Merging Duplicate Entries in Fieldworks

Language Explorer

Craig Farrow
Version 3
June 2014

Introduction

It is a common request from Fieldworks 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 Fieldworks 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. Entries may be associated with texts or grammatical analyses, in which case deleting one entry is not desirable.
4. There can be conflicting data, where the value from one entry is to be preserved, and the other discarded.
5. When discarding conflicting data, the order in which entries are merged can affect the result.
6. Some homographs might legitimately be different entries, so shouldn't be merged.

This document presents 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.

There are dangers in any automated system of making irreversible mistakes, and so the system enables the user to review all changes beforehand. After bulk merging there will typically be a need for manual data clean-up. The Fieldworks bulk edit tools and possibly additional FlexTools modules will be helpful for this task.

Fieldworks Merge

The Fieldworks Help (version 8.0) notes the following about the behaviour of the manual merge (since version 6.0.1)[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.*[2]
>
> *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.*[3]

FLExTools Solution

Two FLExTools modules are provided that implement a two-step process. The first module puts 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.

A third module is provided for merging duplicate senses. This functionality is split into its own module so that the user can control the steps more carefully. Also, the ability to merge duplicate senses is useful even when there has been extensive manual merging (since the manual merge doesn't merge duplicate senses.)

The details of this process is as follows:

1. FlexTools module *Report Duplicate Entries*
   a) Ignores:
      i. affixes, variants and complex forms (except phrases);
      ii. entries with data in FTFlags;
   b) Finds all sets of homographs with:
      i. the same MorphType, and
      ii. the same set of Grammatical Categories. I.e. group nouns with nouns, etc; if an entry has more than one Grammatical Category (e.g. Noun and Verb), then it is grouped with other homographs that have exactly the

[1] Under the "Important" section in the topic "Merge entries."
[2] Note, that even if the grammatical information is the same, the senses are just moved to the target entry—the senses themselves are not merged in any way.
[3] In the present FLExTools system, only homographs are merged, which means that this scenario of merging entries with non-identical lexeme forms isn't supported. In this case the user can either manually merge the entries, or else unify the lexeme forms and then use these modules.

same set of Grammatical Categories. (Entries with different combinations of Grammatical Category will have to be merged manually.)

c) Writes the tag "m" to FTFlags in all groups of entries found in b) above.
d) If any homographs turn out to have only one in the set, then these are reported to the user, and "review" is written to FTFlags.[4]

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) For all homographs with the same MorphType and same set of Grammtical Categories:
      i. If there is an entry tagged 'mt' then select this as the target entry, otherwise chooses one of the entries tagged 'm.'
      ii. Merges 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.
      iii. All entries that were merged are tagged with 'merged' in the FTFlags field.
   b) Deletes all entries tagged 'del'
4. The user can review all the entries that are tagged 'merged'.
5. FLExTools module *Merge Senses*
   a) For all lexical entries with top-level senses that have the same Grammatical Category and the same Gloss:
      i. Merges later senses into earlier matching ones. Definitions, etc. are *appended*.

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 a set of duplicates to merge. They enter command tags into FTFlags and run the *Merge Entries* module only.

[4] For example, we don't want to assume that homographs with [Noun] and [Noun, Verb] should be merged, but we want to flag these so the user can decide what to do. If they should be merged, then that can be easily done in Fieldworks.

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 (since non-default WS values may be different)
- fixes lexical entry references
- merges all the fields (appending any collection/sequence items including senses)
- merges allomorphs
- merges MSAs (the grammatical information)
- updates homograph numbers (handled when the old entry is deleted.)

There is one boolean 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).