Textflo
Text Analysis Description
Version 1.6

[User Guide]

Kieran Greer,
Email: help@distributedcomputingsystems.co.uk.
http://distributedcomputingsystems.co.uk/textfilter.html

Table of Contents

1 Introduction

This document gives a more technical description of the text analysis options that are available for the Textflo application. The descriptions have been moved here to reduce the size of the original user guide document and it is possible simply to use the results of the analysis without knowing exactly how they work. There are two main types of analysis that can be performed. Word lists can be ordered or sorted by some pre-defined ordering, or they can be clustered based on popular similarities in them.

2 Analysis

This section gives a more technical description of the available text analysis options, including exactly what they try to work out. They can be used to give information about the similarities of the content of each text document.

2.1 What Analysis is Carried Out?

The analysis essentially performs some statistical operations on word sequences in the text, including individual words. The analysis options include the standard line, word and character counts that you would find in a Word Processor. For this application, some words can be removed as part of the text pre-processing and so word counts can be altered. In the `Analysis` panel, you need to add the `Word Counts` option to the `Analysis options` list to include these default counts. In addition to this, there are then a number of different algorithms that can be applied to analyse the text further. Two different types of analysis can be carried out. The first is to analyse raw text files, to produce individual term popularity counts. The second is to compare the analysis files themselves and return a percentage score of how similar they are. This similarity percentage would be split equally between each analysis feature that is considered. If there is a word count and a sequence count analysis for example, each would be assigned 50% of the final score.

2.2 Analysis Options

The analysis options that are available depend on what analysis type has been selected. They are declared beside the `Analysis Types`, or algorithms, where you select from the combo box and then `Add` the option to the list. You can then select an option on the list and click the `Remove` button to remove it again. There are also a set of check boxes, as well as the different algorithms types, that will pre-format the text and change how the analysis is performed. A summary of the check box options is as follows:

1. **Formatted / Filtered:** If this is selected, then only the text in the main GUI window will be analysed. You can therefore filter or change the text content first, before analysing it. You cannot add this changed text to a file list however and so to analyse with other texts,

you would need to save it first and then list the file path instead. If this option is not selected, then the list of file paths are read and their texts stored for analysing.

2. **Analyse together:** If this is selected, all of the text files are read and then analysed as a single document. There is then only a single analysis result.
3. **Analyse separately:** If this is selected, it forces the file list to be analysed as separate documents. Because there are then a number of analysis results, they are saved to related files instead. The saved analysis file is assigned the name of the original file plus an `.anls` file type extension. So without this option all files are treated as a single group, but with this option, each file is analysed separately.
4. **Filter first:** If this is selected, a saved filter procedure is used to process every text file first, before the other options are applied. The stored filter procedure can be browsed for and added to the `Stored filter` text area. Stored filter procedure files have a `.fpr` file extension. So, both the file and this check box option need to be specified. Then, each file in the list is read and processed by the filter procedure first. The resulting text is then further processed by the analysis options before the analysis is applied.
5. **With letters:** If this box is selected, then each word that is considered must contain at least one letter. So numbers only would not be considered.
6. **Exclude words:** If this box is selected, then the common word list is removed from the text first, before analysing the remaining words. For example:

   `The cat sat on the mat.`
   With common words excluded would become:
   `Cat sat mat.`

7. **Exclude XML:** If this box is selected, then the XML tags are removed from any XML document, before the remaining text is analysed. Note however that this also reformats the text into a single line. Filtering from the manual panel can re-format in other ways.
8. **Word stem:** If this option is selected, then word stemming is applied, to try to group the same word with different endings together. For example, ‘word’ and ‘words’ would be considered to be the same. Note that this only applies to English language words at the moment.

2.3 Analysis Algorithms

The following analysis algorithms are currently supported. They are based mainly on word or term frequency and there is an option to search for a specific term only. A different analysis might produce a different type of information and care must be taken when removing text during pre-processing, if exact line numbers are being returned. Some of the algorithms also now have a ‘Suggestions’ section at the start. If there is a particular result that is repeated, it is put into a suggestions section for your attention. Once you have selected the analysis options, you click the `Run Analysis` button to generate the output and then the `Save Analysis` button the save the analysis to a file. The analysis now automatically treats all

words as lower case and so 'THE' and 'the' are considered to be the same word, for example.

