Release Notes for TAMS Analyzer 1.01

1. Cancel button added to Code Definition Sheet

Users can now choose to cancel when defining a new code. The text will not be coded and no code is added to the central code file, if the user picks cancel on the sheet.

2. Options for end of file handling

One major problem with late versions of 1.0 was the handling of the end of file. Basically nothing was done for end of files if no {!end} tag was encountered. This new version has a preference option for treating the end of files as though they had an {!end} put there, for most circumstances, and especially for beginning users (who may not be sticking {!end} tags in their documents). If the user wants to take charge of the placement of {!end} tags, they can turn off this option.

3. Depreciating codes

Often users when they recode choose to retire codes in favor of other codes. Rather than delete the old code, users may want to keep the definition in the central code file, but mark them as "retired" so they don't appear in the code list. Putting the word *retired* as a comment in the closing tag of the code definition (in centralized code files) will now keep the code from appearing on the list.

4. XML export of search results

While some users have requested implementing TAMS as an XML kind-of-look-alike system, I've resisted. The one place I have found that XML might make sense in terms of sharing results is in saving the results of searches. That export option is now available in the SAVE TO… dialogue.

In this relatively unsupported version the root tag is <tamsdata>. Each row of a search is returned as a <tamsrecord> in which each column is given a tag in order (so depending on universal and repeat codes these tags composing a <tamsrecord> could vary! Here is a very simple (one record) XML file exported from a study I'm doing on Human Subjects Review Board paperwork:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tamsdata>
      <tamsrecord recnum="1">
            <site>KSU</site>
            <type>full</type>
            <_code>product</_code>
            <_data>{product}20.\tWhat do you intend to do with the data
collected?  (i.e., publish data, present paper, erase tapes,
etc.)\n\n{/product}</_data>
            <_begin_loc>14220</_begin_loc>
            <_end_loc>14340</_end_loc>
            <_document>ksu.rtf</_document>
```

```
        </tamsrecord>
</tamsdata>
```

This file has <site> and <type> as universal codes, and no repeat datas. The other fields are standard TAMS columns.

TAMS Analyzer 1.1 Notes

TAMS Analyzer 1.1 adds additional tools for analyzing results. It incorporates some (not all) of the ability of DataBoiler, and should decrease time spent sorting and selecting in programs like Filemaker and Excel (which is where my data often ends up).

Changes/Improvements in version 1.1

-Menu renaming and rearranging. Prior to 1.1 there was basically one long TAMS menu. The functions in that menu have now been split into two menus: one for coding and one for analyzing and working with results.

-Results can now be sorted in a variety of ways (alphabetically, numerically, by date). They can be sorted on multiple criteria (using the "sort within" functions). Sorts can now be up or down, before only sorting up was possible.

-Results can be further analyzed by searching within the fields that TA returns (Selecting). Complex subsets can be created using Select (which selects from the visible records) and Select additional (which selects from all records found in a given TA search). Selecting can be done through text, numeric, and date criteria. Date formats can be chosen from an easy to use sheet, and other date formats can be entered by picking "Other" and entering a format using Apple's date string formats. Results sets can then be recoded/or add-coded as appropriate.

-Result interface made slightly tighter to provide room for a record count field. This always shows the number of records showing over the number of records total returned from TA search.

Notes on release 1.10a6

- fixed record counter so it's fixed in position to upper left corner
-select reverse
-select regular expressions using AGRegex
-error message restored if you try to search with no data files but have an init

Notes on release 1.2

New features:
-select reverse
-select regular expressions using AGRegex for both find/replace in data documents and in results

Bug fix:
- fixed record counter so it's fixed in position to upper left corner
-error message restored if you try to search with no data files but have an init

Version 1.2 adds two major features to the selection of records: select reverse, which shows all records not showing in the current results window from a search, and regular expressions which allows much more subtle searching through results. Regular expressions or regex has also been added to the find and replace dialogue for documents. This is implemented through the AGRegex class, which is a wrapper for PCRE (Perl C regulear expressions). See that documentation and other documentation around the web for how to use this powerful (possibly deadly feature).

Notes on 1.1 and 1.2

1. Selecting results. The key new feature of 1.1 is the ability to "select" results. In essence, one can "reduce" the results by searching within them for particular strings or for date and numeric criteria. A simple example: you have a set of documents concerning international treaties. You have marked them concerning discourses concerning identity and boundaries. Now you can search your documents for the code "boundaries>national". Now that you have the results, it would be nice to look "in" the results for records that reference "Spain". Click on the head of the "_data" column. and pick "Select" from the Results menu. Fill in Spain and choose the type of search you'd like from the pop-up menu. For text searches you can also use regular expressions by checking the regular expression box (See the documentation for PCRE for details on regular expressions). Now all the results window will show are those records with the word "Spain" in the _data column. You can get back to all of the results by picking "Select all" from the Results menu. You could also find all records that do not contain "Spain" by picking "Select reverse." (1.2 only)

2. Sorting results. Shown results (not the same as all results since you may have used the select feature described in 1) can be sorted up and down on multiple columns. To sort the codes alphabetically, pick the alpha menu option from the "Sort up" sub menu from the Results menu (Results->Sort up->alpha). If you want to sort your data within the sorted codes, pick "Results->Sort up->alpha within." If you keep picking the within options, the computer will remember your previous sorted information. To wipe out computers memory of which column is sorted choose the options without the "within" word.

NOTE: You should always do your first sort using the non-within menu options.

3. Regular expressions are the key new feature of version 1.2. Regex's are a syntax for specifying complex patterns that can be searched. The format is very succinct and learning to use them effectively is practically like learning a programming language. One quick example, and then you're on your own. Imagine that you have a focus group on whether the U.S. should go to war. Here is one exchange:

Bob: I don't believe in war.

Sally: Bob, you're a dupe.

Now we have to code these. You can use one search and replace to turn every instance of someone speaking to their name surrounded by {speaker}{/speaker} tags.

Using the "Find->Find…" menu option you fill in

