Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2423033 A1 2002/03/28

(21) 2 423 033

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2001/09/25
(87) Date publication PCT/PCT Publication Date: 2002/03/28
(85) Entrée phase nationale/National Entry: 2003/03/24
(86) N° demande PCT/PCT Application No.: AU 2001/001198
(87) N° publication PCT/PCT Publication No.: 2002/025479
(30) Priorité/Priority: 2000/09/25 (PR 0338) AU

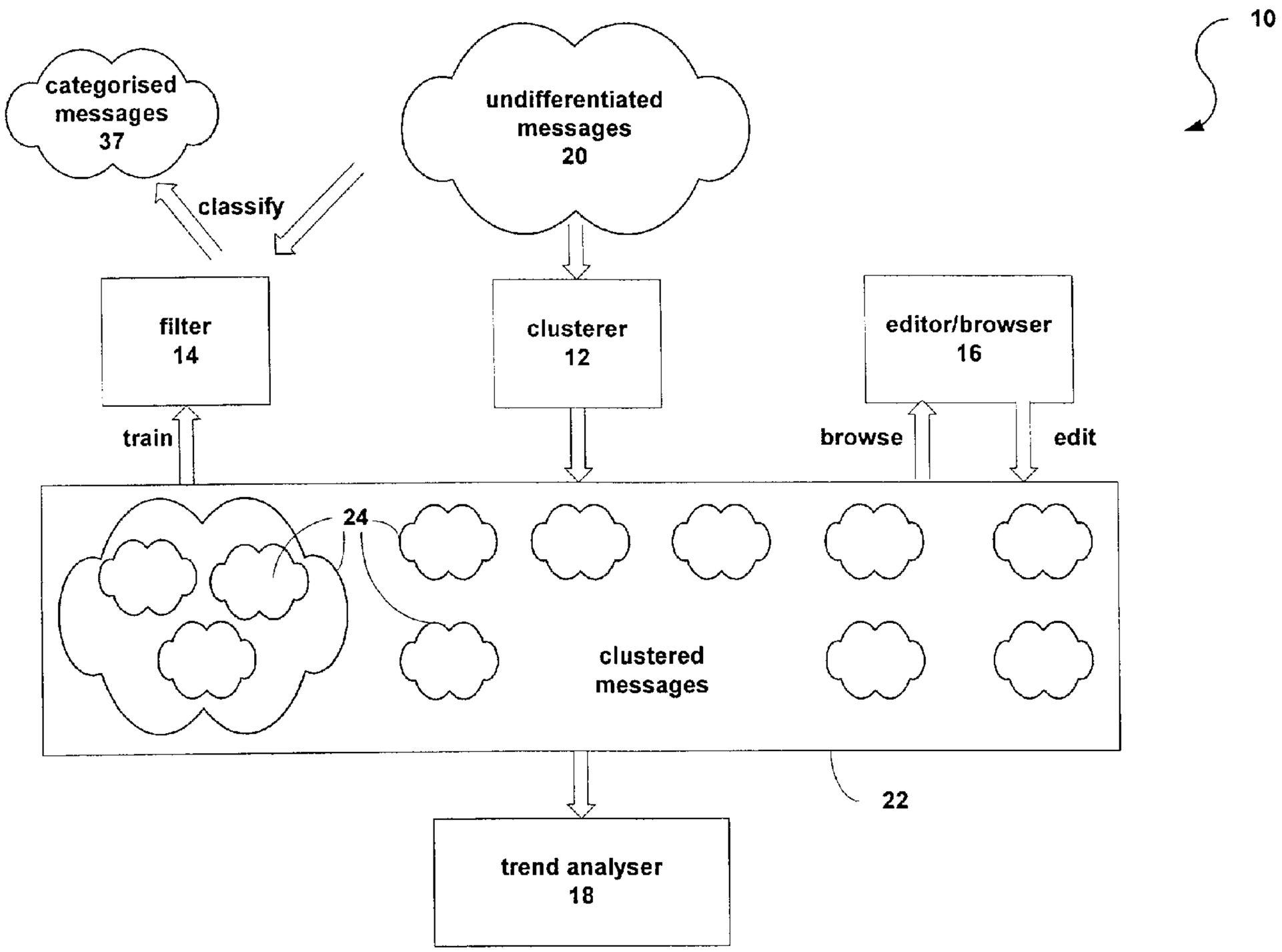
(51) Cl.Int.^{7}/Int.Cl.^{7} G06F 17/30

(71) Demandeur/Applicant:
TELSTRA NEW WAVE PTY LTD, AU

(72) Inventeurs/Inventors:
RASKUTTI, BHAVANI, AU;
KOWALCZYK, ADAM, AU

(74) Agent: SMART & BIGGAR

(54) Titre : SYSTEME DE CATEGORISATION DE DOCUMENTS
(54) Title: A DOCUMENT CATEGORISATION SYSTEM



(57) Abrégé/Abstract:
A document categorisation system, including a clusterer for generating clusters of related electronic documents based on features extracted from said documents, and a filter module for generating a filter on the basis of said clusters to categorise

CA 2423033 A1 2002/03/28

(21) 2 423 033

(13) A1

(57) Abrégé(suite)/Abstract(continued):
further documents received by said system. The system may include an editor for manually browsing and modifying the clusters. The categorisation of the documents is based on n-grams, which are used to determine significant features of the documents. The system includes a trend analyzer for determining trends of changing document categories over time, and for identifying novel clusters. The system may be implemented as a plug-in module for a spreadsheet application, providing a convenient means for one-off or ongoing analysis of text entries in a worksheet.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 March 2002 (28.03.2002)

PCT

(10) International Publication Number
WO 02/25479 A1

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/AU01/01198

(22) International Filing Date:
25 September 2001 (25.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PR 0338 25 September 2000 (25.09.2000) AU

(71) Applicant *(for all designated States except US)*: **TELSTRA NEW WAVE PTY LTD** [AU/AU]; ACN 070 562 935, 242 Exhibition Street, MELBOURNE, Victoria 3000 (AU).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **RASKUTTI, Bhavani** [AU/AU]; 4 Empress Road, SURREY HILLS, Victoria 3127 (AU). **KOWALCZYK, Adam** [AU/AU]; 8 Cappella Court, GLEN WAVERLEY, Victoria 3150 (AU).

(74) Agent: **DAVIES COLLISON CAVE**; WEBBER, David, Brian, PRYOR, Geoffrey, Charles, LESLIE, Keith, 1 Little Collins Street, MELBOURNE, Victoria 3000 (AU).

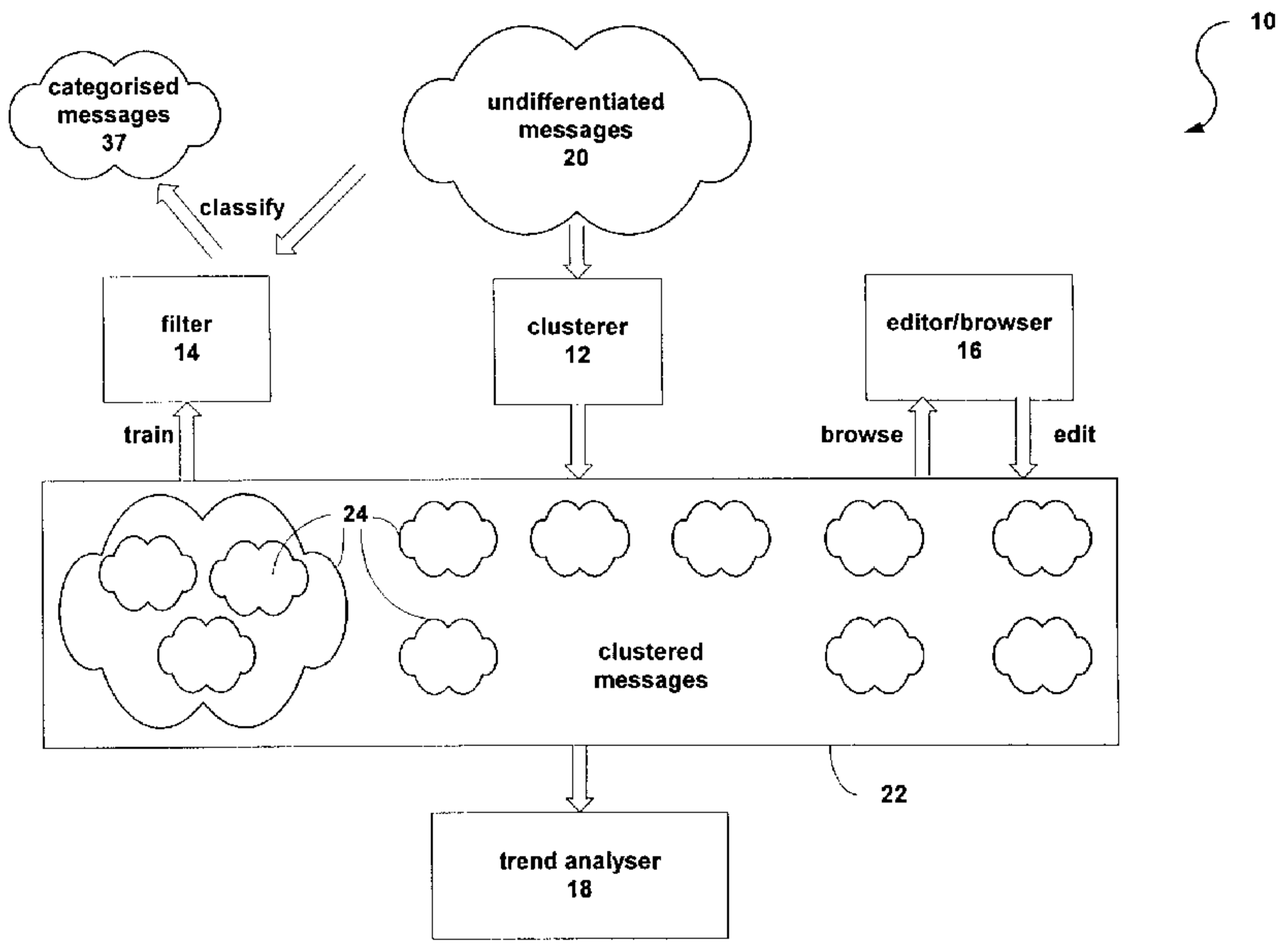
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: A DOCUMENT CATEGORISATION SYSTEM



(57) Abstract: A document categorisation system, including a clusterer for generating clusters of related electronic documents based on features extracted from said documents, and a filter module for generating a filter on the basis of said clusters to categorise further documents received by said system. The system may include an editor for manually browsing and modifying the clusters. The categorisation of the documents is based on n-grams, which are used to determine significant features of the documents. The system includes a trend analyzer for determining trends of changing document categories over time, and for identifying novel clusters. The system may be implemented as a plug-in module for a spreadsheet application, providing a convenient means for one-off or ongoing analysis of text entries in a worksheet.

WO 02/25479 A1

A DOCUMENT CATEGORISATION SYSTEM

The present invention relates to information systems, and in particular to a method and system for categorising electronic documents and for characterising the resulting categories.

The information age brings with it the risk of information overload. In particular, large service organisations typically interact with an enormous number of customers, and the introduction of electronic message handling systems into such organisations necessitates some method of efficiently dealing with large numbers of electronic messages or other forms of electronic documents. It is desired, therefore, to provide a system and method for categorising electronic documents and for characterising the resulting categories, or at least provide a useful alternative to existing systems.

