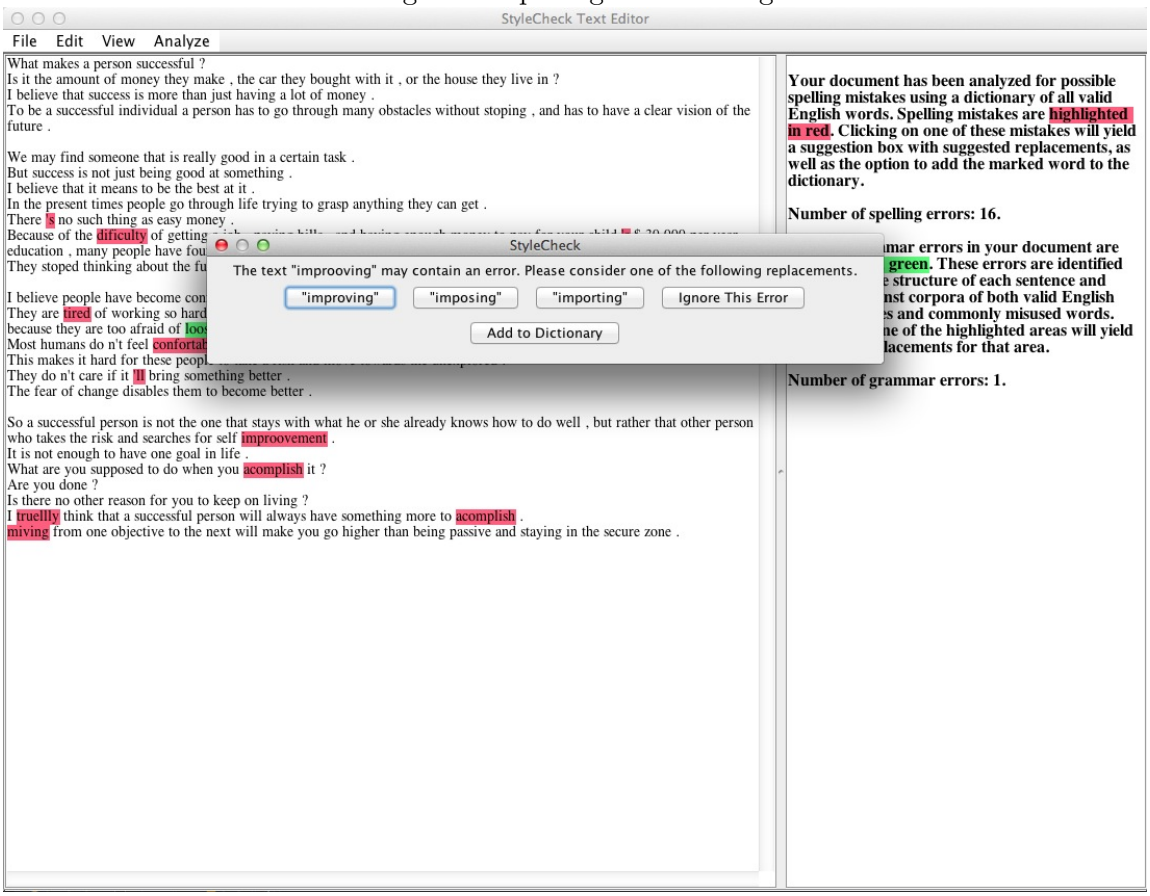


Figure 8: Grammar Error Dialog

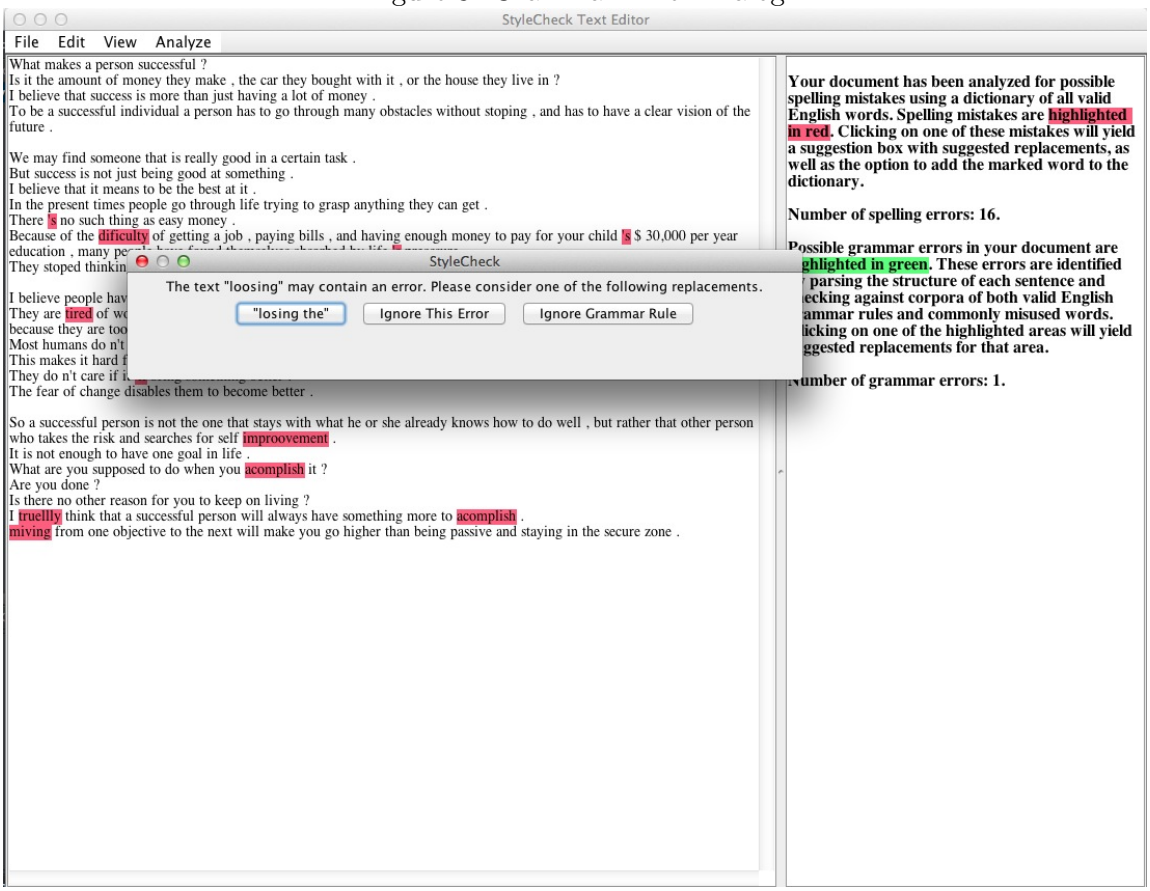


Figure 9: Style Analysis