```
^([a-zA-Z]*)\:
```

and in the replace you fill in

```
{speaker}$1{/speaker}\:
```

You can see how absolutely cryptic this is, but it will surround everyone's name with the speaker codes so that the results of clicking the Replace All button will be

{speaker}Bob{/speaker}: I don't believe in war.

{speaker}Sally{/speaker}: Bob, you're a dupe.

It would do this for your whole document. That's a lot of coding time saved!!!

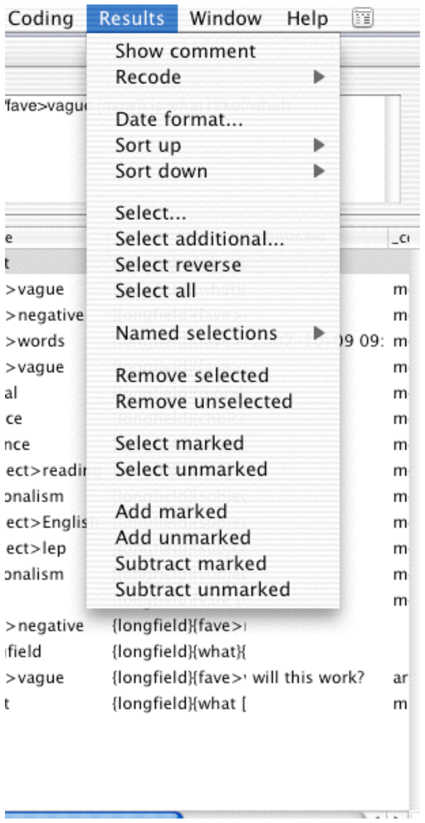
Just so you understand what those weird strings mean:
^ says start at the beginning of a line when matching, the parentheses mean remember this part of what you find. [a-zA-Z] means letters only; * means 0 or more times (meaning multiple letters, not just one letter); and the colon is just matching that character.

In the replace string, the $1 refers to anything it found in the parentheses in the search string. If you had two things in parentheses, things matching the "regular expression" in the second one could be referenced with $2. $0 refers to the whole matched string. Complex. Don't use it if you're not comfortable with it, just keep it unchecked. Read the documentation in the AGRegex folder for more (and important) details about what you can and cannot do with this implementation of regex..

NOTE: Only three escape codes are supported in the replace string (all PCRE escape characters are supported in the find string): \t for tab, \n for new line and \r for return. All other escaped characters are taken as literals. That way you could add returns after every paragraph by searching for \n and replacing it with \n\n.

NOTE: Regular expressions can also be used in the Select box of results.

TAMS Analyzer 1.21: New tools for qualitative analysis

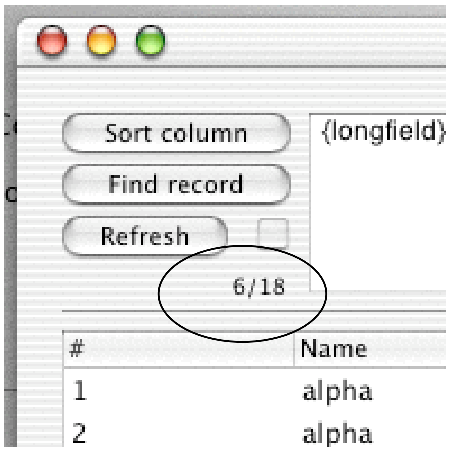


TAMS Analyzer has two key concepts involved in analyzing data: marking and selecting. Marking has been in TA for a while. It is a way of indicating which sections in text returned by a search you want to recode or add a code to, back in the original documents.