In accordance with the present invention there is provided a document categorisation system including:

a clusterer for generating clusters of related electronic documents based on features extracted from said documents; and

a filter module for generating a filter on the basis of said clusters to categorise further documents received by said system.

The present invention also provides a document categorisation system including:

a clusterer for generating clusters of related electronic documents based on features extracted from said documents; and

an editor for browsing and modifying said clusters.

Preferably, said clusterer is adapted to extract features from electronic documents, determine significant features from said extracted features, and generate clusters of said documents based on said significant features.

Preferably said features include at least one of *n*-grams, words and phrases. Preferably the clusterer further includes a cluster describer module for generating text describing each cluster.

The present invention also provides a document categorisation system including:

an editor for browsing and modifying clustered documents; and

a filter module for generating a filter on the basis of features of said clusters to categorise further documents received by said system.

The present invention also provides a document categorisation system including:

a clusterer for generating clusters of documents by executing unsupervised learning on said documents; and

a filter module for generating a filter to categorise received documents by executing supervised learning on said clusters.

Advantageously the system may further include an editor to adjust said clusters.

Advantageously the system may further include a trend analyzer for determining trends of document categories over time.

The present invention also provides a method for categorising documents, including creating categories for said documents based on feature extraction, where said features include at least one of *n*-grams, words and phrases.

The present invention also provides a method for categorising documents, including:

creating categories for said documents, based on feature extraction; and

manually modifying said categories with a category editor.

Preferably, said method includes selecting features of said documents based on a respective discriminating ability of each feature.

Preferably, said discriminating ability is based on similarities for said documents with and without said feature.

The present invention also provides a method for categorising a document, including:

creating a document filter for a pre-existing document category by analysing pre-existing documents in said category; and

applying said filter to said document in order to determine whether said document belongs in said category.

Preferably, said category is determined by features which are generated by an n-gram extraction process.

Advantageously, said filter may also be used to produce descriptive labels for large document sets.

Advantageously, said descriptive labels may include at least one of phrases and sentences.

Advantageously, said categories may be described by an n-gram extraction process.

Preferably, said method includes determining a trend of a document category over time.

The present invention also provides a data categorisation module for use with a spreadsheet application, said module including:

a cluster module for generating clusters of related data from data in a document of the spreadsheet application, based on extracted features of said data; and

a training module for generating a filter on the basis of said clusters to categorise further data

Preferably, said data categorisation module includes a filtering module for categorising said further data on the basis of said filter.

Preferably, said data includes a plurality of entries within a worksheet of said application.

Advantageously, said entries may include text data to be used for categorising said plurality of entries, and structured data.

Preferably, said cluster module is adapted to generate a cluster identifier for identifying a cluster to which an entry of said data belongs, and a cluster size value for identifying the size of said cluster.

Preferably, said cluster module is adapted to generate at least one category descriptor for said entry.

Preferably, said cluster module is adapted to generate a worksheet column for identifying a category of each entry of said data.

Preferably, said cluster module is adapted to generate a formatted version of text data of an entry for indicating a category descriptor of said entry.

Preferably, said data categorisation module includes a module for testing said filters by categorising training data on the basis of filters generated using said training data.

Preferably, said data categorisation module includes labelling functions for generating a category identifier for an entry of said data. Preferably, said labelling functions include a labelling function for generating a plurality of category columns of said worksheet for identifying at least one category of an entry.

Preferably, said filtering module generates a respective score for each category of an entry. Preferably, said filtering module generates an error for an entry if any one of said scores is inconsistent with a respective category identifier. Preferably, a score indicates that the corresponding entry belongs to a respective category if said score exceeds a pre-determined value. Preferably, the default value of said pre-determined value is zero.

Preferably, scores exceeding said pre-determined value are formatted differently than scores less than said pre-determined value. Preferably, the score may be used to calculate the probability that said entry belongs to the corresponding category.

The present invention also provides a data categorisation module for use with a spreadsheet application, said module including a cluster module for generating clusters of related data from data in a document of the spreadsheet application, based on extracted features of said data.

The present invention also provides a method of data categorisation in a spreadsheet application, including the steps of:

a cluster module for generating clusters of related data from data in a document of the spreadsheet application, based on extracted features of said data; and

a training module for generating a filter on the basis of said clusters to categorise further data.

Preferred embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, wherein:

Figure 1 is a block diagram of a preferred embodiment of a document categorisation system;

Figure 2 is a block diagram of a clusterer of the system;

Figure 3 is a block diagram of a filter module of the system;

Figure 4 is a block diagram showing components of a preferred embodiment of a plug-in data categorisation module for a spreadsheet application; and

Figures 5 to 9 are screenshots of a spreadsheet application with the plug-in module.

A document categorisation system 10 includes a clusterer module 12, a filter module 14, an editor/browser module 16 and a trend analyser module 18. The clusterer 12 processes electronic messages or other documents 20 and groups them into a set 22 of clusters 24 of related documents and then creates a description in the form of a set of keywords and key phrases for each cluster 24. The editor/browser 16 provides an interactive user interface

that allows a data analyst to browse through the clustered documents 22 and to modify the clusters 24 if desired so that each cluster contains a coherent set of documents. The filter module 14 analyses existing clusters in order to categorise new documents. The Trend Analyser 18 analyses clusters over various time periods and compares different categorisations of the same documents to determine a coherent and useful classification for these documents and determines novel clusters.

Together, the modules 12 to 18 constitute a document categorisation system 10 which can, for example, automatically categorise large numbers (typically 10,000 – 25,000) of electronic text messages, such as complaints and survey text, into a much smaller number (approximately 250—500 in this example) of groups or clusters. These messages are typically written under time constraints and they are therefore terse and contain abbreviations as well as typing and spelling errors. The system 10 can also track specific message categories, route messages (e.g., emails), and alert users when there are novel (*i.e.*, unusual) messages.

The clusterer 12 groups together related documents. For navigation and editing purposes, each group is labelled using a cluster description technique, as described below. The grouping and labelling allows a large document repository to be navigated and modified easily. The clustering implementation is based on a document clustering methodology described in Salton, The SMART Retrieval System – Experiments in Automatic Document Processing, Prentice-Hall, New Jersey, 1971 ("Salton"). As shown in Figure 2, the clusterer 12 includes a feature extraction module 26, a feature selection module 28, a cluster generator 30, and a cluster describer module 32. The feature extraction module 26 processes each document to produce a vector of token frequencies, where tokens may be n-grams, words or phrases, and where some tokens may be excluded using feature selection criteria of the feature selection module 28. A similarity measure is then defined as a function of these document vectors to quantify the similarity between any two documents. Finally, the clustering generator 20 uses this similarity measure to group similar documents into clusters.

The feature extraction module 26 extracts *n*-grams, words, and/or phrases as tokens to represent a document or message. N-grams are especially suited for processing of noisy and/or un-grammatical text, such as SMS messages. The *n*-grams are sequences of characters of length *n*, and they are extracted as follows. First, each document is represented as a sequence of characters or a vector of characters (referred to as document-sequence vector). Next, each document-sequence vector is processed to extract *n*-grams and their frequencies (number of occurrences) for that document. For example, the sequence of characters "to build" will give rise to the following 5-grams "to bu", "o bui", " buil", "build". Words and phrases extracted may be stemmed, although that is not necessary. Because the clustering process uses *n*-gram vectors rather than word vectors only, it is tolerant of spelling and typing errors, because a single error in the spelling of a long word nevertheless yields *n*-grams that are the same as those from the correctly spelled word.

The feature selection module 28 of the clusterer 12 is executed before clustering to include in the document vectors only those features that provide significant information for clustering. This not only reduces time for clustering by reducing the feature space, but also increases the accuracy of clustering, because noisy features are eliminated. The feature selection process determines the discriminating ability of a feature. The ability of a feature to discriminate depends on how many documents a feature appears in (referred to as DF), and the frequency of that feature in each document (referred to as TF). Those features that appear frequently within few documents are more discriminating than those that appear infrequently in most of the documents. Traditionally, in information retrieval, this reasoning is captured using the TFs and DF of each feature to arrive at an importance value.

In the document categorisation system 10, the computed discrimination ability is based on the premise that if a very good discriminating feature is removed from the feature space, then the average similarity between documents in the repository increases significantly. Thus, by computing average similarity between documents with and without a feature, it is

possible to determine the feature's discriminating ability. The similarity of a particular document is computed with a cosine coefficient, as is described in Salton. The average similarity of a document set is determined by summing the similarities between each document and the centroid (where the centroid is the average of feature frequency vectors of all the documents in the set). This method for feature selection has been traditionally ignored due to its computational cost, since it involves nm similarity computations, where n is the number of documents and m is the total number of words or features. As the following equation shows, each cosine similarity computation itself has a computational complexity of m^2:

$$s_{ij} = \frac{\sum_{k=1}^{m} f_{ik} f_{jk}}{\sqrt{\sum_{k=1}^{m} f_{ik}^2} \sqrt{\sum_{k=1}^{m} f_{jk}^2}}$$

where s_{ij} is the similarity measure, $i, j = 1, \ldots, n$ represents the document number in the document set, and f represents the frequency of occurrence of a feature denoted by the integer k.

However, by storing the norm of frequency vectors and their dot products with the centroid, *i.e.*, $\sqrt{\sum_{k=1}^{m} f_{ik}^2}$ and $\sum_{k=1}^{m} f_{ik} f_{jk}$, respectively, with j being the centroid document, the feature selector is able to compute scores for each feature in linear time. This score determines discrimination ability features for clustering better than the standard inverse document frequency-term frequency (IDF-TF) scores described in Salton.

After the features have been extracted and selected, the clusters are defined by the cluster generator 30. The cluster generator 30 uses a clustering algorithm that is an enhancement of a single-pass, non-hierarchical method which partitions the repository into disjoint sets, as described in E. Rasmussen, "Clustering Algorithms", Information Retrieval, W. B. Frake and R. Baeza-Yates ed., Prentice-Hall, New Jersey, 1992. The algorithm proceeds as follows: the first document $D1$ is used to create the first cluster $C1$. Each remaining

document, *Dk*, is assigned to the nearest cluster *Cj*, or a new cluster if none is sufficiently close. In order to compare documents to clusters, each cluster is represented by its centroid, which is the average of word frequency vectors of all the documents in the cluster. A new cluster is started when none of the existing clusters are sufficiently close, based on a specified similarity or distance threshold *T*.

As mentioned above, the clustering algorithm creates no hierarchies. However, hierarchies can be implemented by clustering at the first level, and then clustering the clusters using cluster centroids. Since the algorithm itself is not hierarchical, the complexity is $O(nm)$, where n is the number of entities to cluster at each level and m is the number of clusters.

In order to determine if documents are sufficiently close, traditional single-pass algorithms require the threshold *T*, the separation between groups, to be specified prior to clustering. This may result in sub-optimal clustering, because the groupings are primarily dependent on the contents of the repository. In contrast, the algorithm used by the clusterer 12 determines the number of clusters by creating different groupings of the data set at different separation thresholds and then evaluating these groupings to determine the best grouping. The evaluation takes into account the affinity within groups as well as separation from other groups, as described in B. Raskutti and C. Leckie, "An Evaluation of Criteria for Measuring the Quality of Clusters", pp. 905—910, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999. Thus, the clustering algorithm imposes a structure (a hierarchy) on previously unstructured documents. Since the clustering is based on the tokens (*i.e.*, n-grams, words and/or phrases) in a message, as well as tokens in other messages, it typically captures the conceptual relations.