2.3.1 Linear Count Analysis

With this algorithm, two options currently exist, that count the most popular words and word sequences. These are calculated as follows:

1. Popular Word Counts

The most popular word count reads every word, after pre-processing, and counts the number of times that it occurs. The specified top number of words is then saved to the analysis model. Word stemming can also be applied here. The count total is also scaled by the number of documents involved, so that it is more compatible with analyses of single documents.

2. Popular Word Sequences

This option looks for popular word sequences in the text. You can specify the minimum and maximum length of word sequence to look for and also the number of each sequence to save. For example, if you specify:

- Number in sequence From = 2.
- Number in Sequence To = 5.
- Number to Output = 3.

The operation will look for the top 3 counts for two, three, four and five words, in sequence together. Note that the output will only show the top sequences as they are stored. If, for example, you ask for the top 3 sequences of 2 words and there are 5 sequences that all occur 10 times, then only the first 3 stored will be output.

Note: The analysis will currently only add a sequence if the frequency count is larger than 1. It will also include any sequence that contains a smaller one, without incrementing the max sequences count, so the max sequences count value relates to new sequences, where the output list can be larger and have sequence parts that are repeated. This is just to add some more variability.

2.3.1.1 Search Using the Analysis Term

If you have a very large file, then sorting it could take some time, which is why the numbers of results to search for and keep can be changed. You can also select to look for a specific word or term, through the Analysis term field. The search will then only consider word combinations that include the term and may reduce the processing times.

2.3.1.2 Compare with First Analysis Only

Another small option is the `CWF` check box that tells the analyser to compare analysis files with the first analysis only. So the first document in the file list is compared to by all of the other documents. You can also find this result in the output of a full comparison.

2.3.1.3 Analysis Suggestions

This algorithm has a suggestions section. If there is a word sequence that is repeated more than once, or maybe has a higher count, then the most popular group of these will be listed at the top of the analysis results. This would be common, where a smaller word sequence can easily be included in one or two larger ones.

2.3.2 Line Clustering

With this algorithm, you can ask the analyser to cluster lines, or indicate which lines contain the same text sequences. This option firstly calculates the popular word sequences using the linear count analysis of section 2.3.1. It calculates what lines contain the popular word sequences and then lists them, together with the word sequence. There are two options here:

- Cluster lines based on an exact word sequence. This means that each word must follow the other one exactly for a match.
- Cluster lines based on words that exist in the same order. This means that the words must occur in the correct order, but anywhere on the line. In practice, this might be very similar to the first option.

These lines can then even be fed back into the system through a query, to highlight them on the original document. This can be seen in Figure 1, where the query form has actually completed executing the query. When the query form `Highlight` option is selected, the second combo box will display all of the popular word sequences. If one of these is selected, then it is added as the query that is displayed, along with the related line numbers. For this result the following steps were performed:

1. The ‘menu.xml’ file is loaded into the application.
2. In the `Analysis` tab, the ‘Line Cluster’ analysis type is selected.
3. The ‘Sequential Word Sequences’ option is added, with the default analysis model configuration.
4. The analysis is run (`Run Analysis`), and the result is the line clustering.
5. The `Query Form` is opened.
6. With the `And-Or` query tab, the `Value` is set to ‘Highlight’ and the text sequence to ‘belgian waffles’.
7. The query form `Add` button is clicked, when the line numbers should be displayed.
8. The query form `Execute` button is clicked, when the selected lines are highlighted in the main text, as shown in Figure 1.

Note: The analysis starts with popular word sequences, created as in section 2.3.1 and so the condition of that apply here as well.