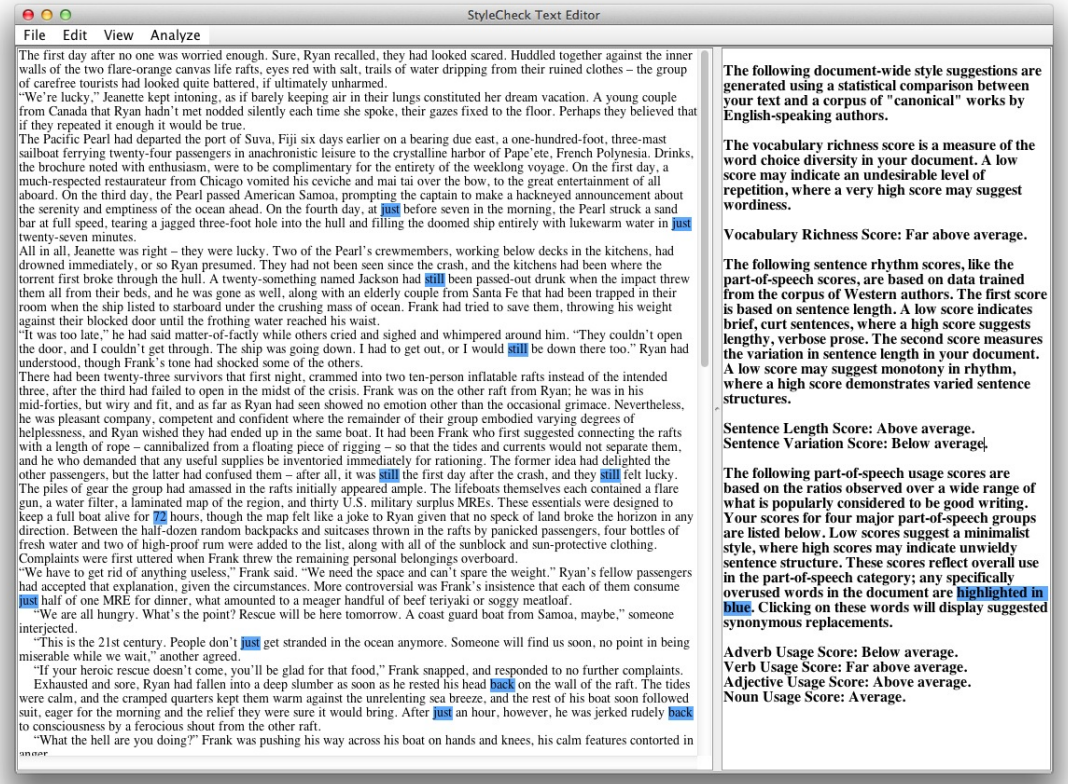


Figure 10: Style Suggestion Dialog

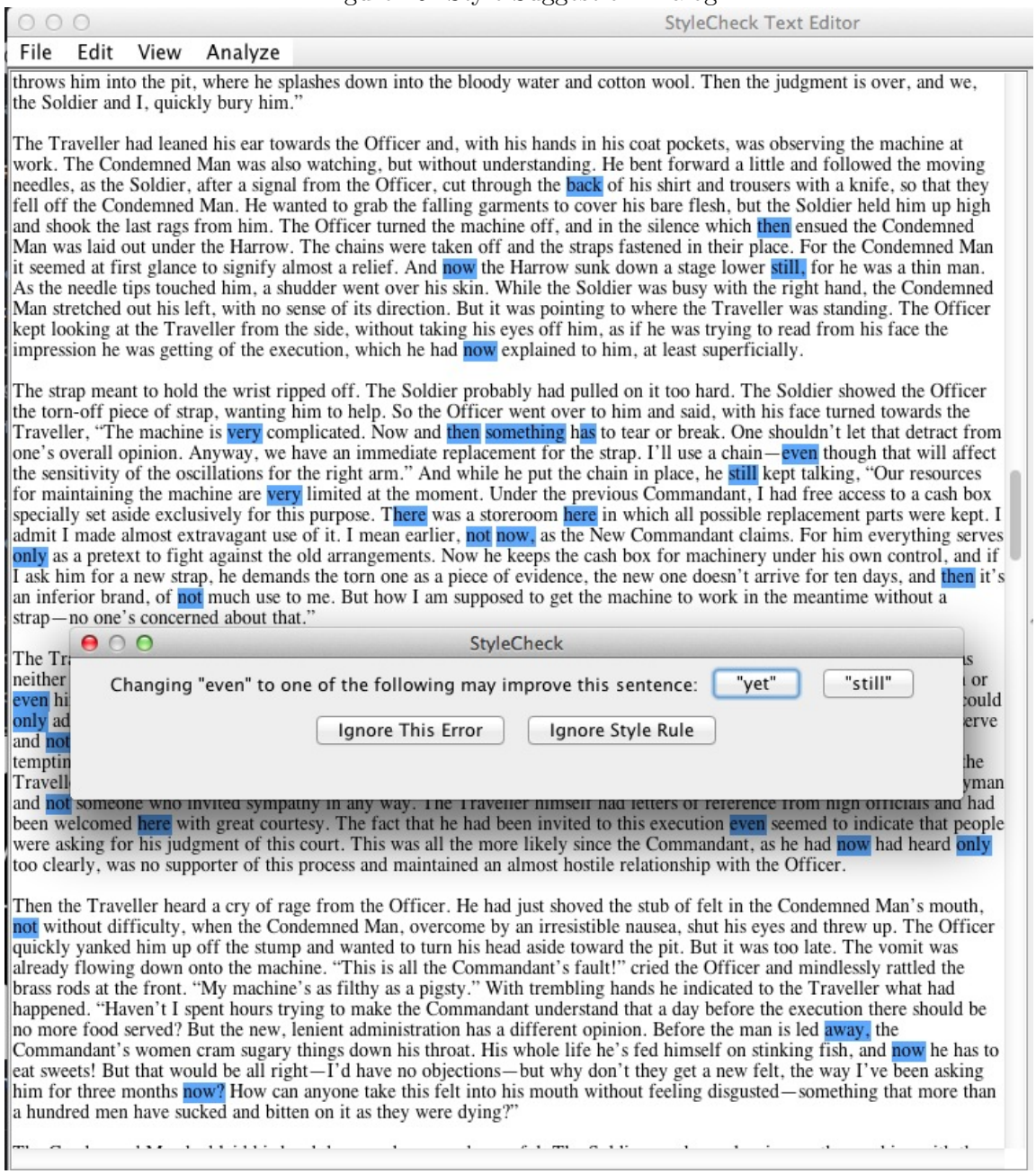


