Working Paper.
ENGLISH ONLY

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Paris, France, 28-30 April 2014)

Topic (i): Selective editing / macro editing

Text Analysis Tools for Editing and Verification

Prepared by Wendy L. Martinez,[1] Bureau of Labor Statistics, United States

I. Introduction

1. Data in unstructured and semi-structured text fields of survey records are typically underutilized in most processing procedures and analyses. This paper will highlight some approaches in text analysis that can be used to extract and to utilize information from these fields. An illustration of how these text analysis tools can be employed to assist humans in applying labels to accident reports from the U.S. Department of Labor, Occupational Safety and Health Administration (OSHA) is presented.[2] Other potential data sources include text descriptions of job requirements, benefit package documentation for workers, time-use patterns, nutritional intake summaries, and consumer expenditure habits; as well as paradata produced through interviewer descriptions of the contact and interview process.

2. There are many tasks and objectives one might want to accomplish when analysing documents. In this paper, we use the word *document* to refer to any portion of free-form or unstructured text. This could be a phrase in a survey record, sentence, paragraph, chapter, email, etc. [Martinez and Measure, 2013]. We will focus on one application of text analysis in this paper, which is the *supervised classification* of documents.

3. In supervised classification, we have a set of documents (or a *corpus*), where each document is coded with a label. This label can refer to any grouping of interest, e.g., the document topic, type of event for the incident, or the occupation title for a job description. The goal is to use the labelled corpus to construct a classifier, which can then be used to assign labels to unlabelled documents. Previous work using text analysis for classification include the computer assisted coding of records in the Survey of Occupational Injuries and Illnesses (SOII) [Measure, 2013].

4. This working paper discusses the basic goals for employing text analysis in data editing and verification. For example, we could do the following:

(a) use the narratives in a report and their associated event type to build a classifier using methods in computational statistics;
(b) apply the classifier to the data; and
(c) conduct editing and verification tasks, such as:
    - examination of misclassified records for editing and consistency;
    - classification of miscoded records.

[1] The author is grateful for help received from several Bureau of Labor Statistics colleagues, including John Eltinge, Polly Phipps, and Stephen Pegula.
[2] This work reflects the views of the author and does not reflect official U.S. Department of Labor policy, procedures, or applications.

II. OSHA Enforcement Data

5. The data used to illustrate the process of exploiting unstructured text fields for editing and verification came from OSHA Fatality and Catastrophe Summaries, which are also known as Accident Investigation Summaries – OSHA 170 form. These forms are used after OSHA conducts an inspection following a catastrophe or fatality. The forms have many fields that together provide a complete description of the incident and can include event details and causal factors.

6. The main websites to access the data and other relevant information are given here (accessed February, 2014):

(a) Catalog of data sources: http://ogesdw.dol.gov/views/data_catalogs.php;
(b) OSHA data download: http://ogesdw.dol.gov/views/data_summary.php
(c) Data dictionary: http://enforcedata.dol.gov/views/data_dictionary.php

III. Text Analysis

7. Some background information on basic steps in text analysis is given in this section, and we include some of the common approaches in the literature for each of the steps [Martinez and Measure, 2013]. It should be noted that this paper outlines initial work in applying text analysis methods to the editing process, and it is meant to generate ideas, research, and potential applications. There are other options and methods for text analysis than what we used for our example, and some of these are mentioned for additional background.

A. Pre-Processing the Text

8. The first step in the analysis of text data is to clean it up by removing any special characters, such as %, $, #, @, *, ~, colons, semi-colons, etc. It is also a good idea to convert to lower case and to remove *stop words*. These are words that do not enable us to discriminate between documents in terms of their meaning. Stop words are typically common words, such as *the, or, him, it*, etc. However, they can also be task and domain specific [Martinez and Measure, 2013]. As an example, the word *employee* has low utility for classification or discrimination, if it appears in every document.