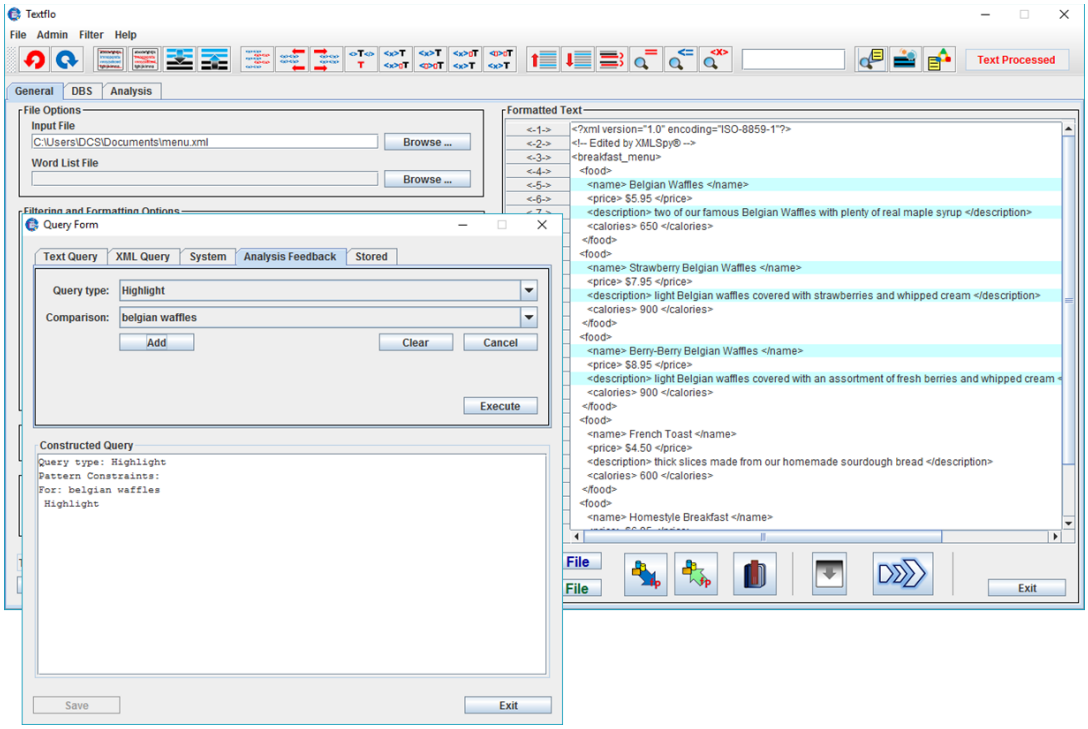

Figure 1. Highlight query has highlighted the related lines on the main document.

This option will now highlight the correct lines in almost every case, even when content is removed as part of the clustering analysis. If XML element tags are removed (`Analysis` panel – `Exclude XML` check box option) however, this could upset the line numbering and might cause incorrect lines to be listed. It would be better to remove the XML tags first and then perform the analysis over the re-formatted text, or spaces can be added between the tags and the test to identify all of the separate words. Another danger might be blank lines at the very beginning of the text. It is relatively easy to check the highlighting however, by scrolling through the document and checking a few of the highlighted lines..

2.3.2.1 Analysis Suggestions

This algorithm has a suggestions section. Any line numbers from any result that occur more frequently are listed in a single suggestions category first. They can therefore represent any sequence of words that might commonly occur.

2.3.3 Cluster Algorithm

This can be used to compare analyses only. It allows for the selection of a number of popular metrics to compare the similarity/difference between pairs of analyses:

- The Euclidean function is a direct measure of how close the frequencies are for all of the terms. Similarity function is a simple count of number of terms that are the same. It does not consider word frequency.
- Cosine similarity and Jaccard coefficient measure the same sort of thing. It is a set similarity that does not consider exact placement.
- The CF Inverse Doc Freq can also be used for comparisons and is also the basis for the other metrics [1]. It creates the word list that they compare.