Figure 11: Simultaneous Spelling, Grammar, Style Analysis

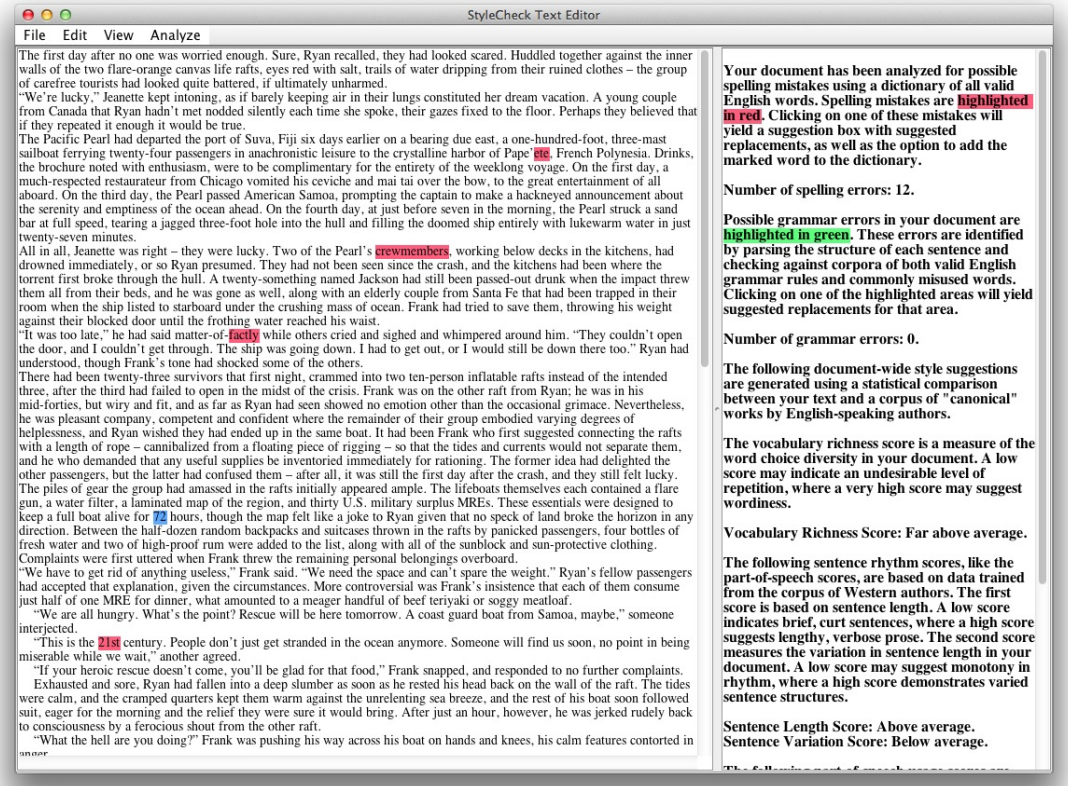
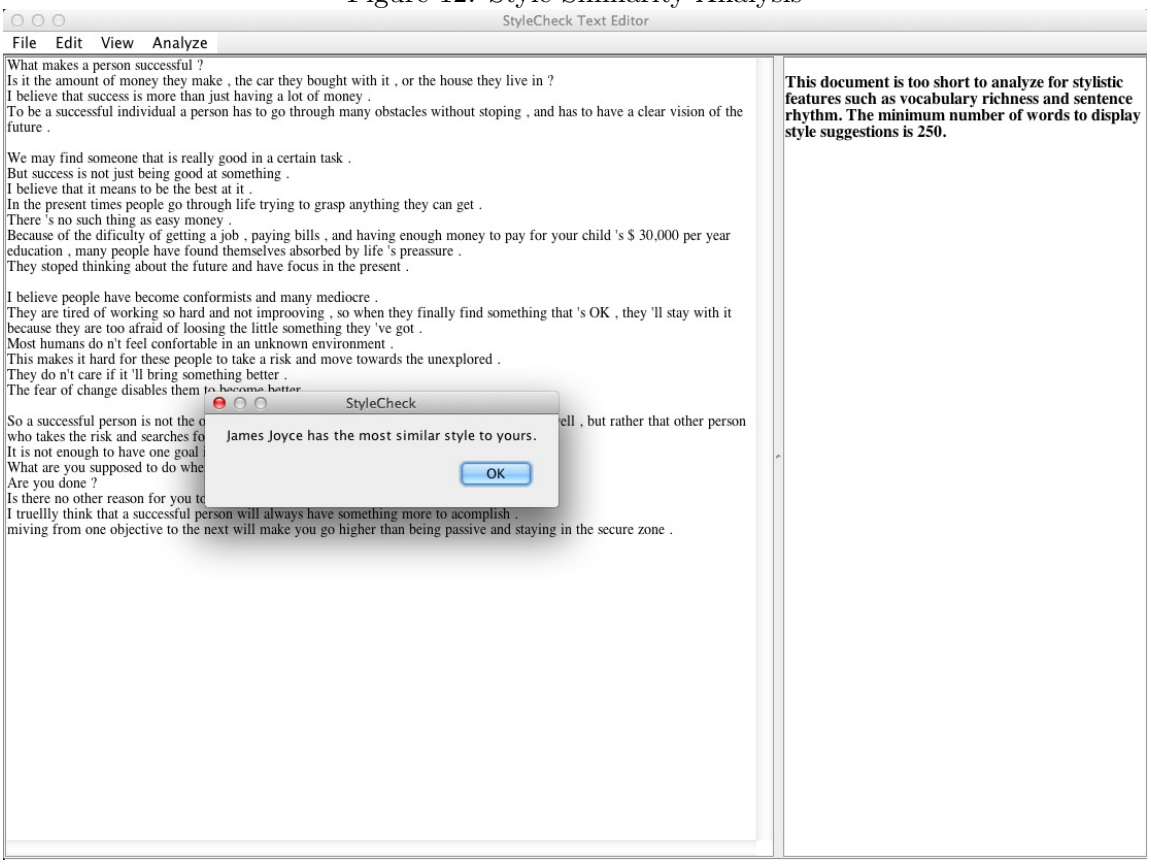


Figure 12: Style Similarity Analysis



4 User Feedback

In total, approximately two dozen people, primarily personal friends of the author, have personally interacted with the StyleCheck program and provided verbal feedback about the positives and negatives of the program.

