Enabling Synchronous Collaborative Text Manipulation for Detailed Post-Edit Data Collection

For the final project, I will design and implement a database for maintaining a detailed revision history of edits of free-form text, represented as trees, along with associated discussion information. this database will power an interface for editing text, specifically in the context of post-editing machine translation output as means for collecting data for human computation approaches to generating better and alternative data for machine translation systems. The user interface has already been designed and developed (http://students.washington.edu/kuksenok/imt), but the existing database does not support robust checks for current/past state(s) that would enable synchronous, collaborative editing and branching using a web application on multiple machines. The purpose of this project is to create an alternative database to the one in place and compare its performance.

Statistical machine translation (SMT) models use large parallel corpora of sentences translated from a source into a target language. Obtaining parallel corpora can be difficult and expensive, motivating interest in human computation approaches for generating such corpora. However, human expertise expressed in the process of editing a translation is lost by systems that consider only the final result. I work on a project that seeks to investigate whether/how fine-grained edit data representing changes as they are made by human contributors can positively impact the reach and effectiveness of human participation in improving machine translation. Fine-grained edit history can be used in novel ways, both in enabling real-time feedback for human editors (e.g., suggestions), and incorporating human feedback into a human-in-the-loop MT mechanism, such as by applying statistical methods for automating the iterative improvement of translation output. This work has thus far involved design of a UI, an initial prototype, and preliminary experiments with the web-based prototype [3]. The existing application is built using PHP, using mysqli to access a MySQL database (on a UW webserver) and create appropriate abstractions for the application layer. A new implementation of the database may require changes to the specific queries needed, but the abstractions will remain unchanged, allowing the consistent use of the same front end.

Although the overall goal is to use this database to maintain a record of edits enacted on text, the fundamental representation is trees. Suppose a statement in the *source* language (here, English) reads, “the cake is a lie,” as an initial MT system (in this case, Google Translate) translates the statement as “el pastel es una mentira” into the *target* language (here, English. In the process of translation, the system has also identified a specific phrase translation *alignment:* “the cake” → “el pastel” and “is a lie” → “es una mentira.” We can model the translation itself as a transformation of “is a lie” into “es una mentira.” Because the target translation can mean the same thing with only “es mentira” (without “una”), a viable edit can be to delete “una,” which can be represented as a transformation from “es una mentira” into “es mentire.” To maintain the semantic meaning of deletion, and the semantic meaning of phrases, we represent words as leaf nodes, phrases as parent nodes to associated words, sentences as parent nodes to phrases, paragraphs as parent nodes to sentences, and so on, to incorporate sections, chapters, book parts, stanzas, and other organizational elements that exists for various types of texts. This representation affords maintaining a detailed record of changes made to the text, which is a key functionality of this application for gathering post-edit machine translation data.

In addition to maintaining a record of edits, it is necessary to enable the editors of translation to carry out conversations as they collaborate on changes. In this way, the challenge is similar to that addressed by DB-Wiki [1], but limited to text; further, our goal to enable robust synchronous collaboration requires a web-based implementation with minimal delay-inducing overhead, such as that introduced by using a Java-based layer. The representation of the tree model of text itself can be done either by modeling changes as sequences of edits, which can upon request be used to create snapshots of the text at different points in time; or in a way that allows much more efficient ability to view the state of the data at a particular time, at the cost of changing the tree structure at the time of an edit [2], which would also help with reliability.

By the May 13th milestone, I will have finished designing the database to be integrated with the existing UI, as well as the experimental evaluation plan for comparing the existing database design to the new database design, and begun to integrate the new DB with the existing front-end. After implementation, I would need to instantiate the existing and the new database designs with test data to compare relevant performance, prior to presenting the results on May 29th.

1. Buneman P., Cheney J., Lindley S., Mueller H. The Database Wiki Project: A General-Purpose Platform for Data Curation and Collaboration. ACM SIGMOD Record 2011.
2. Buneman P., Khanna S., Tajima K., Tan Wang-Chew. Archiving Scientific Data. ACM Translations on Database Systems 2004.
3. Kuksenok K., Brooks M., Bangalore S., Fogarty J. Collecting High-Quality Fine-Grained Human Feedback on Machine Translation Errors with *ChoiceWords*. Submitted to ACL2012.