Selected records simply refers to the records (i.e., the rows of data provided in a search in TA) that are visible. Initially all of the records of a search are visible. By using the Select… menu option on the results menu you can view just a portion of the returned results. Basically each results window keeps two lists: the records you can see, and all the records the search returned. Saving to… will save only the selected (i.e., visible) records. This way you can save very specialized searches for further analysis in FileMaker or Excel. Once you've selected a few records, you can add to the visible pile by picking "Selecting additional." You can also reverse which records are selected and "unselected" by picking "Select reverse." You can also get back to the pile of all records by picking "Select all."

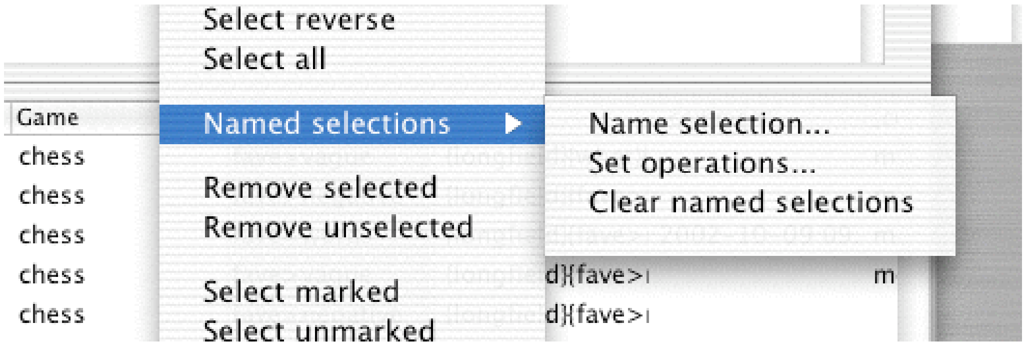
Selecting and its variations ("Select additional", "Select reverse", etc.) does not delete records from the pile of results returned from a search. It just hides and shows different ones.

Two menu items do remove records from the total pile. "Remove selected" and "Remove unselected" effect the total pile, not necessarily the visible pile.



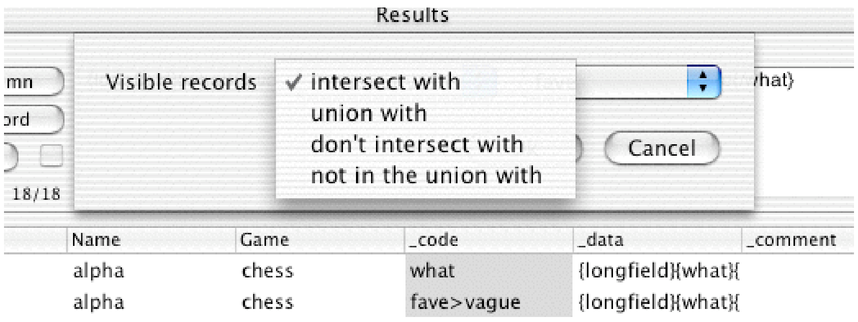
The number under the refresh button check indicates both the number of records showing and the total number of records showing

As I already noted, marked records which are picked using the Results->Recode submenu, determine which records will be recoded back in the original document. Marked records have a + by the record number. There are 6 menu options that allow the selected records to be related to the marked records. You can select (i.e., make visible) the marked records by picking "Select marked." Conversely you can select all the "unmarked" records by picking "Select unmarked." If you have a subset of records showing, i.e., a selection, you can add all the marked records from the total records pile by picking "Add marked." Conversely, you can add all the unmarked records from the total pile to the currently viewed selection by picking "Add unmarked." Finally, you can remove any marked records from the visible selection by picking "Subtract marked." Conversely you can remove all of the unmarked records from the visible selection by picking "Subtract unmarked."