Positives User reviews of the GUI were overwhelmingly positive, citing it as simple and easy to understand without prior experience. In addition, the most lauded feature of the GUI was the one-click replacement buttons for all of the highlighted errors. Other positive feedback included the novelty of the author similarity feature and the ease of identifying and replacing overused words using the style analysis highlights.

Negatives While in general test users felt that the program was useful, user feedback identified several shortcomings of the program that were previously unobserved. One criticism concerned the relatively limited set of authors available in the author similarity feature, a problem that could be resolved simply by downloading and profiling more works from Project Gutenberg. Another common criticism was in the vagueness of the document-wide stylistic feedback; the most common request was for the document-wide feedback to include highlighted examples of general suggestions like "vary sentence length more," with one-touch replacements like overused word errors have. In addition, users felt near-unanimously that grammar check provided the weakest suggestions of the three analyses. Finally, users requested the ability to open and save Microsoft Word documents (with a ".docx" extension, as users generally do not store documents as plain text), which would require implementing a parser capable of extracting the plain text from a ".docx" file and replacing it without affecting the internal formatting, a complicated task outside the scope of this thesis.

References

[1] Strunck, William and White, E.B., *The Elements of Style*. MacMillan, New York, 3rd Edition, 1979.

[2] *The Chicago Manual of Style*. University of Chicago Press, Illinois, 15th Edition, 2003.

[3] William K. Zinsser, *On Writing Well: The Classic Guide to Writing Nonfiction*. HarperCollins, New York, 30th Anniversary, 7th Edition, 2006.

[4] James J. Kilpatrick, *The Writer's Art*. Andrews, McMeel, and Parker, Kansas, 1984.

[5] Socher, Richard and Bauer, John and Manning, Christopher and Ng, Andrew, *Parsing with Compositional Vector Grammars*. The Stanford Natural Language Processing Group, California, 2013. Accessed at `http://nlp.stanford.edu/pubs/SocherBauerManningNg_ACL2013.pdf/`

[6] Rohan Gadad, *Automated Essay Scoring: The Great Debate*. Department of Computer Science, Dartmouth College, 2013.

[7] Christiane Fellbaum, *WordNet and Wordnets*, Encyclopedia of Language and Linguistics, Oxford Press, England, 2nd Edition, 2005. Accessed at `http://wordnet.princeton.edu/`

[8] Pauls, Adam and Klein, Dan, *Faster and Smaller N-Gram Language Models*. Computer Science Division, University of California, Berkeley, 2011.

[9] Wikipedia, *Common Misspellings - Homophones, Repetition, and Grammar*, Wikipedia: The Free Encyclopedia, 2014. Accessed at `http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings`.

[10] Wikipedia, *List of Commonly Misused Words*, Wikipedia: The Free Encyclopedia, 2014. Accessed at `http://en.wikipedia.org/wiki/Wikipedia:List_of_commonly_misused_English_words`.

[11] The Oxford Text Archive, *The Birkbeck Misspelling Corpus*. University of Oxford, U.K., 2014. Accessed at `http://ota.ahds.ac.uk/catalogue/index.html`

[12] Google, *Corpus of 31 Million N-Grams*. Google N-Grams, 2014.

[13] Linux, *The Linux English Words List*. Linux, 2014.

[14] Project Gutenberg, *Project Gutenberg: Public Domain E-Books*. San Francisco, C.A., 2014. Accessed at `http://www.gutenberg.org/`

[15] TOEFL, *Essays by Non-Native English Speakers*. Princeton, N.J., 2014. Accessed at `http://www.ets.org/research/policy_research_reports/publications/report/2013/jrkv`.

[16] Grammarly, *Grammarly Automated Essay Scoring*. New York, N.Y., 2014. Accessed at `http://www.grammarly.com/`

[17] Hemingway App, *Hemingway Automated Style Analysis*. San Francisco, C.A., 2014. Accessed at `http://www.hemingwayapp.com/`

[18] I Write Like, *I Write Like: Stylistic Identification*. New York, N.Y., 2014. Accessed at `http://iwl.me/`