The CF Inverse Doc Freq [2] performs primarily a popular word count, based on the inverse document frequency that considers both popularity and rarity of words across a group of documents. If only one document is analysed, this can produce a similar score to the linear word count. The algorithm looks for the most popular words in a document, but also considers if it is popular in other documents as well. It looks for the most distinguishing features in the document, which means that it should not be popular in every document, but preferably in one document only. A text about ‘computer hardware’, grouped with texts about ‘computer software’, for example, should rank a word like ‘CPU’ highly, because the software texts would not include it. Common words might still rank highly in any text, because they are so numerous, and so they can be filtered out first using the pre-processing options. This type does not use a specific analysis term, as it only considers single words for its clustering.

2.3.3.1 Analysis Suggestions

This algorithm considers all of the best matches over all of the selected metrics and lists the most popular ones with a count of number of occurrences. It is a little bit like a hyper-heuristic, as it combines results from different metrics.

Note: The different metrics can produce different scores, sometimes just in terms of magnitude, so it would be a matter of using the ones that are most appropriate to you. It might also be the case that using a single clustering type is better than combining the results of more than one, especially if they produce different clusters. So do not assume that adding more options will produce better results.

2.3.4 Organiser

This option refers to category groups in the Organiser only. You browse and select a number of groups, where sub-groups are also included. You can then run one of two algorithms to compare the groups to see how well they match with each other. This is based on

comparisons of their keyword lists. The only configuration option is the popular words number field, used here to define the maximum number of comparisons to output in the analysis results.

2.3.4.1 Organiser Keywords Entropy

This is a new algorithm that has been created as part of a research project. It is not the typical entropy measure but a very quick measure of group coherence or similarity. It is used here to look at the Organiser groups, to check if they are consistent with each other. It gives an overall coherence values for all keyword lists. It then suggests how the value would improve if you removed some of the groups. The value is presented as a percentage.

2.3.4.2 Organiser Keywords Count

This is a more standard count of similar keywords in each group, where the top number of comparisons are listed. In this case, comparisons between two groups only are listed, but it can be traced through to see what groups would link together.

2.4 Analysis of Individual Files or File Groups

The files to be analysed are typically read from the file list in the `Analysis` tab. The only exception to this is if there is an existing filtered or formatted document and the 'Formatted / Filtered' check box is selected. In that case, the analysis applies to the filtered text instead. Some of the analysis algorithms can analyse more than one file at a time. This would result in a score that comes from all texts being analysed together. In fact, the only problem here should be the `Line Clustering` algorithm, because it is required to return exact line numbers and so this needs to apply to one document only. If several documents are combined, the resulting score is scaled by the number of documents involved. This should make the analysis of several files together more compatible for comparison with the analysis of a single file.

If a list of files are specified, if any are recognised as special files (analysis or group) then they are removed before the analysis process starts. Only raw text files can be analysed this way. This type does not use a specific analysis term, as it only considers single words for its clustering.

2.5 Comparison of File Analyses

When the analysis is generated, you can save it in XML format to a file, where you can generate several analyses and save them to separate files. You can then compare the different

analyses to calculate how similar they are, by using the `Compare Analyses` option. The file list is then read and processed as follows:

- If you reference existing analysis files, they are read as is. Note that the analysis type for the comparison is taken to be the one currently selected in the `Analysis Type` combo box and so the analysis files should also be of that type.
- You can also reference raw text files. In that case, they will be converted into analysis models first, based on the selected algorithm and options, before being compared to the referenced analysis files. Each raw text file will produce a new analysis model.
- You can also reference file lists stored in your organiser. The `Category Selection` area of the panel allows you to browse through your saved organiser categories and to add a group of 3 categories. If this is added to the analysis process, the file list relating to the category group are retrieved and analysed as a single group. This produces a single analysis model that is then compared with the other ones. This is a useful way to help to determine what category or group a new file might belong to, as part of a clustering process.

The comparison analysis now produces a comparison for every analysis file against every other one. If there are three files – A, B and C, for example, the analysis produces a comparison of A with B and C, B with A and C, and C with A and B.

