**Excellent**

**Berman**

**Scriptorium: A web application for text categorization and analysis of the Hebrew Scriptures.**

Scriptorium is an interesting software program that uses author attribution methodology to disentangle the provenience of distinct sections of a multiauthored document: the Bible. The project will create an online system for text/attribution analysis whereby users can select from the online, parsed biblical corpus their own training texts (e.g., "labeled examples of texts in each specified category") that are "exploited to find a classifier based on the statistically significant common characteristics of those texts". The statistically significant characteristics that define these sets of texts (the Appendix example is for Early vs. Late Hebrew) are then used to categorize a text that the user wishes to evaluate along the same criteria measured in the analyzed texts of known provenience.

I am not familiar enough with advances in author attribution work, but it seems that the learning component of Scriptorium with its impressive "number-crunching" abilities is a noteworthy advance as is the weighing, as the PI describes it, of a "constellation of thousands of markers together toward the end of delineating a text class or category". In the same vein, the PI notes, Scriptorium will provide "simultaneous analysis of hundreds of linguistic markers to determine text categorization."

Scriptorium will be accessible to biblical scholars "with no expertise in the implemented computational methods". As a social scientist who has collaborated with computational linguists I know well the frustration of being unable to process applications that my computer science colleagues have though simple and self-explanatory. The possibility of extending Scriptorium to Latin and Greek texts, however, would seem to depend on the availability of tokenized and parsed texts

The accuracy of the characteristics selected by Scriptorium as marking a particular category, however, is dependent on the accuracy of the claim that the inputted texts are representative of the category being analyzed by the Scriptorium algorithms. For example, in Appendix C the user selects books "that define the class Early Hebrew" and those that "define Late Hebrew" to produce a set of statistical criteria that separate texts from these two periods. Clearly, then, the validity of the set of characteristics that will be given as diagnostic of Early from Late Hebrew is dependent on the accuracy of the texts that the user selected as representative.

I have some questions about the methodology.

First, it seems that if the user wishes to test hypotheses about the author of a given section of the bible the first step would be generate a series of statistics about texts by the targeted author. This is done by analyzing through Scriptorium texts that are known to be authored by the writer in question. The methodology would seem to be one of selecting two sets of texts of known provenience: Set A = texts known to be by the targeted author; Set B = texts known not to be by the targeted author. This is the impression given in appendices B and C. Yet it would seem that the best results would be obtained by holding all other variables (except authorship) constant between the two sets, A and B. In this way the statistical analysis of differences in the two sets would more closely reflect authorship, the most salient difference between the two sets of materials.

Second, there is a place for "Advanced Options" but this seems to focus on user modification of the algorithm. But it is not clear how much freedom the user has to affect the output directly by inputting his or her own ideas about characteristics that define a particular set or by inputting the results of previous research on the topic. Assume, for example, that a published article claims that a particular set of features is indexical of a given author. Can the user of Scriptorium input these features to test their validity, or use them to characterize an author's text irrespective of the data created by Scriptorium?

The preceding comments do not take away from my very positive evaluation of this project, particularly the statistical processing and display. Given that the biblical texts are already morphologically and syntactically parsed, a major developmental hurdle, the creation of a morphological and syntactic parser, is avoided. The expertise of the team to the careful consideration of implementation of a software program that will be accessible to the targeted users. The fact that Koppel has tested his computational discovery methods and that feedback will be sought on the beta version of the software is important and should help ensure adequate feedback and evaluation necessary to a highly functional final product. It is also encouraging that the major portion of requested funding is to go to software development by two contracted experts.

**Overview**

Intellectual significance: The use of computationally generated statistical models to develop a set of shared characteristics among a group of texts by a known author is here used to generate models for disentangling the provenience of various threads in a multi-author document. Although applied to the Hebrew of the bible, the PI notes that the same methodology could be used for Latin and Greek texts.

Impact on research and technology: The tool that will be developed will be web based and allow users to submit texts for computational analysis and then apply the results to find other sections of text that manifest characteristics of the same author.

Innovation: The very advantage of web-based source material (a tagged Bible source) and web-based queries to a user-inputted set of texts, might be too inflexible for some users who would perhaps want to input their knowledge of an author into the characteristics that define this same author's style.

Proposal development: The proposal is well presented and argued.

Feasibility: It seems that the algorithms and graphical user interfaces are well advanced.

Qualifications: The team is highly qualified. The budget for 4 months of software development seems appropriate for the tasks at hand.