9. *Stemming* is another step that is sometimes used. Stemming reduces the number of unique words in the *lexicon* (the list of unique words in the corpus) by removing suffixes and prefixes, leaving us with the stem or root of the word. For example, the words *protecting, protected, protects* would be reduced to the word *protect* [Solka, 2008; Porter 1980]. Stemming was not used in the preliminary analysis described in this paper.

B. Encoding the Text

10. The text has to be changed to numbers, in order to apply methods in computational statistics. A common approach to encoding text is to use the *term-document matrix* (TDM). This matrix has dimensions $n \times p$, where p is the number of words (or terms) in the lexicon, and n is the number of documents. Using our typical statistical terminology, the number of variables is given by the number of words, and the observations correspond to the documents. The use of a TDM in text analysis is sometimes called the *bag-of-words* approach, because the relative positions of the words in the documents are lost.

11. The elements of the TDM contain some indicator of the occurrence of each word in the document. For example, the i,j-th element could have a 1 if the i-th word appears in the j-th document or a 0, if it does not. We could also use the raw frequencies, denoted by $tf_{i,j}$. This represents the number of times the word appears in the document. Researchers in the information retrieval community developed term weights to incorporate other information about the words. A popular term weight is the *inverse document frequency*, which is a function of the number of documents in the corpus that contain the word.

Thus, it accounts for how common the word is in the corpus. It will down-weight words that appear with high frequency and will up-weight words that occur rarely [Berry and Browne, 1999; Berry, 2001].

C. Removing Noise – Latent Semantic Indexing

12. We usually have a set of very high-dimensional data when we use the TDM as our encoding method, and it is an example of a small *n*, large *p* problem, where we have many more variables than observations. In order to apply some statistical methods (e.g., creating a naïve Bayes' classifier), we have to reduce the dimensionality. It is also a good idea to reduce the number of dimensions because the TDM is very sparse and noisy; i.e., has many variables with values of zero.

13. A common approach to dimension reduction in text analysis is called *latent semantic indexing* (or *analysis*) – LSI/LSA. This is based on the singular value decomposition of the TDM [Deerwester, et al., 1990; Landauer, et al., 1998]. The singular value decomposition is a well-known method in linear algebra, and it seeks a factorization of a rectangular matrix, which in our case is the TDM.

14. It enables us to write the term-document matrix as a product of three factors:

$$X = TSD^T$$

where X is the TDM, T is a matrix containing the left singular vectors; S is a diagonal matrix of singular values; and D^T is the transposed matrix of right singular vectors. By convention, the singular values are ordered in decreasing magnitude. It is interesting to note that the left singular vectors span the document space, and the right singular vectors span the term or word space [Solka, 2008].

15. We can reduce the dimensionality (and the noise) by keeping the k largest singular values and their corresponding singular vectors to get a lower-rank approximation to the TDM:

$$\tilde{X} = T_k S_k D_k^T$$

Note that reducing the dimensionality using the singular value decomposition is similar to principal component analysis for square matrices, and many of the concepts are similar. In particular, we can project onto the word space, as follows:

$$X_m = X^T D_m$$

where X^T is the transpose of the TDM, and the columns of D_m correspond to the singular vectors corresponding to the m largest singular values. The matrix X_m will have n rows and m columns, where $m < p$.

16. Another approach to dimension reduction that is often used for text data is *nonnegative matrix factorization* – NNMF [Lee and Seung, 1999]. NNMF seeks a factorization of the TDM, which is similar to LSA and principal component analysis. However, NNMF seeks factors that keep the features nonnegative. This produces a factorization that makes more sense in the context of the TDM, because all of the original entries in the TDM are nonnegative.

D. Document Classification

17. Now that the text data have been cleaned and converted to numbers, we are ready to analyse the documents. Two main methods in text analysis are *supervised document classification* and *unsupervised classification* (or *clustering*). We describe the first task here and provide an illustration of how this can be used for editing and verification in the next section. We also briefly mention how document clustering can be used to explore documents for editing in the final section of this working paper.

