Merging Duplicates

Craig Farrow
Version 2
May 2014

Introduction

It is a common request from FLEx users for an automated way of merging duplicate entries. The typical scenario is where two or more databases have been merged, and there is significant work involved in removing all the redundant entries. A number of issues arise in this process, such as:

1. The manual merging process within Flex is very time consuming.
2. Often there is unique data in each entry, so the user doesn't want to simply delete the duplicate entries.
3. There can be conflicting data, where the value from one entry is to be preserved, and the other discarded.
4. When discarding conflicting data, the order in which entries are merged can affect the result.
5. Some homographs might legitimately be different entries, so shouldn't be merged.

This document presents a proposal for a semi-automated merge system built with FlexTools. The goal is to reduce the amount of manual work required when merging a large number of entries. Since there are dangers of an automated system making irreversible mistakes, the proposed system will enable the user to review all changes beforehand. After bulk merging there will still be a need for manual clean-up. The FLEx bulk edit tools and possibly additional FlexTools modules should be helpful for this task.

FLEx Merge

The FLEx Help lists the following notes about the behaviour of the manual merge (since version 6.0.1):

> *If the target entry has existing content in a field, any content in the equivalent field from the merged entry is appended into the target entry.*

If the target entry and merged entry have different grammatical information, additional senses are created.

When merging entries, if the two lexeme forms are identical, the two entries merge without changing the headword.

When merging entries, if the two lexeme forms are not identical, the lexeme form from the source entry will become an allomorph of the target entry. This maintains interlinear text morpheme lines.

Proposed Solution

Two FlexTools Modules will be provided that implement a two-step process. The first module will put tags in the FTFlags entry-level field indicating potential candidates for merging. The user then reviews those tags and edits them to control the merging. Next the second module merges the entries and updates FTFlags so that the user can review the changed entries. The details of this process is as follows:

1. FlexTools Module *Report Duplicate Entries*
    a) Ignore:
        i. affixes;
        ii. entries with data in FTFlags;
        iii. entries with multiple or undefined grammatical info (i.e. POS)
    b) Find all sets of homographs with:
        i. the same MorphType, and
        ii. the same single MSA (Grammatical info). I.e. only group nouns with nouns, etc. If an entry has more than one grammatical category (e.g. Noun and Verb), then it will be ignored.
    c) Write the tag "m" to FTFlags in all entries found in b) above.
2. The user reviews the tagged entries and edits FTFlags as follows:
    a) Filter the FTFlags field for the value 'm' (and select Whole Item)
    b) Control the merge operation by setting FTFlags to:
        i. 'mt' for merge target: this specifies the 'master' entry that all others will be merged into. If there is no entry tagged 'mt' then one of the entries tagged 'm' will be used as the target.
        ii. 'md' to specify that text data from this entry is to be *discarded* if there is data in the target entry.
        iii. 'm' to specify that text data from this entry is to be *appended* if there is data in the target entry.
        iv. 'del' to delete the entry.
3. FlexTools Module: *Merge Entries*
    a) If there is an entry tagged 'mt' this is the target entry, otherwise choose one of the entries tagged 'm.'

b) Merge entries tagged ‘m’ and ‘md’ into the target entry. The ‘m’ entries are merged first, *appending* any duplicate text data. The ‘md’ entries are merged second, *discarding* any duplicate text data.
c) All entries that were merged are tagged with ‘merged’ in the FTFlags field.
d) Delete all entries tagged ‘del’

4. The user can review all the entries that are tagged ‘merged.’

Other Use Cases

- The user sets up some merge command tags in FTFlags before running *Report Duplicate Entries.* Whatever is already set will be ignored. For example 4-way merges can be configured manually, or true homographs can be tagged with ‘-’ or ‘ignore’.
- The user finds one set of duplicates to merge. They enter command tags into FTFlags and run the *Merge Entries* module only.

Technical Information

ILexEntry overrides the MergeObject method and implements the above behaviour in these steps:

- creates an allomorph if the source lexeme form is different
- merges the lexeme form (e.g. non-default WS values may be different)
- fixes lexical entry references
- merges all the fields (appending any collection/sequence items)
- merges allomorphs
- merges MSAs (the grammatical information)
- updates homograph numbers

There is one binary argument (fLoseNoStringData) which controls whether conflicting text data is merged (i.e. appended) or discarded (i.e. the source data isn’t copied if the destination field contains data.)