The clusterer 12 also labels groups based on their contents using the cluster describer module 32. It extracts words, phrases and/or sentences from a document that are most descriptive of that document, in contrast to standard approaches which extract words only. The method is similar to that described in J. D. Cohen, "Highlights: Language and Domain Independent Automatic Indexing terms for Abstracting", Journal of the American Society

for Information Science, 46(3): 162: 174, 1995 for generating highlights or abstracts for documents that are retrieved by an information retrieval system.

In order to determine the words and phrase that describe a group of documents, the following steps are performed by the cluster describer module 32. First, all documents within a group are combined into one *hyper-document*. In all further analysis, the hyper-documents are treated as a single document, and it is these hyper-documents that are to be described or labelled. Second, the distribution of the *n*-grams over the document space is computed by counting the occurrence of *n*-grams in the documents. Next, each *n*-gram is assigned a score per document that indicates how novel or unique it is for the document. This novelty score is based on the probability of the occurrence of the *n*-gram in the document and the probability of occurrence elsewhere (*i.e.*, in another cluster). Next, the novelty score of each *n*-gram is apportioned across the characters in the *n*-gram. For example, the apportioning could be so that the entire score is allocated to the middle character, and the other characters are assigned a score of zero. This apportioning allows each character in the sequence (hence each entry in the document-sequence vector) to be assigned a weight. Finally, these weights are used to compute a score for each word or phrase based on their component characters and their scores. These scores may be used as is or combined, if necessary, with language-dependent analysis, such as stemming, to filter out non-essential features.

Thus, for each document (*i.e.*, the hyper-document for each group of documents), the output of the cluster describer 32 is a set of descriptors (words or phrases) in the document with associated scores indicating how well they describe the document. In the case of a hyper-document, these scores measure how close the descriptors are to the topic of the cluster, and are used to create a succinct description of a cluster. This also means that a descriptor may have different scores for different clusters. For instance, the word "vector" may have a higher score in a cluster about vector analysis and a lower score in a cluster about vector analysis for search engines. This score is then used to rank different descriptors for a cluster such that the most descriptive term is listed first.

Because the system 10 is based on *n*-grams, there is no necessity for language-dependent pre-processing such as stemming and removal of function words. Hence, the extraction process is language-independent and is tolerant of spelling and typing errors. Furthermore, this technique can be used to extract words or phrases or sentences or even paragraphs (for large documents), since the fundamental units for determining novelty are not words but character sequences. Moreover, these character sequences need not even be text. Hence, with minor modifications, the same technique may be used to pick out novel regions within an image or audio track.

The clusterer 12 described above imposes a structure on a previously unstructured collection of documents. This structure, typically, captures the appropriate conceptual relations; however, as with all categorizations, whether manual or automatic, the clusters may require further refinement. The editor/browser 16 provides the ability to manually alter the automatically created groups for greater coherence and usability. It provides a visual user interface to allow browsing of the cluster hierarchy and a number of editing functions. These editing functions include file manager like functions, such as the ability to create and delete clusters, edit cluster descriptions, and move messages/clusters to other clusters. However, it also allows the user to (i) highlight words and/or phrases in documents to indicate the cluster topic, (ii) create sub-clusters within very large and diverse clusters (using the clusterer 12), and (iii) create a filter for a cluster/category, using the messages under that cluster as examples (by using filter module 14).

As shown in Figure 3, the filter module 14 includes the cluster describer 32 described above, a filter generator 31, a feature extractor 26, and a filter applier 35. The filter module 14 has two modes: a training mode and a classification mode. In the training mode, the filter module 14 learns the features of messages belonging to a particular category based on examples of earlier documents. These example documents for the categories may have been grouped by the clusterer 12 and/or the browser/editor 16, or created by some other means. The filter module 14 generates a filter 33 based on the characteristics of messages in the categories. The resulting filter 33 is subsequently used by the filter module 14 in its classification (categorisation) mode, whereby each new message is tagged with scores

indicating the likelihood that it belongs to a particular category. For particular applications, such as complaint analysis, if a new message does not fit in with any of the previously defined categories, it can be defined as a novel message, and a data analyst is alerted by e-mail or otherwise.

The filter training process begins with a feature extraction process, executed by the feature extractor 26 cluster describer 32, that extracts terms (words, phrases and/or *n*-grams) that are useful to distinguish one category from another. This advantageously uses the cluster describer for extracting words and phrases, however, if desired, other feature extraction processes may be used. The words, phrases and *n*-grams so selected are called terms, and these terms are stored in a dictionary 27. Each example in the training set is then represented by a term vector, describing frequencies of appearance of terms from dictionary 27 in the example. The filter generator 31 then uses these term vectors to learn models for discriminating each category from the others. The filter generator 31 then generates a filter 33 containing data representing the resulting models. This requires special machine learning technique capable of dealing with sparse high-dimensional data spaces. The particular learning technique that is used is that for Support Vector Machines (SVM). The technique is described in the specification of Australian Patent Application PQ6844/00, which is incorporated herein by reference. SVM is a relatively novel approach to supervised learning used for both classification and regression problems. The word supervised refers to the fact that the data is labelled. The aim of training in such a case is to look for patterns in the data that conform to this labelling, and to build a model correlated with the labels and capable of predicting labels for new examples with high accuracy. This is different from the unsupervised learning or clustering used in the clusterer 12, whereby a natural or inherent groupings are developed, based on patterns in the data.

SVM techniques are particularly useful for text categorisation due to the fact that text categorisation involves large sparse feature spaces, and SVMs have the potential to handle such feature spaces. The training algorithm is used by the filter module 14 both to learn from very few examples as well as to learn long-term filters using large numbers of examples.

The filter 33 contains information that is used by the filter applier 35 to determine a numerical score for a new message which has been converted to a term vector. The term vector provides occurrence frequencies for terms in the dictionary 27 that appear in the message. In the simplest case of an SVM with linear kernel, the filter 33 for a category is simply a vector of weights, one for each entry in the dictionary 27. The filter applier then determines a dot product between the term vector of the message and the weight vector for the category to obtain a numerical score:

$$score_{ij} = \sum_{k=1}^{m} w_{ik} f_{jk},$$

where w_{ik}, $k = 1,.., m$ is a weight vector for the category i and f_{jk}, $k = 1,\ldots, m$, is the term vector for message j.

The Trend Analyser 18 performs two primary functions. The first is to track specific categories of messages over different time periods. This may be as simple as a comparison of the number of messages in a chosen category at different time periods. However, the analyser also performs a thorough analysis of category variation at different periods, such as movement of the centroid (average of word frequency vector for the messages in that category), categorisation confidence, and other indications of category change. Thus, this functionality is focussed on understanding a single document category over time. If there are very few messages for a particular category in consecutive time periods, this might indicate that the corresponding filter is no longer required. Conversely, if the number of messages within a group is getting larger and unmanageable, it may be necessary to use the clusterer 12 to create finer partitioning of this category.

The second function of the trend analyser 18 is to compare different categorisations of the same documents in order to determine a coherent and useful classification for these documents. When using the filter module 14 to create categorisations for ongoing analysis, it may be necessary to check that the nature of the actual documents has not deviated too far from the filters that were created, and that the filters are up-to-date. This requires a user to compare the categorisations produced by the clusterer 12 with those produced by the

filter module 14, and to highlight the differences so that filter definitions may be modified accordingly. Thus, this functionality is focussed on analysing the whole message space and what categories need to be defined to understand that space.

The document categorisation system 10 may be used in several modes. Several examples of applications are:

(i) One-off analysis, *e.g.*, to understand information from customer surveys. This mode requires only the clusterer 12 and editor/browser 16 in order to understand broad tendencies expressed in surveys.

(ii) On-going analysis of unstructured information, *e.g.*, to understand and monitor customer complaints. In this mode, firstly messages are clustered and hierarchies edited during the initial period in order to identify coherent categories, and then filters are created for these categories in order to classify new data. Novel messages may then need to be clustered in new categories. Thus, this mode requires all four modules 12 to 18 of the system 10.

(iii) Relevance feedback mode for on-going analysis of information that is well understood, but not necessarily categorised. In this mode, data analysts may know roughly what categories they need, although they may not have a large set of examples to define these categories. In this mode, the following steps are executed:

(a) create a filter for a specific category by learning from very few examples;

(b) use the filter to sort all messages so that those in the category appear at the top;

(c) use the sorted list to manually increase the number examples for that category; and

(d) iterate through steps (a) to (c) until a satisfactory filter is created.

This mode uses manual feedback information regarding the accuracy of the filter to modify the filter very quickly. This can be used as a tool to quickly create a training set from a large uncategorised set of messages.

This feedback loop may be incorporated into a number of different applications. In a search engine, this loop may be used to refine results based on relevance

feedback. In a spam filter, it may be used for quickly creating examples of spam and non-spam email messages. This mode requires the filter module 14 and the editor/browser 16 to be suitably tailored.

The relevance feedback mode can be used for once-off analysis as described in (i) on its own, or in conjunction with clustering.

(iv) Adaptive learning of multiple filters, *e.g.*, routing of customer communications where different filters not only provide rejections, but also suggestions as to which other filter should have been the recipient of a particular message. This mode is similar to the previous mode except that multiple filters are adjusted simultaneously, in response to feedback from multiple human operators.

(v) Thesaurus creation mode, for understanding how different words/acronyms used in the messages relate to each other. The words are clustered based on which documents they appear in, and this organisation may be browsed and edited to understand the relationship between words.

Service providers such as telecommunication companies, banks and insurance companies often need to process large numbers of short text messages, such as complaints, exit interviews, survey information, and so on. As described above, these messages are typically written down under time constraints, and consequently they may be short and contain abbreviations and typing/spelling errors. Despite this, such messages generally capture the writer's intent/meaning very clearly. Once captured, these text messages are usually associated with structured information such as the category of customer, location of service, etc. The structured information is typically analysed using a spreadsheet application, while the text messages themselves are ignored or interpreted manually, by a customer service representative, for example.

In an alternative embodiment of the present invention, the clustering and the filtering methods of the categorisation system are embodied in components of a plug-in module 34 for a spreadsheet application 36 such as Microsoft Excel®, as shown in Figure 3. The plug-in module 34 includes a ClusterData module 38, a TrainFilters module 40, a

TestFilters module 42, a FilterData module 44, support function modules 46, and interface data 48. The combination of the spreadsheet application 36 and the plug-in module 34 provides a convenient system for categorising short text messages, and allows the spreadsheet application 36 to be used to analyse databases that include both structured information and free-text messages, such as a customer complaints database. In particular, the plug-in 34 provides the ability to:

(i) automatically group related messages into clusters that have natural affinity, and describe these clusters for easy browsing;

(ii) manually browse and alter the groupings to suit business needs;

(iii) learn models for these groups and save them so that new messages can be classified using these models; and

(iv) classify new messages using earlier models.