18. Almost any type of statistical pattern recognition method can be applied to text data after it has been encoded and not all require a reduction in dimensionality. Some common methods are:

(a) Naïve Bayes;
(b) *K* nearest neighbours (*k*-NN);
(c) Classification trees.

19. *Classification trees* are also known as *decision trees*. A tree represents a multi-stage decision process, where a binary decision is made at every stage or node in the tree. A decision tree is used to predict a response, which would be the class label in supervised classification. See Figure 2 for an example of a classification tree.

20. To predict the response for a given observation, one would start from the root node and make the appropriate decision at each node, until the leaf node is reached. Each leaf has a label (or response) associated with it, which is then attached to the observation. Classification trees can produce complex decision boundaries and can be used with continuous and/or categorical data (i.e., predictors). See Duda, et al. [2001] and Webb [2002] for more information on classification trees and other pattern recognition approaches.

21. There are many unsupervised classification or clustering methods in the literature, and we list a few of them here [Everitt, et al., 2011]. With clustering, we usually do not have class labels for our records, or we chose not to use them.

(a) *K-means clustering* is used quite often. With this approach, we divide the data into *k* groups, such that the within-group sum-of-squares is minimized. Most algorithms for *k*-means use an iterative algorithm to solve the optimization problem. The value for *k* must be specified in advance.
(b) *Agglomerative clustering* provides partitions of the data for any (and all) number of clusters. The user can specify the desired number of groups, once the data are partitioned.
(c) Another clustering approach is *model-based clustering*, which uses an estimate of a probability density model to obtain the groups. The model that is used for the density function is a finite mixture. This is a weighted sum of probability density functions, where each term in the mixture represents a group or cluster. [Fraley and Raftery, 1998].

IV. Application to OSHA Reports

A. Data

22. As mentioned earlier, we use OSHA investigation summaries of work-place accidents in this paper to illustrate our ideas. We downloaded OSHA summary reports for November and December, 2011, yielding 358 documents or observations. There are several files with entries that correspond to one report; see the data dictionary on the OSHA website for an explanation of the fields and in what table they reside.

23. The records have several free-form text fields. A partial list is given below:

- Event description – a short text phrase that summarizes the event
- Event keywords – nature, body part, source, etc.
- Abstract – free-form text field describing the accident

24. The goal is to use the abstract of the accident and the associated event type to build a classifier, which will then be used for editing and verification. There are 14 different event type codes (see Table 1). After looking at the frequency distribution of events in November and December, 2011, we decided to do the following:

- Combine the two *falling* events – codes 04 were converted to 05;
- Combine the two *struck* events – codes 06 were converted to 01;
- Keep only codes 01, 02, 05, and 14 for the analysis.

25. We also had 5 records with event type codes 00, which are obviously incorrect. When processing the records based on their event codes, we also found that some investigations had more than one event type attached to it. To our knowledge, there should be only one investigation summary, regardless of the number of events in the accident. Thus, there should be one event type for the investigation, and these duplicate records need to be edited.

Table 1. Event Type Codes - OSHA 170 Form

01	Struck by	08	Inhalation
02	Caught in or between	09	Ingestion
03	Bite/Sting/Scratch	10	Absorption
04	Fall (same level)	11	Rep. motion/pressure
05	Fall (from elevation)	12	Card-vascular/breathing failure
06	Struck against	13	Shock
07	Rubbed/abraded	14	Other

26. We were left with 332 records after we combined and extracted the categories as described above. The frequency distribution of the records is given in Table 2. Note that there is some overlap between the documents in the groups that need to be edited; i.e. some records need editing for more than one reason.

Table 2. Summary of Categories used in Analysis

Category	Frequency		Category	Frequency
00[a]	5		05	106
01	100		14 [a]	27
02	78		Duplicates [a]	22

[a] Records in these categories need editing and verification

B. Processing the Text

27. We processed the abstracts or narratives in the 332 accident reports by removing all special characters and converting to lower case. We added four words to the stop word list, based on a previous analysis [Martinez and Measure, 2013] and removed them from the documents. These were *employee*, *approximately*, *November*, and *December*. This produced a lexicon with 3,665 words. Thus, we have $n = 332$ and $p = 3665$.