2.5.1 Linear Word Count Comparisons

This comparison produces a score based on a points (counting) system, over the popular words and word sequences. If two different analysis results have the same word at any point in their files, then the similarity measure is awarded the maximum score. If they have the same word but at different positions, then the size of the difference is also measured. So, for example, if the popular words for two different analyses are:

Analysis 1	**Analysis 2**
computer	computer
service	client
word	word
file	filter
text	text
filter	service

Then the words `computer`, `word` and `text` would add the maximum score to the total; the words `service` and `filter` adding smaller scores, with ‘filter’ being slightly more similar than ‘service’. The differences in the index positions are used to determine the reduction in the score value. The score is measured by adding up the differences and comparing this to a

possible worst value. Each analysis score is scaled in the same way, which makes comparing analyses possible. For the index differences, the possible worst score is taken to be:

*Possible worst score = (number of terms + 1) * 2*

This tries to take account of the fact that if terms are at positions 1 and 5 already, for example, then other ones cannot also occupy those positions. A total for the differences is then calculated. The same term at positions 1 and 5, for example, produces a difference value of 4. The final difference score is then calculated as:

Final term difference = possible worst score / (possible worst score + term difference)

For bag-of-words structures, this is then weighted further by comparing the exact word or term count values. If two sets of words are the same, but the first has counts of 100, 80 and 50; while the second has counts of 100, 70, 30; then this will not score 100% for a Comparison. This is calculated as follows: For each term that is the same, the counts are retrieved. They are both added to a total count value. The difference between them is also calculated, with the absolute value added to a difference score. They are also scaled as follows:

Final count difference = total count value / (total count value + difference count value)

The term difference and count difference should therefore be in the range of 0 to 1, with 1 being the best value. A count is also made simply of the number of terms that are the same in any order and the number that are different. For the above example, the count for the same term is 5 and for different terms it is 1. This is also factored into the comparison equation to give a final equation of:

*Comparison % = (term same / (term same + term different)) * 100 * term index difference * term count difference*

Where the scaling factors of term position or count differences are not always present. These are the main equations that are used for the analysis comparisons, but they can produce different scores depending on how the popularity counts are measured. Therefore, if comparing analyses, do not just consider the actual percentage value, which could be small, but also consider the maximum value simply because it is the closest comparison. Note that the category clustering uses the same linear comparison to compare its term frequencies.

2.5.2 Linear Word Sequences Comparisons

For the word sequences it is slightly different. In that case, the match must be the same word in exactly the same position in any sequence, as the word order is semantically important. So, for example, the two sequences of length 6:

Analysis 1: The cat sat on the mat.
Analysis 2: A cat ran over the mat.

Would be awarded the score of 3 points, based on the words cat, the and mat. The total score can then be made into a percentage of the total possible if the two files are an exact match. So for example, if all words being an exact match gives a total of 10 and the actual match total is 8, the percentage match is (8 * 100 / 10) = 80%. If you specify several analysis files to compare, the result produces a comparison of each file with every other one, in the form of a percentage match between each set of two files. The output can then be saved to a file in XML format.

2.5.3 Document Clustering Comparison

Analyses can also be compared with each other. This can be for single files or file groups that would be saved in the organiser. Two different counts are calculated. One count calculates exact matches between two sets of analyses. The other count measures a more general match without exact positions. For the line analyses, these values would probably be different. If using the clustering algorithms, the metrics have no real order and so the values should be the same. The text of Figure 2 is an example of the sort of analysis that is produced. This is for a single group of documents stored in the organiser, where the documents themselves are compared with each other. The group categories were selected and the files themselves loaded. Stop words were removed and then word stemming, with a clustering algorithm. The suggestions show at least 5 clusters inside of the group itself. Note that there can be other comparisons with positive values, but the suggestions try to display the top *x* set of values only.

COMPARISON ANALYSIS FOR: Analysis1

Suggestions:
arcs2005.pdf
    Dressler05.pdf

merkle00ant.pdf
    p931-jansen.pdf

EJOR-ANt-permu-flow.pdf
    stigmergyAnts.pdf

Dooley97.pdf
    chiang05.pdf
    LiuEtAlCoupledComplexity.pdf

15_ains02_behavior.pdf
    eco-sys05.pdf
    eco-sys04.pdf

15_ains02_behavior.pdf
    Dressler05.pdf: 0.16641453% : 0.16641453%
    chiang05.pdf: 0.12478336% : 0.12478336%
    LiuEtAlCoupledComplexity.pdf: 0.0% : 0.0%
    stigmergyAnts.pdf: 0.0% : 0.0%
    wac2004.pdf: 0.07540395% : 0.07540395%

Figure 2. Example of a document clustering comparison analysis.