As shown in Figure 3, the plug-in interface data 48 provides a number of additional buttons 50 to 76 to the spreadsheet application toolbar area 52, allowing a user of the application to analyse free-text data. The buttons include presentation buttons 50 to 60, label buttons 62 to 68, and module invocation buttons 70 to 76. The module invocation buttons 70 to 76 will be described first. A *ClusterData* button 76 invokes the ClusterData module 38 of the plug-in 34, allowing the user to cluster data selected in a worksheet 78 into groups or categories in order to provide one-off analysis, *e.g.*, to understand information from customer surveys. Each row of the worksheet 78 is considered to be a separate entry or record to be clustered. Each entry may be associated with one or more category labels which are integers that identify the categories to which the entry belongs, as described below.

A *TrainFilters* button 70 invokes the *TrainFilters* module 40 for learning a model of the various groups or categories on the basis of selected data in the worksheet 78. The selected data includes the textual data for each entry, and category labels used to identify the categories to which each entry belongs. Such models may be used for on-going analysis of unstructured information, *e.g.*, to understand and monitor customer complaints.

A *TestFilters* button 72 invokes the TestFilters module 42 that tests the models created by the TrainFilters module 40 on data whose labels are known. For example, a simple sanity check for a new model is to select the data that was used for training in order to confirm that the model can at least classify the training data correctly.

A *RunFilters* button 74 invokes the FilterData module 44 that uses the models created by TrainFilters module 40 on new (*i.e.*, uncategorised) data. In this case, the labels of input entries are unknown, and the models are used to predict labels.

The remaining buttons 50 to 68 invoke support function modules 46 that allow easy visualisation and manipulation of groups and individual entries in order to facilitate the structuring of textual data within the spreadsheet application 36. These support functions 46, along with the analysis buttons 70 to 76, may be used to perform one-off analysis of textual data, *e.g.*, to understand information from customer surveys, and/or on-going analysis of unstructured information, *e.g.*, to understand and monitor customer complaints. In the latter mode, it is first necessary to cluster and edit groups in order to identify coherent categories, and then create filters for these categories for classifying new data. Novel messages may then need to be clustered and new categories created if there are sufficient numbers of novel messages of any one category.

For example, column C of the worksheet area 78 of Figure 4 includes customer text sent to a service provider from mobile telephones using short message services (SMS). The text entries contain typing errors and non-alphanumeric characters. In this example, columns A and B contain structured data which is ignored: Column A provides a list of unique message identifiers, and column B provides a timestamp indicating when the corresponding message was received. The plug-in module 34 of Figure 4 allows related messages to be identified by invoking the ClusterData module 38. The ClusterData module 38 groups together textually related entities within a selected region of data in the worksheet area 78, and highlights key descriptors in each group to facilitate visual determination of whether a group is homogeneous or not. When the ClusterData button 76 is selected, the ClusterData module 38 begins processing the selected data. If the first row

of the selection is the heading row (*i.e.*, row number 1), it is ignored. Each of the other rows of the selected data is considered an entity, and textually related entities are brought together by rearranging the order of rows within the selection using the clustering methods described above. Each row may include structured data for other analyses (*e.g.*, columns A and B in this example), but the last column (which may be the only column) contains the textual information that is used for clustering. The output of the ClusterData module 38 is provided in a new worksheet and allows the user to readily perceive the key clustering concepts. As shown in Figure 5, each row includes a cluster identification number column 80, a cluster size column 82, description columns 84 to 92 for each cluster, an initially empty filter identifier column 94, and the original input data in the last three columns 96 to 100. The maximum number of descriptions per cluster is specified by the user, and has been set to five in the example implementation shown in Figures 6 to 9. The last column 100 of the original input containing the textual data is modified so that key descriptors of the cluster are highlighted by colour. The rows are sorted so that rows with the same cluster identifier are together, and clusters with larger number of entries appear before those with smaller number of entries to ensure that a large number of entries can be processed quickly by the user. Initially, only one row per cluster is presented so that key concepts/topics from the textual data may be readily perceived by visual inspection. Alternate cluster rows are shaded in order to visually distinguish adjacent clusters from each other.

This initial presentation may be altered by means of presentation buttons 50 to 60. A *HideDescription* button 50 allows the user to hide the five columns 84 to 92 containing the cluster descriptors. The highlighting of descriptors in the textual data ensures that it is still possible to determine the descriptions when the descriptor columns 84 to 92 are hidden. A *ShowDescription* button 52 reverses the effect of the *HideDescription* button 50. An *ExpandOne* button 58 expands the presentation to show all the entries for one cluster. An *ExpandAll* button 60 performs the same function for all clusters within the selection. This enables quick visual inspection of one or all clusters so as to identify whether clusters are homogeneous or not. A *CollapseOne* button 54 and a *CollapseAll* button 56 reverse the effects of the *ExpandOne* button 58 and *ExpandAll* button 60, respectively.

Using the presentation buttons 50 to 60, clustered data may be easily inspected for major conceptual categories that emerge from the data. For instance, cluster numbers 1, 4 and 7 of Figure 5 belong to the same conceptual grouping of people expressing greetings. Depending on the outcome required, these groups may be further merged with other groups such as 6, 9 and 21 as a single category that does not require any action from the service provider.

After text categories have been determined, a large number of entries may be annotated with a filter/category label. The process of annotation is facilitated by means of label buttons 62 to 68. A *DefaultLabels* button 66 labels each entry, *i.e.*, creates a filter identifier label in the filter identifier column 94, by copying the numeric value from the cluster identifier column 80. If the number of clusters is small, and the clusters are homogeneous, this is a quick method for labelling entries. A *LabelOne* button 62 copies the label (from the filter identifier column 94) from the first entry (*i.e.*, row) of a cluster down to all other entries of the same cluster. A *LabelAll* button 64 performs this action for all clusters within the selection. These buttons 62, 64 simplify labelling because all the user is required to do is to first check the homogeneity of clusters, then label the first entry of each cluster, and then use the LabelOne button 62 or the LabelAll button 64 to label the other entries of the clusters.

A *CreateMultipleLabels* button 68 may be used to allow entries to belong to multiple categories by expanding the single label provided by the filter identifier column 94 to provide one column for each of the different labels (*i.e.*, categories) within the selected region, as shown in Figure 6. Fields in the header row provide descriptors for the categories, and each entry within a data row provides an integer boolean value (*i.e.*, 1 or 0) indicating whether the corresponding entry belongs to that category.

The TrainFilters module 40 uses supervised learning to model the categories that have been defined by the ClusterData module 38 and may have been modified by the user editing the worksheet directly and/or using the labelling buttons 62 to 68. Input to the

TrainFilters module 40 has a particular form, beginning with label columns 102 and ending with the textual data column 100, as shown in Figure 6. The category label columns 102 in this example were obtained by using the CreateMultipleLabels button 68 after labelling the data using the cluster identifier, the highlighting of keywords to determine whether entries belong to a particular group, and the LabelOne button 62 and the LabelAll button 64. After this initial labelling, an entry may be placed into multiple categories, if desired (*e.g.*, message numbers 10 and 27). The labelled data, which was first sorted in the order of cluster identifier for ease of labelling, has been resorted so that it is in ascending order of the message identifier. If desired, zero values in the category columns 102 can be hidden to improve data visualisation, as shown in Figure 8.

The TrainFilters module 40 learns a model for each category/label after the required rows/columns in the worksheet 78 are selected. Because the category/label information and the textual data for learning come from an arbitrarily-positioned selected portion of a single worksheet 78, the TrainFilters module 40 requires the number of labels and the starting column of labels to be input by the user, and for the label columns to be contiguous. The first row is assumed to be the header row and is ignored. Models are then learnt from the textual data and the labels (which identify a filter) in the other selected rows using the training method described above.

The output from the TrainFilters module 40 is provided as a new worksheet, as shown in Figure 7. The worksheet includes the input data columns 96 to 100, the label columns 102, and additional information in newly introduced columns (coloured blue) 104, 106. The new score columns 104 provide a score for each label/entry combination. The scores are colour-coded, with positive scores in a black typeface and negative scores in a red typeface. For the purposes of categorisation, an entry is deemed to belong to all those categories for which it has a positive score. However, for a higher confidence level, the threshold can be changed from 0 to a positive value. The error column 106 indicates whether there is an error in classification for each entry; that is, whether the classifications indicated by the category columns 102 are consistent with the values in the score columns 104. Errors in the training set are generally rare, since the model is learnt on the basis of

the training set. A more thorough validation of the model may be provided by using the TestFilters module 42.

The TrainFilters module 40 produces internal models for each of the categories. These models can then be used for classification using the TestFilters module 42 or the FilterData module 44. Optionally, the scores may be automatically converted to confidence levels, which represent an estimate of the probability that an entry belongs to the corresponding category. The TestFilters module 42 uses the models created by TrainFilters 40 on a data set whose labels are known. It provides a more thorough validation of a model, because the model was created without reference to these labels and data. The input and output formats for the TestFilters module 42 are the same as those for the TrainFilters module 40. However, the error column is more likely to be populated when the models are used to categorise new data.

The FilterData module 44 uses the models created by TrainFilters 40 to classify a data set whose labels are unknown. The input format is the same as that for clustering (one entry per row, with the first row assumed to be a header row). As shown in Figure 8, the FilterData module 44 creates score columns 104 and category columns 102. The first *n* columns 104 (where *n* is the number of labels/categories) provide the score, with positive scores in black and negative scores in red. The next *n* columns 102 have either a 1 for all those labels that have a score greater than 0 (or the threshold value set by the user), or are empty. The 1 in a column corresponding to a label indicates that the entry belongs to that category/label. Because the scores are present, the threshold may be adjusted if a more stringent classification is required, without the need for further analysis.

Many modifications will be apparent to those skilled in the art without departing from the scope of the present invention as herein described with reference to the accompanying drawings.

AMENDED CLAIMS:

1. A document categorisation system including:
a clusterer for generating clusters of related electronic documents based on features extracted from said documents; and
a filter module for generating a filter on the basis of said clusters to categorise further documents received by said system.

2. A document categorisation system including:
a clusterer for generating clusters of related electronic documents based on features extracted from said documents; and
an editor for browsing and modifying said clusters.

2. What changes?

3. (Amended) A document categorisation system as claimed any one of claims 1 and 2, wherein said clusterer is adapted to extract features from electronic documents, determine significant features from the extracted features, and generate clusters of said documents based on said significant features.

4. A document categorisation system as claimed in any one of the preceding claims, wherein the clusterer further includes a cluster describer module for generating text describing each cluster.

5. (Amended) A document categorisation system including:
an editor for browsing and modifying clusters of documents; and
a filter module for generating a filter on the basis of features of said clusters to categorise further documents received by said system.

6. A document categorisation system as claimed any one of the preceding claims, wherein said features include at least one of n-grams, words and phrases.

7. A document categorisation system including:

a clusterer for generating clusters of documents by executing unsupervised learning on said documents; and

a filter module for generating a filter to categorise received documents by executing supervised learning on said clusters.

8. A document categorisation system as claimed in claim 1 or claim 7, further including an editor for adjusting said clusters.

9. A document categorisation system as claimed in any one of the preceding claims, further including a trend analyzer for determining trends of document categories over time.

10. A method for categorising documents, including creating categories for said documents based on feature extraction, where said features include at least one of n-grams, words and phrases.