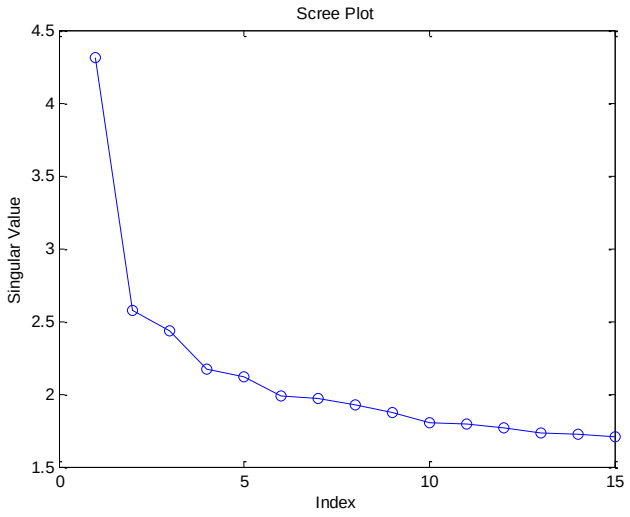

Figure 1. This is a scree plot showing the singular values on the vertical axis. We look for an elbow in the curve to estimate the number of dimensions to keep. We could pick 2, 4, or 6, and we chose $m = 4$ as a reasonable value.

28. We created a TDM using these data, where the entries represented the raw frequencies. As we will see in the next section, we used these data to build a classification tree. We could have used the original data with 3,665 variables; but it is usually good practice to reduce the amount of noise before processing. It also makes it easier to visualize the data and the results from the analysis.

29. We use the singular value decomposition approach as outlined in section III.c. A scree plot was used to help determine the number of dimensions m to keep [Jackson, 1991], looking for an elbow in the curve as a good value. The scree plot is shown in Figure 1, and a reasonable value seems to be $m = 4$.

C. Analysis and Results

30. After processing the abstracts and reducing the dimensionality, we are left with a data matrix **X** that has 332 rows and 4 columns. Our analysis will use supervised classification to edit the event type of records. In particular, we will build a classification tree and use it for the following editing tasks:

- Determine event type labels for the records that have code 00
- Propose a specific event type for records with code 14 (*Other*)
- Suggest an event type for the duplicate records
- Verify misclassified records

31. We removed 48 observations from our data matrix before constructing our classifier. These observations corresponded to the duplicates and the records with code 00 or 14. This means that our classifier will not have seen them during the training process. We used the remaining 284 observations to construct (or train) a classification tree.[3]

32. The resulting tree (after pruning) is shown in Figure 2. In general, decision trees are easily explained and understood, and they also provide some insight on the importance of the variables. The classification tree for our data shows that only two variables are needed for discrimination – the second and third. The resubstitution error for this classification tree is 0.1479, indicating a correct classification rate of approximately 0.85. We ended up with 42 misclassified observations that we will verify in our analysis.

33. Our first editing task is to use the tree to classify the records with codes 00. Looking at the event description field, it is apparent that only two of the records are likely to fall into one of the three categories – *Struck by* (01), *Caught* (02), or *Fall* (05). The classification tree suggested the labels shown in Table 3, which seem reasonable.

Table 3. Suggested Event Type from Classification Tree for Uncoded Records

Suggested Code	Event Description
01 – Struck by	Truck Tips over with Employee in It
05 – Fall	Employee Does Not Sustain Injuries in Fall from Ladder

34. It is often the case that coders will use the *Other* field rather than something more appropriate and specific to the event. Our next task is to use the tree classifier to predict a class label for some of the accidents coded as *Other* (14). The predicted class labels and the short event descriptions are given in Table 4. At first, these labels might seem inappropriate, but they are consistent with other accidents and their event type. For example, most incidents involving amputation and cuts are coded as 02 – *Caught by*, and this is what we have from our classifier. The third event in Table 4 involves a laceration and could be coded as 02, but it might be the case that the worker was *struck by* the grinder, resulting in the laceration. In any case, it is fairly obvious that these events should be coded as something besides than 14 (*Other*).