For even more advanced analysis of data you can used "Named selections." This simply means that you can give a name to a collection of data, i.e., to the visible records. The name will appear at the end of the "Named selections" submenu. You can look at those records just by picking it's name from the menu! It's a way of bookmarking a selection.

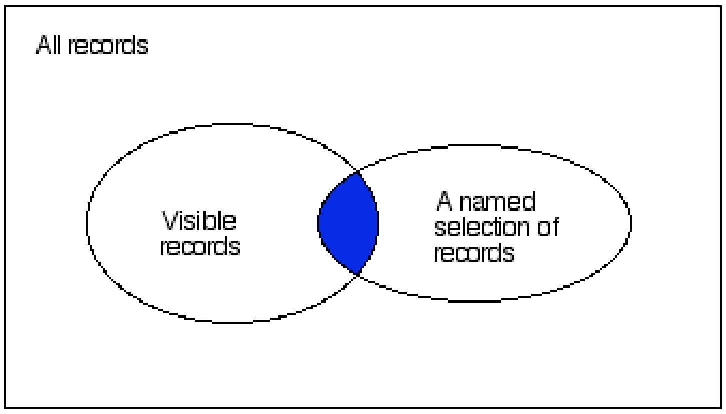
Once you have named selections of data you can do operations to look at the intersection of named selections, or combine named sets into new selections. This is done through the "Set operations…" menu. These operations are done with the current selection (whether it has a name or not). Picking "Set operations" provides you with a list of operations and a list of named sets:



Name	Game	_code	_data	_comment
alpha	chess	what	{longfield}{what}{	
alpha	chess	fave>vague	{longfield}{what}{	

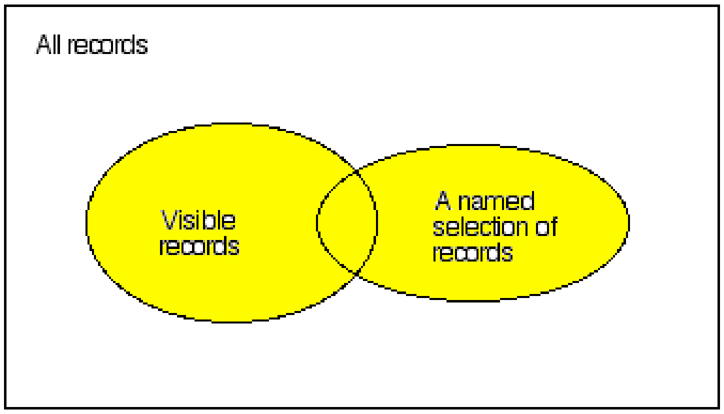
It is helpful to think of the results of these using Venn diagrams. As a user, you will have before you a selection of records, these I mark with a V; there is also the pool of all records, and the records that are in the named selections you have created.

1. Picking intersect with can be represented by



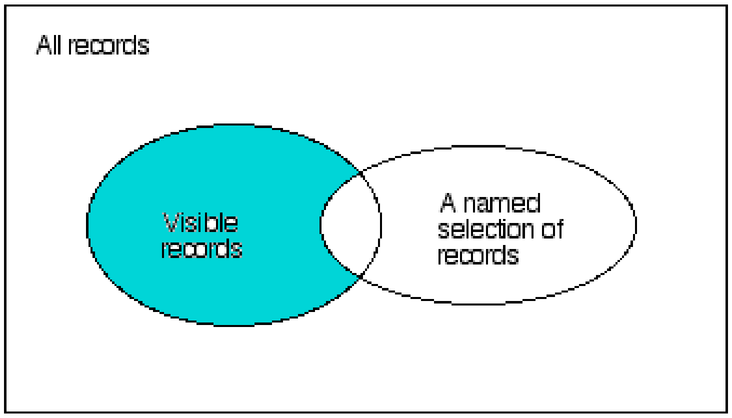
The colored area represents the domain of records that would be returned by this option.

2. "Union with" combines the visible records with the records in the named set. It would be represented by

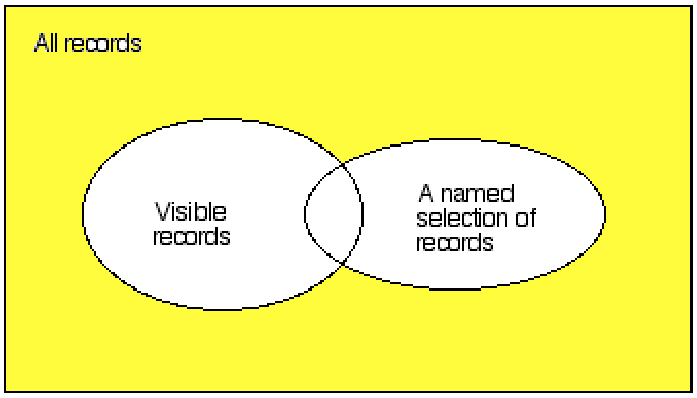


The yellow zone shows the records that would be returned.

3. “Don’t intersect with” returns the visible records not in the zone of intersection. Visibly it would be represented by



4. Finally “Not in the union of” returns all records not in either set:

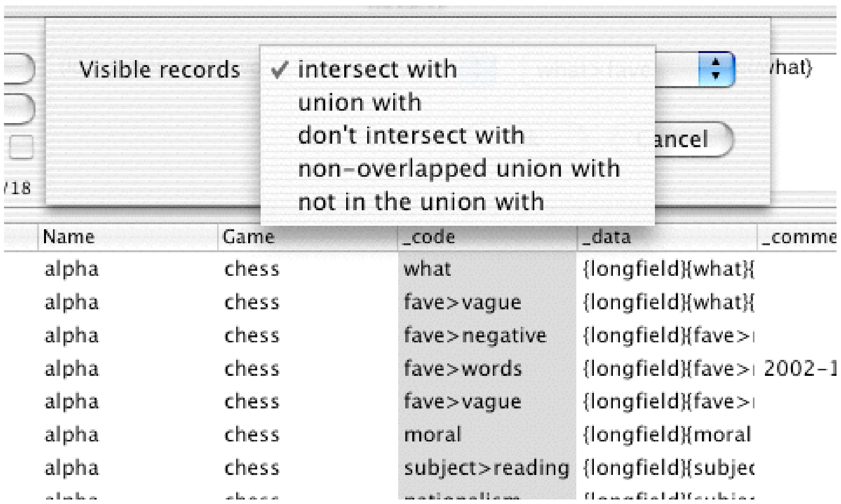


By combining these you can do very complicated analyses of your data finding the information and patterns in your documents. Remember you can always further select your data by clicking on a column (usually _code, _data, or _comment) and selecting additional times. Also remember that you can sort your data to find patterns.