11. A method for categorising documents, including:
creating categories for said documents, based on feature extraction; and
manually modifying said categories with a category editor.

12. A method for categorising documents, as claimed in claim 10 or claim 11, including selecting features of said documents based on respective discriminating abilities of the features.

13. A method for categorising documents, as claimed in claim 12, wherein each said discriminating ability is based on similarities for said documents with and without said feature.

14. A method for categorising a document, including:
creating a document filter for a pre-existing document category by analysing pre-existing documents in said category; and

applying said filter to said document in order to determine whether said document belongs in said category.

15. A method as claimed in claim 14, including generating descriptive labels for large document sets.

16. A method as claimed in claim 14, wherein said filter is also used to produce descriptive labels for large document sets.

17. A method as claimed in claim 15, wherein said descriptive labels include at least one of phrases and sentences.

18. A method as claimed in any one of claims 10 to 13, wherein said features are described by an n-gram extraction process.

19. A method as claimed in any one of claims 14 to 17, wherein said filters are generated using features which are selected using an n-gram extraction process.

20. A method as claimed in any one of claims 10 to 19, include determining a trend of a document category over time.

21. A data categorisation module for use with a spreadsheet application, said module including:

a cluster module for generating clusters of related data from data in a document of the spreadsheet application, based on extracted features of said data; and

a training module for generating a filter on the basis of said clusters to categorise further data.

22. A data categorisation module as claimed in claim 21, including a filtering module for categorising further data on the basis of said filter.

23. A data categorisation module as claimed in claim 21, wherein said data includes a plurality of entries in the document.

24. A data categorisation module as claimed in claim 23, wherein said entries include text data to be used for categorising said plurality of entries, and structured data.

25. A data categorisation module as claimed in claim 21, wherein said cluster module is adapted to generate a cluster identifier for identifying a cluster to which an entry of said data belongs, and a cluster size value for identifying the size of said cluster.

26. A data categorisation module as claimed in claim 25, wherein said cluster module is adapted to generate at least one category descriptor for said entry.

27. A data categorisation module as claimed in claim 21, wherein said cluster module generates a worksheet column for identifying a category of each entry of said data.

28. A data categorisation module as claimed in claim 27, wherein said cluster module generates a formatted version of data of an entry for indicating a category descriptor of said entry.

29. A data categorisation module as claimed in claim 21, wherein said data categorisation module includes a module for testing said filters by categorising training data on the basis of filters generated using said training data.

30. A data categorisation module as claimed in claim 21, wherein said data categorisation module includes labelling functions for generating a category identifier for an entry of said data.

31. A data categorisation module as claimed in claim 30, wherein said document is a worksheet and said functions include a labelling function for generating a plurality of category columns of said worksheet for identifying at least one category of an entry.

32. A data categorisation module as claimed in claim 22, wherein said filtering module generates a respective score for each category of an entry.

33. A data categorisation module as claimed in claim 32, wherein said filtering module generates an error for an entry if any one of said scores is inconsistent with a respective category identifier.

34. A data categorisation module as claimed in claim 32, wherein a score indicates that the corresponding entry belongs to a respective category if said score exceeds a pre-determined value.

35. A data categorisation module as claimed in claim 34, wherein the default value of said pre-determined value is zero.

36. A data categorisation module as claimed in claim 34, wherein scores exceeding said pre-determined value are formatted differently than scores less than said value.

37. A data categorisation module as claimed in claim 32, wherein a score may be used to calculate a probability that said entry belongs to the corresponding category.

38. A data categorisation module for use with a spreadsheet application, said module including a cluster module for generating clusters of related data from data in a document of the spreadsheet application, based on extracted features of said data.

39. (Amended) A method of data categorisation in a spreadsheet application, including the steps of:

generating clusters of related data from data in a document of the spreadsheet application, based on extracted features of said data; and

generating a filter on the basis of said clusters to categorise further data.

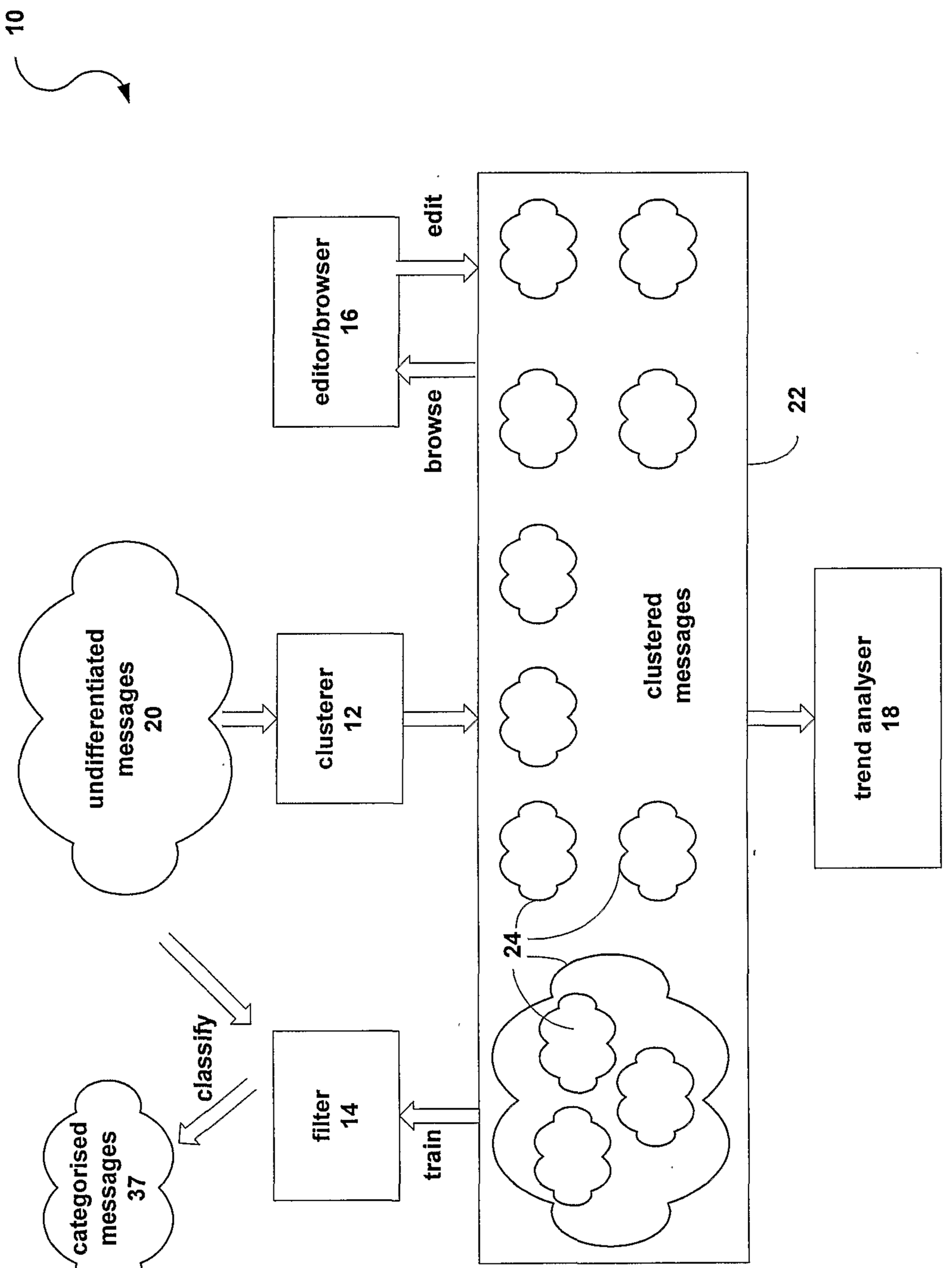

10
editor/browser
16
edit
browse
22
undifferentiated
messages
20
clusterer
12
clustered
messages
trend analyser
18
24
classify
filter
14
train
categorised
messages
37

Figure 1

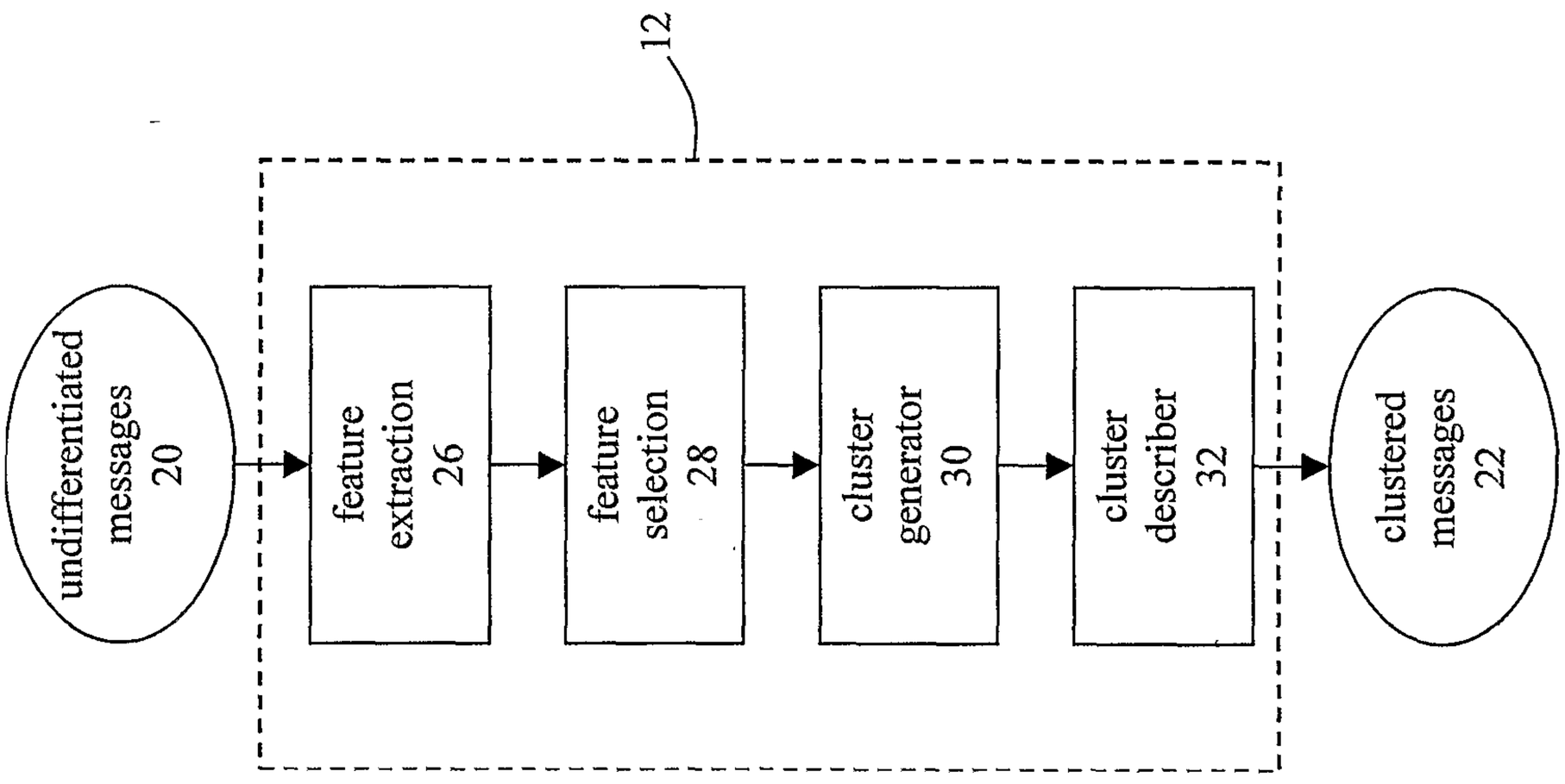

12
undifferentiated messages 20
feature extraction 26
feature selection 28
cluster generator 30
cluster describer 32
clustered messages 22