[3] The MATLAB® software environment was used for all computations in this analysis. Two additional toolboxes were also used – the Statistics Toolbox® (from The MathWorks, Inc.) and the Exploratory Data Analysis Toolbox (http://pi-sigma.info/).

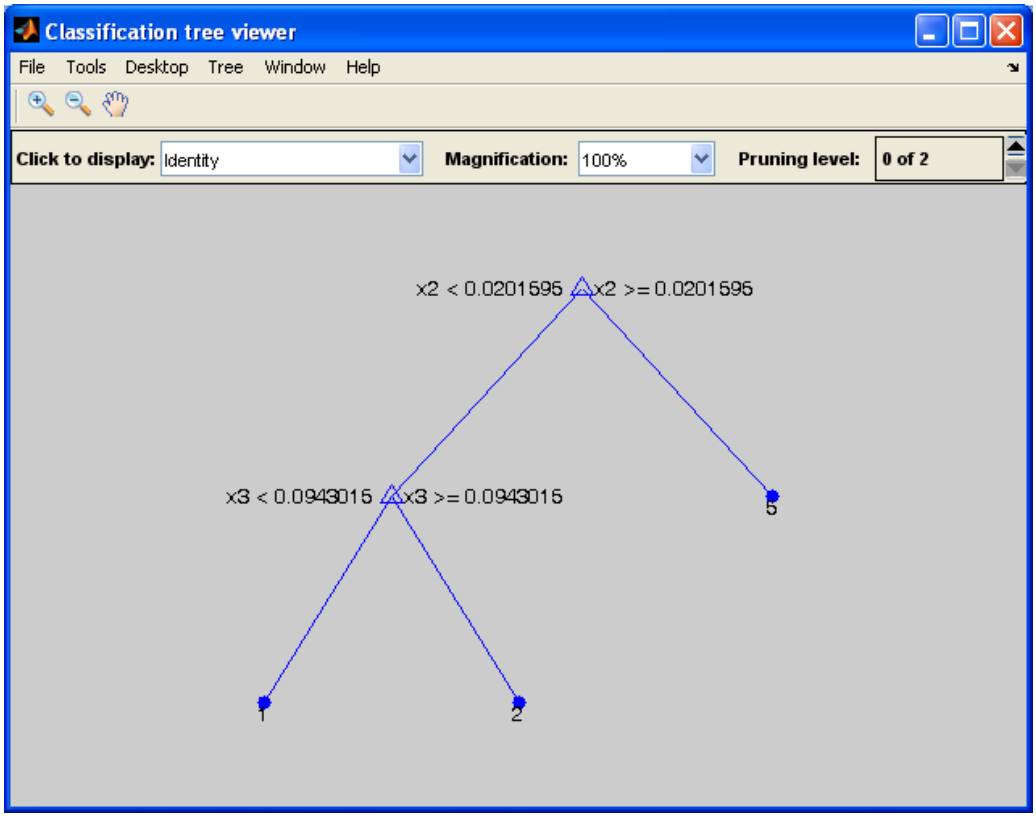

Figure 2. This is the classification tree after pruning. Note that it is a very simple tree in this case; it has only three leaf nodes – one for each class. Say we have an observation with predictor values 0.5, 0.01, 0.2, 0.3 **. To classify this observation, we start at the first node and travel down the left side. We take the right fork at the next node, which brings us to a leaf node with a class label of 2. This is the label (or response) we predict for our observation.**

Table 4. Predicted Event Type for Records Coded as *Other*

Suggested Code	Event Description
01 – Struck by	Employee Is Injured in Trash Truck Overturn
01 – Struck by	Worker Lacerates Hand on Angle Grinder Used on Concrete
02 – Caught	Employee Gets Finger Laceration by Door Clamp
02 – Caught	Employee Gets Finger Amputations with a Table Saw
02 – Caught	Employee's Finger Is Amputated by Miter Saw
05 – Fall	Water Tank Crushes Employee

35. It became apparent when processing the records that we had some investigation summaries with more than one event type. We used the classification tree to estimate an event type for 12 of these cases. The results are shown in Table 5, and we see that 8 of them seem to be given plausible codes (see shaded rows).