3 Producing Sorted Lists

This category allows you to generate sorted lists of words based on certain ordering criteria. These options are available from the main GUI `General` tab, as part of the list of filtering options. It is possible to filter a text document all the way down to a single column of words. After you create the list of words, you might want to order it in some way. This sort of process could be useful for finding consistent structure in the text. You can, for example, only look at certain word sequences and check if they consistently occur together.

You can, by default, order it into ascending or descending order, as determined by the `String` comparison operator. There is also an option that allows you to order the list based on your own specified order. In that case, you add you own word order in the `Word Sort Order` group of components, where that will be the order used to reorder the words. The program also allows you to perform some level of analysis over the generated patterns, for example, produce a sorted list and then count popular word sequences through the analysis tab.

3.1 List Reorder Options

There are two different ways to order the words. The first way is more conventional, where all of the words are ordered based on a specified word sequence. In that case, all `car` words would come before all `van` words, for example. A second option however is intended to produce an ordering that would be useful for finding nested word or concept sequences. In that case, each reordered group only goes to the next instance of the first specified word in the sequence. For example, if the first word is car and the second word is van and the reorder spec is ascending; if there are three instances of each word, the reordering will produce, as shown in Table 1:

Reordering for Nested	Reordering for Conventional
Car	Car
Van	Car
Car	Car

Van Car Van	Van Van Van

Table 1. Example of different word ordering types.

If you specify your own ordering, then if this does not cover all of the available words, there are two options: The generated list of ordered words can be added to the end of the list, with any other not included words placed separately. Alternatively, if the 'only' options are selected, only the words specified in the word order list will be included. For the nested ordering, the words in each nested group that are not included in the ordering list are placed at the beginning for that nested group only, and not for the whole list, or left out completely.

Acknowledgements

See the user guide for the acknowledgements list.

References

[1] Huang, A. (2008). Similarity measures for text document clustering. Proceedings of NZCSRSC, pp. 49-56.

[2] Sahoo, N., Callan, J., Krishnan, R., Duncan, G. and Padman, R. (2006). Incremental Hierarchical Clustering of Text Documents, Published in: CIKM '06 Proceedings of the 15th ACM international conference on Information and knowledge management, ACM New York, NY, USA.

4 Appendix A - Default Analysis Configuration File

The analysis configuration file is written in XML format. The default file is loaded into the system at startup from the `files` folder and performs the currently available options of popular word or word sequence counts. Note that the config file now contains an entry describing exactly what analysis type it belongs to. This must match the type of analysis being carried out. The structure of the file is shown in Figure 3:

```
<Analysis_Model Analysis_Type="Linear Count">
   <Popular_Words_Number>10</Popular_Words_Number>
   <Min_Nesting_Number>2</Min_Nesting_Number>
   <Max_Nesting_Number>5</Max_Nesting_Number>
   <Sequence_Number>3</Sequence_Number>
</Analysis_Model>
```

Figure 3. Default Analysis Configuration File.

The following elements can be configured or changed in the file:

- Popular words number: this is the number of popular words to output. The default value of 10 means that the 10 most popular words will be output with their values. If you change this number then that will change the number that is output.
- Minimum nesting number: This is the smallest number of words in a sequence (consecutive) to measure.
- Maximum nesting number: This is the largest number of words in a sequence (consecutive) to measure.
- Sequence number: This is the number of popular sequences to output for each word sequence number.
- Separate on sentences: This has been removed as it affects the text format and related line indexing too much.

So for example, if the minimum number is 2 and the maximum number is 5 and the sequence number is 3, the analysis will output and store the top 3 sequences for 2, 3, 4 and 5 word sequences. It is easy to test or change this to see what it does. The configuration file is editable, so you can change it to whatever you wish and then load/save the new file.