Note Named selections are wiped clean when you do anything to alter the pile of all found records, this includes refreshing results, as well as removing marked records or unmarked records. That will basically clear the “Named selections” menu.

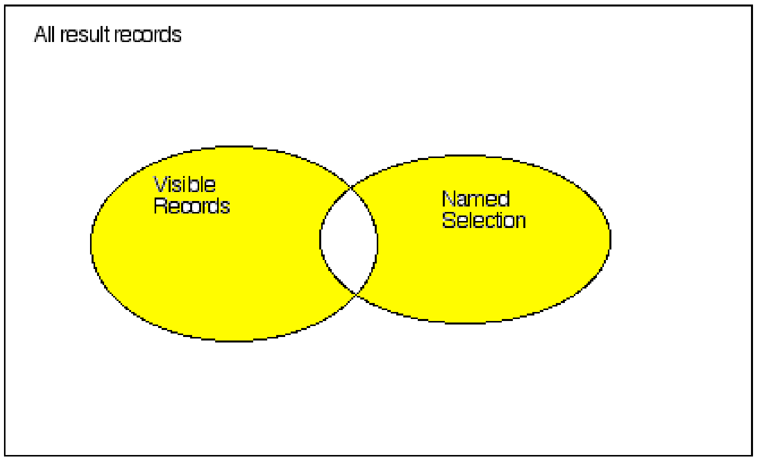
1.21a5 Release notes

TA1.21a5 adds one additional set operation under the Results->Named selections…->Set operation sheet.



Name	Game	_code	_data	_comme
alpha	chess	what	{longfield}{what}{	
alpha	chess	fave>vague	{longfield}{what}{	
alpha	chess	fave>negative	{longfield}{fave>	
alpha	chess	fave>words	{longfield}{fave>	2002-1
alpha	chess	fave>vague	{longfield}{fave>	
alpha	chess	moral	{longfield}{moral	
alpha	chess	subject>reading	{longfield}{subjec	

The new operation is called "non-overlapped union with" which is a logical exclusive or (XOR). it will return a union minus the part that overlaps. A venn representation of this would be



1.22 Release notes

Changes:

sorting up and down numeric values now sorts up and down the correct way rather than backwards.

the EOF of the init file is ignored on empty searches (only affects you if in your preferences you have indicated you want EOF treated as {!end})

Bug fix in regex search and/or replace in selections.

preference option now available on whether universals carry their values across documents

new metatag to allow the contingent setting of repeat and universal variables (see below)

replace dialog variable REGEXCOUNT allows for numbering in find and replace using regular expressions in source documents. (see below)

I. Setting codes conditionally:

Version 1.22 adds what I imagine could be a very useful metatag feature which allows the conditional setting of repeat values based on other repeat values (also works for universal values based on other universal values).

Imagine you have an interview or even better a focus group with a bunch of people Bob, Sally, Fred, and Ingrid. You might have a section of your transcript that looks like this