36. Finally, we looked at the 42 misclassified records in a similar manner – comparing the predicted and assigned event type codes with the short event description. The results of our assessment are listed briefly here:

- Thirty (30) of the records were truly misclassified by the decision tree.
- In our opinion, 6 records did not have a correct event type code assigned to them, and our tree classifier provided a more suitable event label. For example, the accident of "Roof framer falls from an unsecured beam and injures head" was officially coded as 01 (*Struck by*), and we classified it as 05 (*Fall*).
- Six (6) other records had event labels where either label seemed reasonable.

Table 5. Predicted Event Type for Records with Duplicate Event Types[a]

Suggested Code	Event Description
05 – Fall	Elevated Scissor Lift Collapses and Injures Two Workers
05 – Fall	Two Employees Are Killed When Aerial Lift Collapses
05 – Fall	Two Employees Are Injured When Elevated Platform Collapses
05 – Fall	Worker Fractures Leg During Wall Board Movement
05 – Fall	Employee Sustains Head Injuries in Fall Off Roof
05 – Fall	One Employee Is Killed and Four Injured When Floor Collapses
01 – Struck by	Two Employees Are Injured When Forklift Tip Over
01 – Struck by	Employee Is Killed When Crane Strikes Lift; Another Injured
05 – Fall	Employee Is Killed When Struck on Neck by Chain Saw
05 – Fall	Employee Dies from Crushing/drowning
01 – Struck by	Two Employees Killed in Fall from Basket Attached to Crane
01 – Struck by	Employee Is Injured When Hand Gets Caught in Equipment

[a] The predicted event types in the shaded rows seem plausible.

V. Conclusion

A. Summary

37. We provided a brief overview of some text analysis tools and showed how they can be used to edit and verify coded records by analysing the free-form text narratives in accident reports. The goal is to motivate our colleagues in the statistics community to exploit this information-rich data source. This is very preliminary work and is meant to spark ideas and hopefully the development of processes and tools based on the analysis of text.

38. As stated previously, we used MATLAB for all computations presented in this working paper. MATLAB is a computing environment that has extensive capabilities for handling text data (or strings), and it can be expanded through add-on toolboxes, such as the Statistics Toolbox. We also used some functionality in the Exploratory Data Analysis Toolbox that can be downloaded from http://pi-sigma.info/. This toolbox provides an interactive environment for dimension reduction, visualization, clustering, and more.

39. Another resource for conducting text analysis is the open-source computing environment R, which can be downloaded at the Comprehensive R Archive Network (CRAN) at this website: http://cran.us.r-project.org/. There is a Task View (look on the left of the page for a link) that contains information about R packages for text analysis and natural language processing: http://cran.us.r-project.org/web/views/NaturalLanguageProcessing.html.

B. Future Work

40. There are other methods and approaches in text analysis that one could apply to editing and verification based on free-form text fields. For example, we could try improving our ability to discriminate records based on their meaning by stemming the words as part of the pre-processing. Or, we might use term weights like the inverse document frequency to adjust the raw frequencies based on the occurrence of a word among the documents in the corpus.

41. We mentioned previously that the term-document matrix or the bag-of-words approach does not incorporate word order. Martinez [2002] developed the bigram proximity matrix, which does account for word order and has been shown to be an effective encoding for supervised and unsupervised document classification. It would be interesting to apply this encoding to the accident narratives and to assess the document classification results.