Figure 2

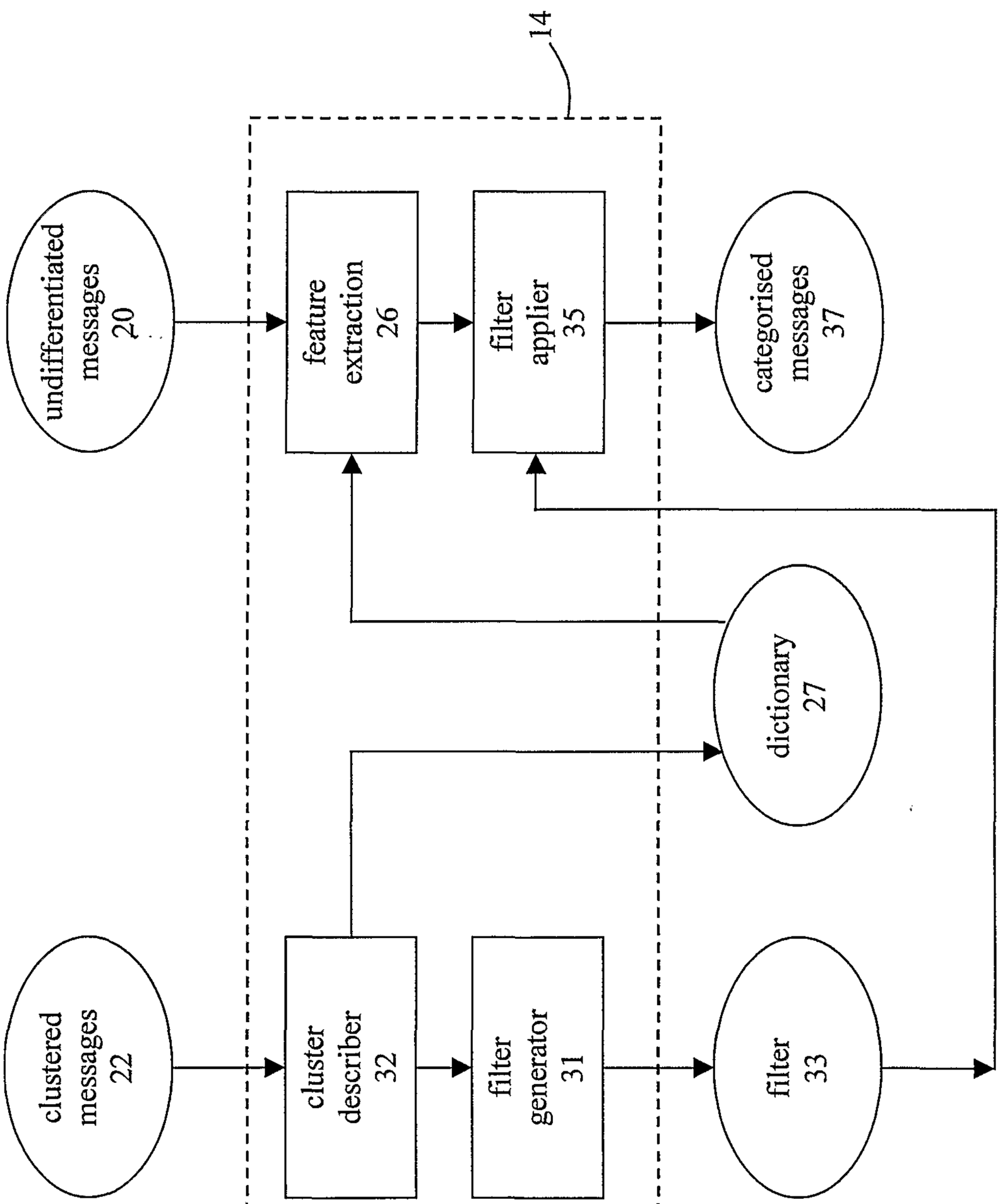

4
undifferentiated messages 20
feature extraction 26
filter applier 35
categorised messages 37
dictionary 27
clustered messages 22
cluster describer 32
filter generator 31
filter 33

Figure 3

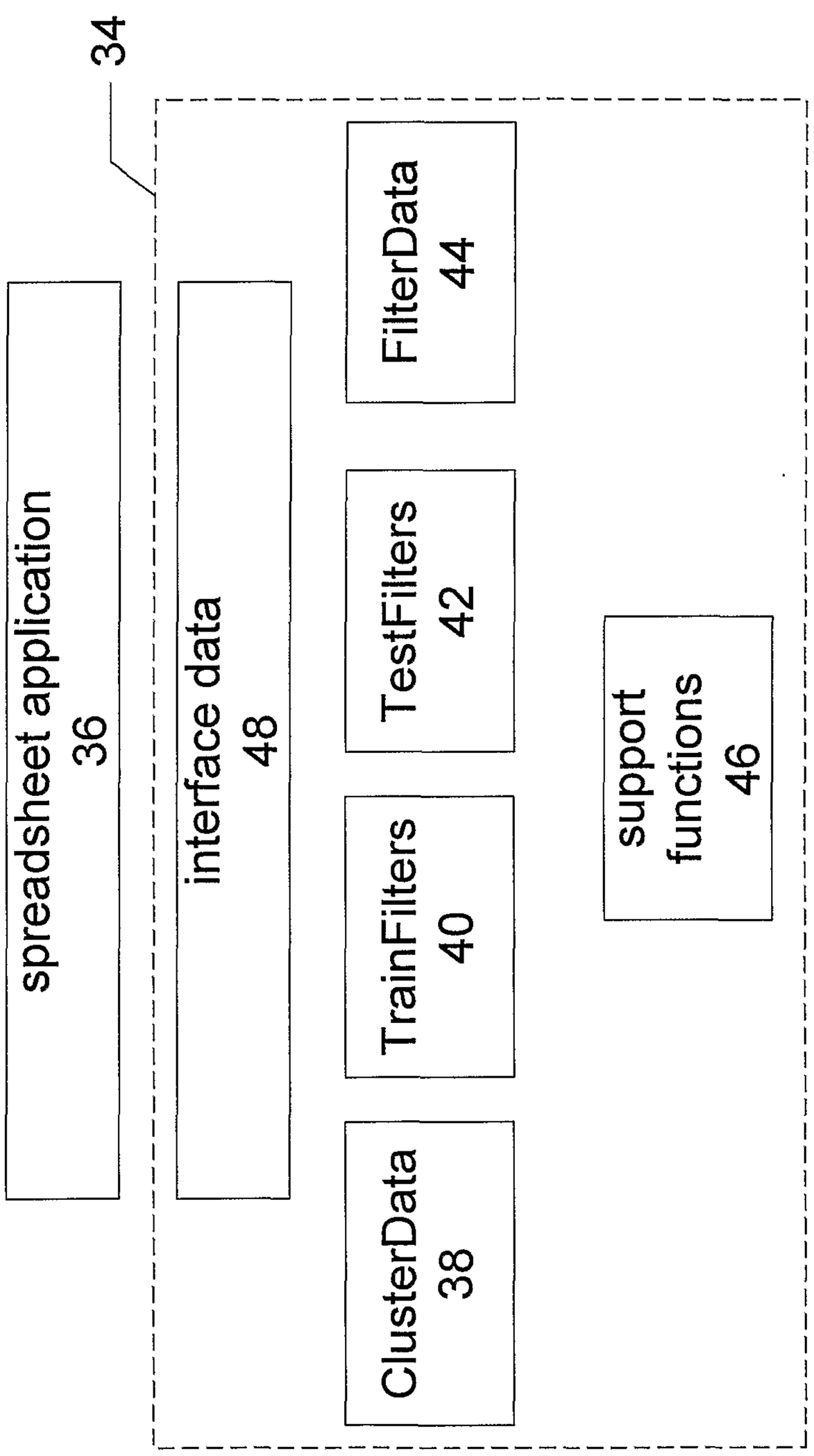

34
spreadsheet application
36
interface data
48
ClusterData
38
TrainFilters
40
TestFilters
42
FilterData
44
support functions
46