```
{speaker}bob{/speaker}
this is a test
{!end}

{speaker}sally{/speaker}
can't do it. no way
{!end}

{speaker}ingrid{/speaker}
Go away!
{!end}

{speaker}fred{/speaker}
I'd do it.
{!end}
```

At some point it might be nice to do a gender analysis. Conditional code setting allows you to enter sex/gender data without a lot of fuss. In the init file (or at the head of the file) you need to declare a new variable, “sex,” as a repeat variable and then set its conditions for assignment:

```
{!repeat sex}
{!if speaker=”bob”=>sex = “male”}
{!if speaker=”sally”=>sex=”female”}
{!if speaker = “ingrid” => sex =”intersex”}
etc.
```

Now TA will monitor the value of the repeat variable speaker and when it hits an {!end} the program will assign the value of "sex" based on speaker's value. The "=>" (equal followed by greater than) symbol means "implies" so the first !if could be read as "if the speaker is bob that implies sex is male." Now without recoding anything you have assigned gender data to your dataset and can do a sex/gender analysis.

To set universal variables conditionally create a dummy universal variable and then add the conditions:

```
{!universal region=""}
{!if school="NCSU"=>region="East Coast"}
```

I suspect that setting universals is much less of an issue than repeats.

Note that TA will be fussy "bob" is not the same as "bob " (with a space afterwards) so be careful coding!

II. REGEXCOUNT

It's nice to be able to reference parts of a document. In some qual analysis programs this is done by fixing the text and using line numbers. I do not see doing this in the foreseeable future. Cocoa's nstext and nstextview around which TA is built has no easy facility for this. On the other hand, you can implement an alternative by numbering sections of your document and then using those numbers as reference points: turn numbers in interviews, for instance. Obviously this is not as fine grained as line numbers, but it's better than nothing, and I don't see an alternative given Cocoa as well as TA's "in the text" coding scheme—which makes stable line numbers unlikely.

To facilitate this idea of numbering parts, I've modified the find/replace dialogue box handling so that you can insert the number of the replace in the replace string. An example will illustrate the technique. You have an interview, it's separated into turns with an {!end} at the end of each person's turn and a {speaker}Name{/speaker} at the start. You could number each of these turns using the regex option in the find box. Find "{speaker" and Replace "{!universal turn = "REGEXCOUNT"}\n$0" The word REGEXCOUNT will be replaced by 1, 2, 3… for each "{speaker" found as TA goes through finding and replacing. So

```
{speaker}Bob{/speaker}
```

will become (if bob's the third person to talk, for instance}

```
{!universal turn = "3"}
{speaker}Bob{/speaker}
```

Now you have a reference for any passages you find. You can refer other researchers to turn #3.

Note: An alternative would be to replace using "{turn}REGEXCOUNT{/turn}" and declaring turn a repeat code, one advantage of using universals is that you can remove all of the turns and get back to your original text quickly with the delete all tags command.

Like I said, it's not line numbers, but it will help add reference points through a text with a minimum of recoding.