42. One could also cluster documents and verify the consistency of the codes by looking at the records that are grouped together based on their accident narratives. To illustrate this idea, we used

model-based clustering (see section III.c.) to group the 284 observations in our training set. A nice aspect of model-based clustering is that it will provide an estimate of the number of groups, and there is a strong indication of three groups in our data set. We show all pair-wise scatter plots of the first three dimensions (for ease of visualization) in Figure 3, where the observations are shown in different colors and symbols according to the cluster number. Note that there appears to be some definite group structure.

43. We can use the result of clustering to verify codes and to ensure their consistency across records. For example, are the reports consistent in how the investigators encode accidents involving amputations or lacerations? Table 6 shows the distribution of the event type codes in the three groups from model-based clustering. Clustering is an exploratory data analysis method [Martinez, et al., 2010], so one should employ different clustering methods and examine the results. Thus, we also applied *k*-means clustering requesting three groups. In that grouping, we found one cluster with 90% of the records being coded as 05 (*Fall*), and another where 53% of the group members were coded as 01 (*Struck by*).

Table 6. Distribution of Codes in Clusters from Model-Based Clustering

Cluster 1	Cluster 2	Cluster 3
01 – 52%	01 – 23%	01 – 41%
02 – 24%	02 – 9%	02 – 50%
05 – 24%	05 – 68%	05 – 9%

44. The OSHA investigation summaries analysed in this working paper came from a census. It would be interesting to apply these ideas to survey records that are obtained via a sample and to explore the impact of a complex sample design on the analysis. Examples of such data sources from surveys include:

- narratives on the most important political problems in the United States in the American National Election Survey (ANES): http://www.electionstudies.org/;
- job descriptions in the Survey of Occupational Injuries and Illnesses–SOII: http://www.bls.gov/respondents/iif/;
- extensive details about a complex mental health condition (e.g., schizophrenia) in medical interviews like the Diagnostic Interview Schedule for Children (DISC) http://www.cdc.gov/nchs/nhanes/limited_access/YDQ.htm.

45. This working paper provides an introduction to text analysis and describes some initial ideas on how it can be used in data editing and verification. Additional work and research needs to be done to make this viable in practice. In particular, interactive tools should be developed, once some methods and processes have been shown to be effective for applications.

VI. References

Berry, M. W. (ed). 2001. *Computational Information Retrieval*, SIAM.

Berry, M. and M. Browne. 1999. *Understanding Search Engines: Mathematical Modeling and Text Retrieval (Software, Environments, Tools)*, SIAM.

Deerwester, S. S. Dumais, G. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, **41**:391–407.

Duda, Rr. O., P. E. Hart, and D. G. Stork. 2001. *Pattern Classification*, 2nd Edition, New York: John Wiley & Sons.

Everitt, B., S. Landau, M. Leese, and D. Stahl. 2011. *Cluster Analysis*, John Wiley.

Fraley, C. and A. E. Raftery. 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis, *The Computer Journal*, **41**:578 – 588.

Jackson, J. E. 1991. *A User's Guide to Principal Components*, Wiley and Sons.