Figure 4

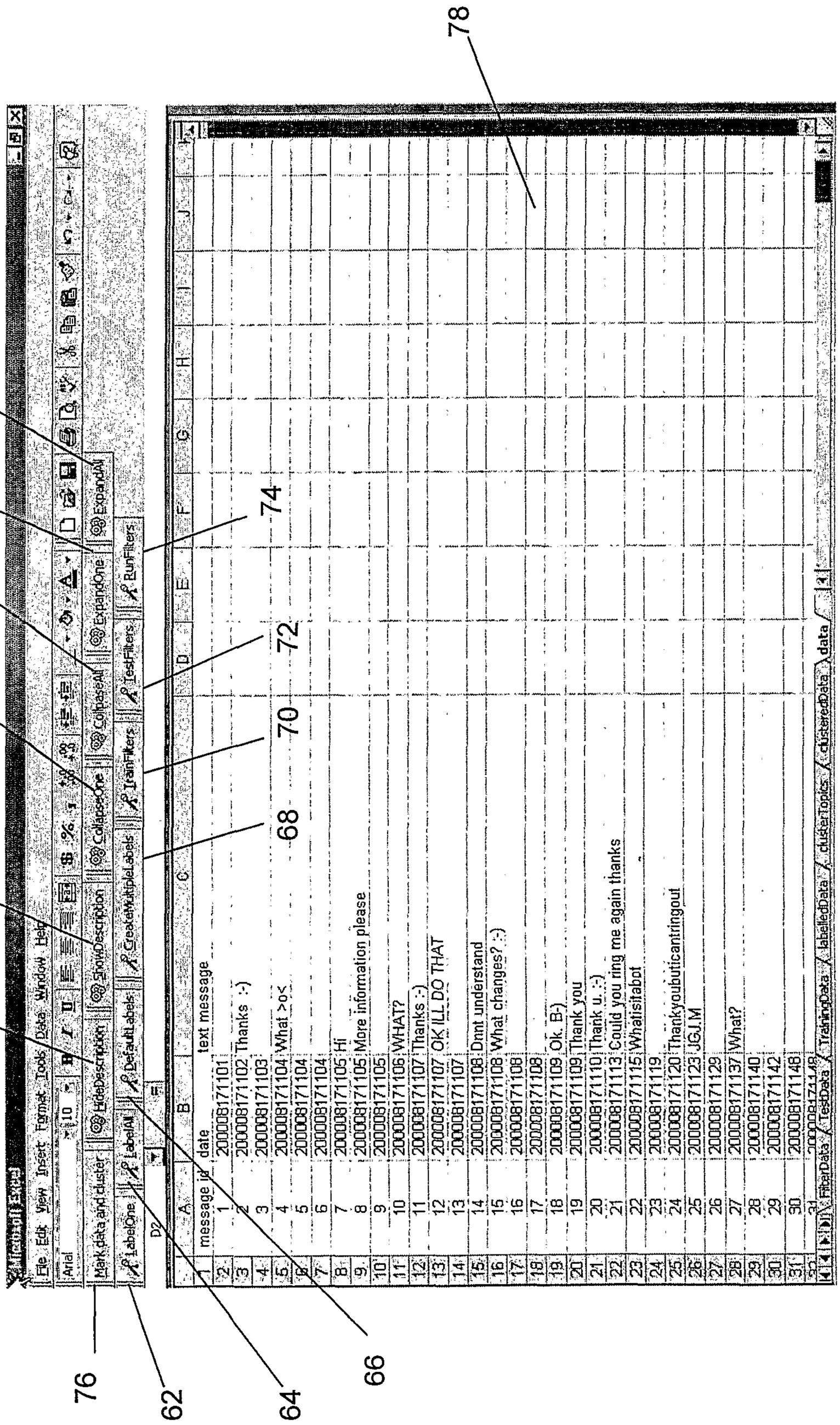


Figure 5

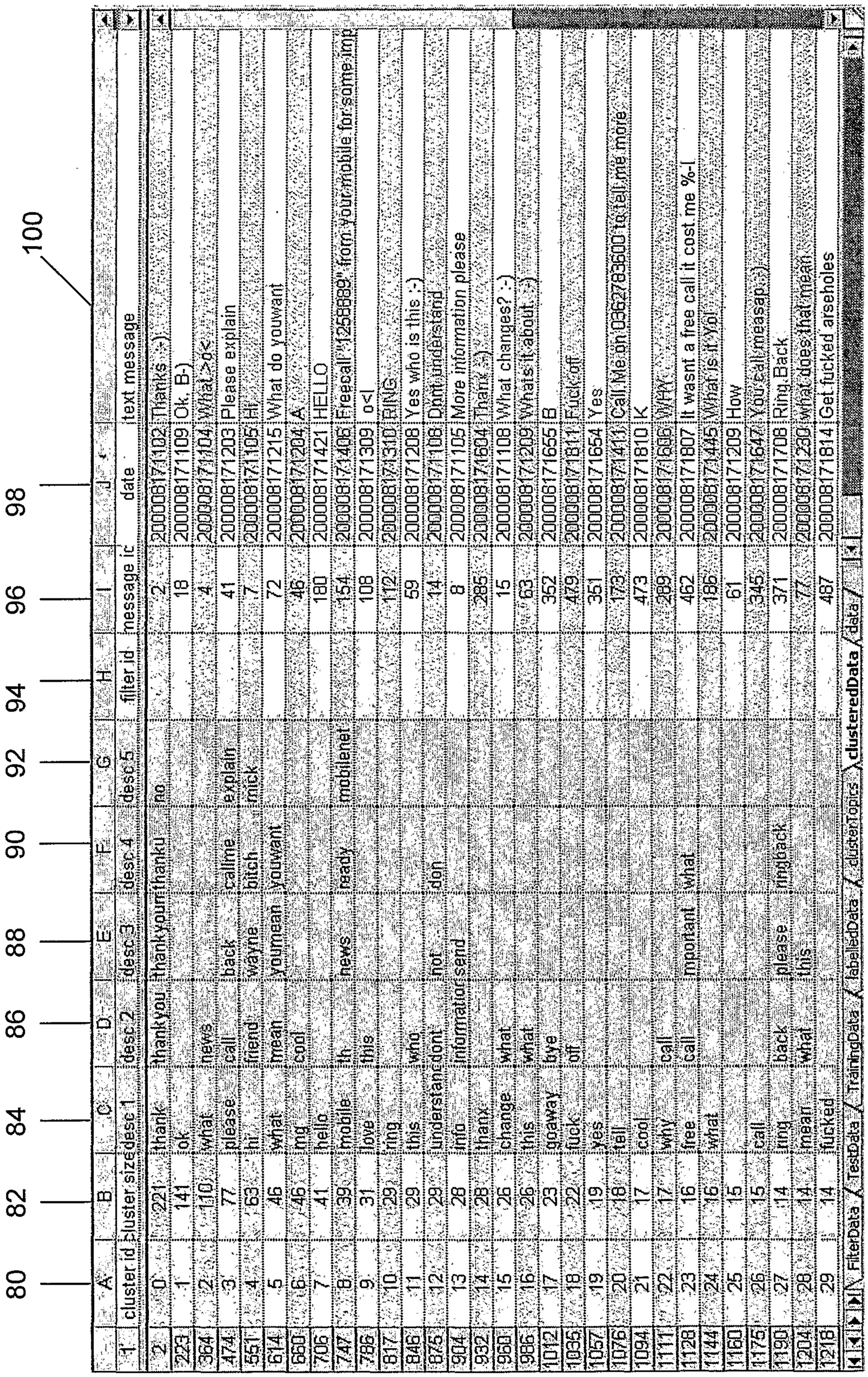
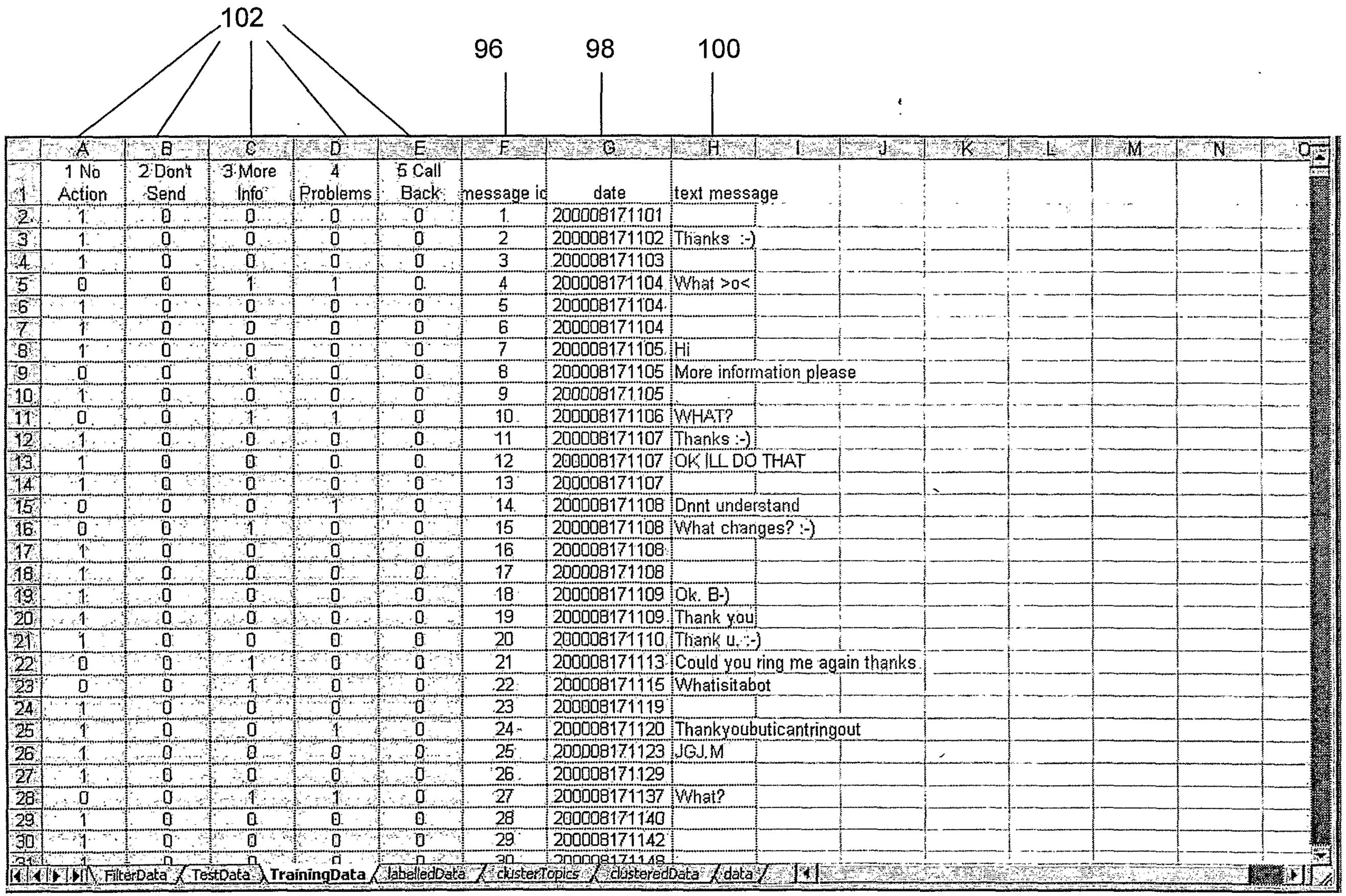

96
98
100
82
84
86
88
90
92
94
80
message ic
date
text message
cluster id
cluster size
desc 1
desc 2
desc 3
desc 4
desc 5
filter id
clusteredData
clusterTopics
labelledData
TrainingData
TestData
FilterData

Figure 6



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1 No Action	2 Don't Send	3 More Info	4 Problems	5 Call Back	message id	date	text message							
2	1	0	0	0	0	1	200008171101								
3	1	0	0	0	0	2	200008171102	Thanks :-)							
4	1	0	0	0	0	3	200008171103								
5	0	0	1	1	0	4	200008171104	What >o<							
6	1	0	0	0	0	5	200008171104								
7	1	0	0	0	0	6	200008171104								
8	1	0	0	0	0	7	200008171105	Hi							
9	0	0	1	0	0	8	200008171105	More information please							
10	1	0	0	0	0	9	200008171105								
11	0	0	1	1	0	10	200008171106	WHAT?							
12	1	0	0	0	0	11	200008171107	Thanks :-)							
13	1	0	0	0	0	12	200008171107	OK ILL DO THAT							
14	1	0	0	0	0	13	200008171107								
15	0	0	0	1	0	14	200008171108	Dnnt understand							
16	0	0	1	0	0	15	200008171108	What changes? :-)							
17	1	0	0	0	0	16	200008171108								
18	1	0	0	0	0	17	200008171108								
19	1	0	0	0	0	18	200008171109	Ok. B-)							
20	1	0	0	0	0	19	200008171109	Thank you							
21	1	0	0	0	0	20	200008171110	Thank u. :-)							
22	0	0	1	0	0	21	200008171113	Could you ring me again thanks							
23	0	0	1	0	0	22	200008171115	Whatisitabot							
24	1	0	0	0	0	23	200008171119								
25	1	0	0	1	0	24	200008171120	Thankyoubuticantringout							
26	1	0	0	0	0	25	200008171123	JGJ,M							
27	1	0	0	0	0	26	200008171129								
28	0	0	1	1	0	27	200008171137	What?							
29	1	0	0	0	0	28	200008171140								
30	1	0	0	0	0	29	200008171142								
31	1	0	0	0	0	30	200008171148								

FilterData | TestData | **TrainingData** | labelledData | clusterTopics | clusteredData | data