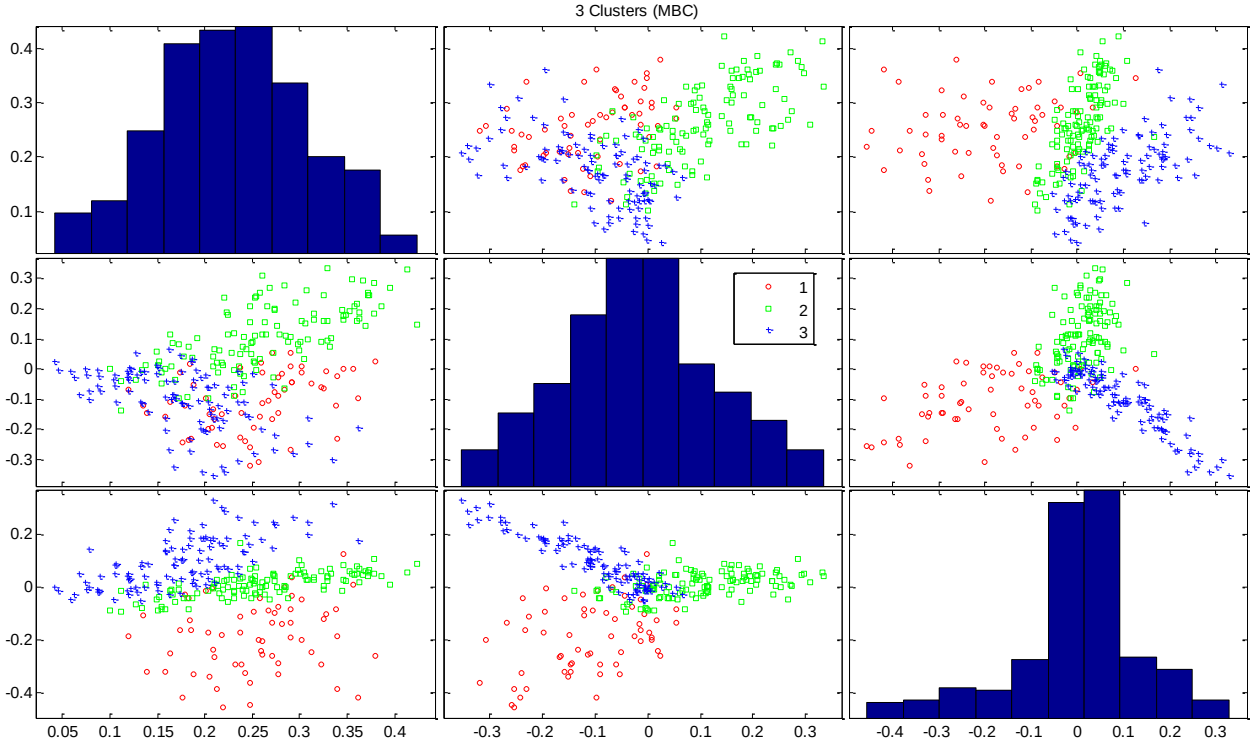

Figure 3. This is a matrix of pair-wise scatter plots of the observations in our training set, with the colors and symbol types corresponding to cluster IDs obtained from model-based clustering. We show the first three dimensions only for ease of visualization. Model-based clustering provides an estimate of the number of groups, which was three in this case. We see some reasonable and rather strong cluster structure in some of the plots, e.g., in the middle plot of the bottom row.

Landauer, T. K., P. W. Foltz, and D. Laham. 1998. Introduction to latent semantic analysis, *Discourse Processes*, **25**:259–284, http://lsa.colorado.edu/papers/dp1.LSAintro.pdf (accessed February, 2014).

Lee, D. D. and H. S. Seung. 1999. Learning the parts of objects by non-negative matrix factorization, *Nature*, **40**:788–791.

Martinez, A. 2002. *A framework for the representation of semantics*, Ph.D. dissertation, George Mason University.

Martinez, W. and A. Measure. 2013. Statistical analysis of text in survey records, *Federal Committee on Statistical Methodology* (FCSM – 2013), http://www.fcsm.gov/events/prior.html (accessed February, 2014).

Martinez, W. L., A. R. Martinez, and J. L. Solka. 2010. *Exploratory Data Analysis with MATLAB*, 2nd Edition, CRC Press.

Measure, A. 2013. Artificial Intelligence in Data Processing. *Presentation at the 2013 FedCASIC*. https://fedcasic.dsd.census.gov/fc2013/ppt/DM%20Fedcasic%202013%20Alex%20Measure%20AI%20in%20Data%20Processing%20FedCASIC%20version%202.pdf (accessed February, 2014).

Porter, M. F. 1980. Algorithm for suffix stripping, *Program*, 130–137.

Solka, J. L. 2008. Text data mining: Theory and methods, *Statistics Surveys*, **2**:94–112. http://projecteuclid.org/euclid.ssu/1216238228 (accessed February, 2014).

Webb, A. 2002. *Statistical Pattern Recognition*, 2nd Edition, Oxford: Oxford University Press.