Figure 7

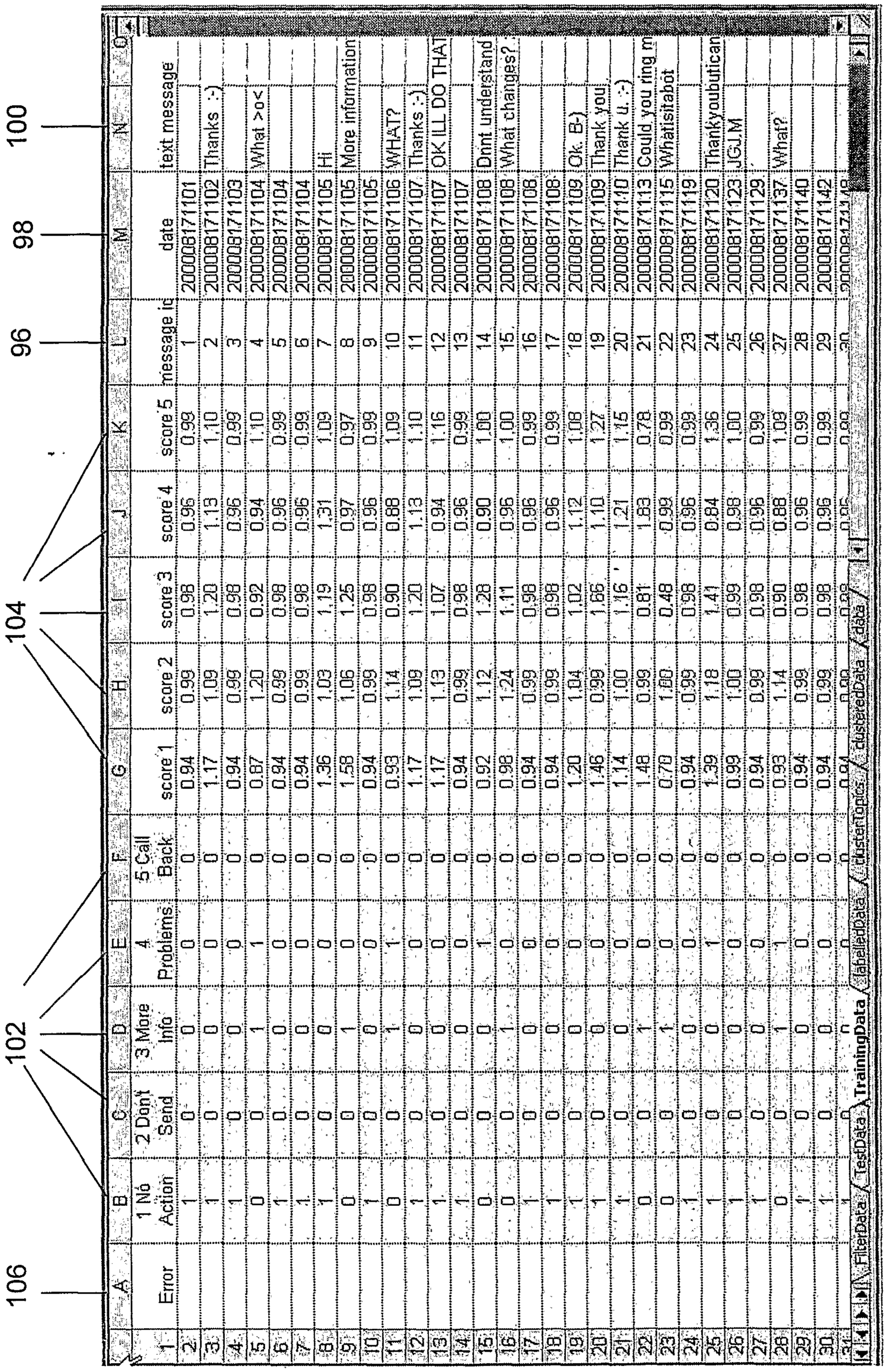

100
102
104
96
98
106
text message
date
score 1
score 2
score 3
score 4
score 5
message id
1 No Action
2 Don't Send
3 More Info
4 Problems
5 Call Back
Error
TestData
TrainingData
LabelledData
ClusterTopics
clusteredData
FilterData

Figure 8

104

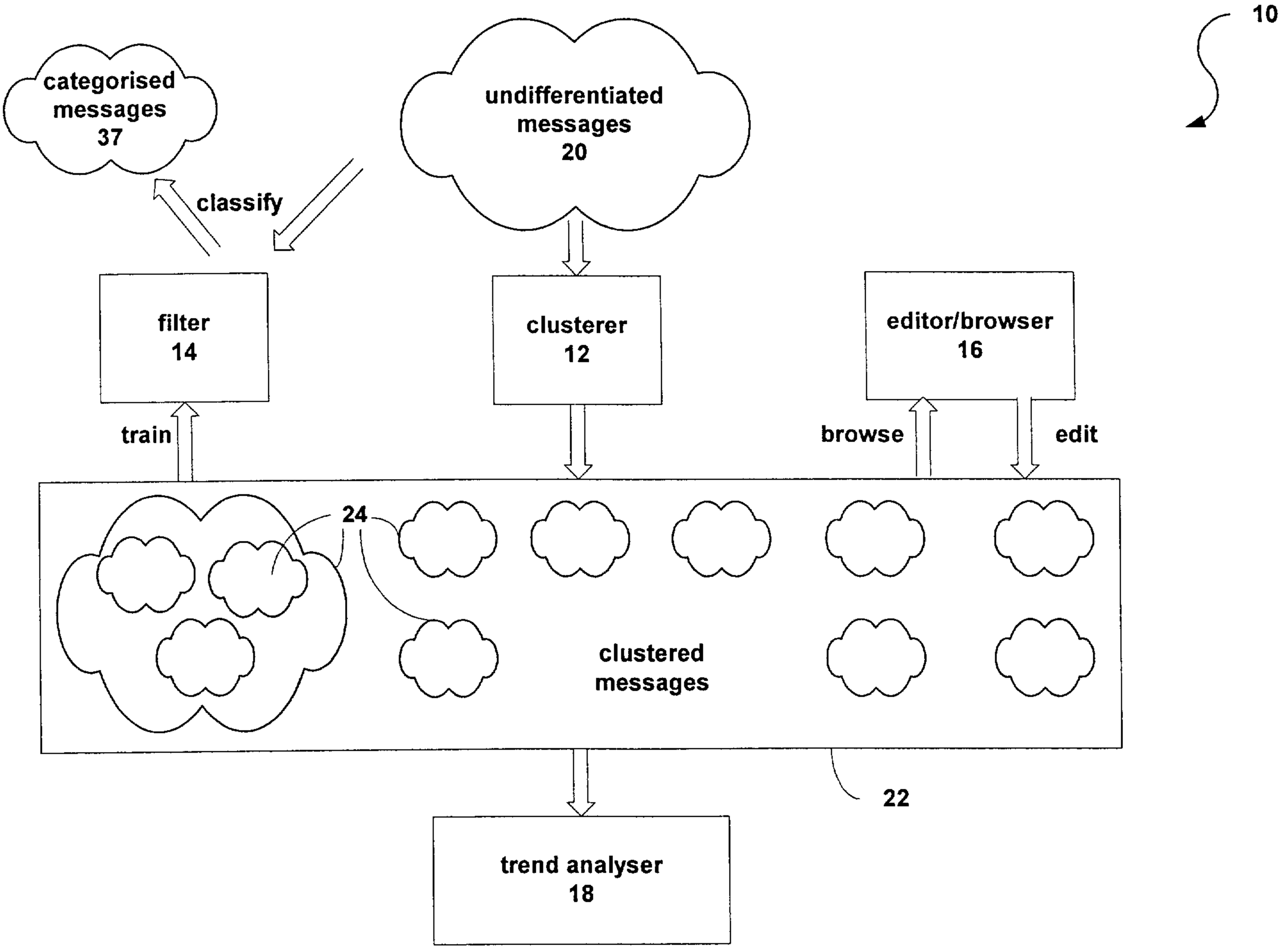
102

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	score 1	score 2	score 3	score 4	score 5	Label 1	Label 2	Label 3	Label 4	Label 5	message id	date	text message		
2	0.94	0.99	0.98	0.96	0.99	1					1910	200008191206	Whoisit		
3	1.16	1.03	1.07	1.07	1.04	1					1911	200008191207	OK		
4	0.98	0.99	1.01	0.13	1.00	1			1		1912	200008191207	Huh		
5	1.52	1.05	1.25	1.37	1.12	1					1913	200008191207	Hi from christine		
6	0.99	1.12	0.78	1.11	1.07	1					1914	200008191207	TO BE CONTINUED		
7	0.93	1.14	0.90	0.88	1.09			1	1		1915	200008191208	What		
8	1.46	0.99	1.66	1.10	1.27	1					1916	200008191208	Thank you!		
9	1.41	1.15	0.89	1.26	1.59			1			1917	200008191208	Can you send more infoin th		
10	1.26	1.28	0.91	1.12	1.34	1					1918	200008191208	GO TO WERE GIVE THE A		
11	1.26	1.28	0.91	1.12	1.34	1					1919	200008191208	GO TO WERE GIVE THE A		
12	0.96	1.04	1.04	0.99	0.99	1					1920	200008191209	:-)		
13	0.98	0.99	0.99	0.98	1.00	1					1921	200008191210	J		
14	0.25	1.00	0.49	1.07	1.26	1					1922	200008191210	Is that u tahlia		
15	0.94	0.99	0.98	0.96	0.99	1					1923	200008191210			
16	0.94	0.99	0.98	0.96	0.99	1					1924	200008191210			
17	1.38	0.72	0.89	0.77	1.10			1	1		1925	200008191211	What :-/		
18	0.05	0.86	0.62	1.18	0.69						1926	200008191211	U should of told me that 2 d		
19	1.61	1.14	1.33	1.11	1.19	1					1927	200008191212	Thankyou :-)		
20	2.23	1.01	1.16	2.55	1.91	1					1928	200008191212	Freecall "1258889" from you		
21	0.78	1.00	0.75	1.12	1.31	1		1	1		1929	200008191212	How much dosms message		
22	0.16	0.33	0.81	1.19	0.65						1930	200008191212	Suck me beautiful (0)		
23	0.94	1.02	1.05	0.88	1.07	1					1931	200008191212	No		
24	0.94	0.99	0.98	0.96	0.99	1					1932	200008191212			
25	1.89	0.89	1.12	1.32	0.74			1			1933	200008191213	Can you give me more Inforr		
26	1.20	0.90	1.28	1.25	1.10	1					1934	200008191213	Hi im mike sounds good :-)		
27	0.94	0.99	0.98	0.96	0.99	1					1935	200008191213			
28	0.94	0.99	0.98	0.96	0.99	1					1936	200008191213			
29	1.08	1.06	1.09	1.04	1.03	1					1937	200008191216	Thanx :-)		
30	0.72	0.70	0.99	0.97	1.01	1					1938	200008191216	Y		
31	1.25	1.58	2.90	1.84	1.48				1		1939	200008191217	Thanks for you msg i dont u		
32	0.96	1.04	1.04	0.99	0.99	1					1940	200008191218	:-)		

FilterData | TestData | TrainingData | labelledData | clusterTopics | clusteredData | data

Figure 9

WO 02/25479 PCT/AU01/01198
CA 02423033 2003-03-24
9/9


10
categorised messages 37
undifferentiated messages 20
classify
filter 14
clusterer 12
editor/browser 16
train
browse
edit
24
clustered messages
22
trend analyser 18