A task-oriented framework for evaluating theme detection systems: *A discussion paper.*

Fidelia Ibekwe-SanJuan

URSIDOC-SII
Enssib, 17 – 21, boulevard du 11 Novembre 1918

&
University of Lyon 3
ibekwe@univ-lyon3.fr

Abstract

This paper discusses the inherent difficulties in evaluating systems for theme detection. Such systems are based essentially on unsupervised clustering aiming to discover the underlying structure in a corpus of texts. As the structures are precisely unknown beforehand, it is difficult to devise a satisfactory evaluation protocol. Several problems are posed by cluster evaluation: determining the optimal number of clusters, cluster content evaluation, topology of the discovered structure. Each of these problems has been studied separately but some of the proposed metrics portray significant flaws. Moreover, no benchmark has been commonly agreed upon. Finally, it is necessary to distinguish between task-oriented and activity-oriented evaluation as the two frameworks imply different evaluation protocols. Possible solutions to the activity-oriented evaluation can be sought from the data and text mining communities.

1. Introduction

Discovering automatically the themes contained in a huge quantity of data is important for several knowledge-driven tasks like ontology population, information retrieval (Chalmers, 1993 ; Hearst & Pedersen, 1996), text mining (Berry, 2004 ; Castellanos, 2004 ; Feldman, 1995 ; Kontosthatis *et al*., 2004), text categorization (Lewis *et al*., 2004), science and technology watch (Schiffrin & Börner, 2004 ; Ibekwe-SanJuan & SanJuan, 2004), knowledge domain mapping (Chen, 2006). Various systems have been developed for this task both from the industry (Autonomy, Temis, SPSS, IBM, NetOwl, Clearforest) and the academia.

While the usefulness of methods for automatically discovering structures in data (exploratory data analysis) has been widely acknowledged, the results have often been criticised because there is no clear way of evaluating the quality of the discovered structures since they are precisely discovered, thus were unknown to the user.

The evaluation of such systems has remained a bottleneck issue because, in most cases, there is no way to determine an ideal number of topics, their precise contents (the text units composing them) and their relation to one another (the topological layout).

Evaluating descriptive algorithms that discover structures in texts is different from evaluating systems for text categorization (TC) or for topic detection and tracking (TdT). In TC, the goal is to assign texts to an existing hierarchy or taxonomy that has been manually designed. Thus the user has already determined the organisation of domain concepts and would like systems to automatically predict to which categories incoming texts belong. In the TC task, the object of the evaluation is quite clear and protocols can be quite straightforward because the answer key is known. Algorithms are evaluated on their ability to predict the right category for each text. A TC task is predictive by nature. Benchmarks for TC exist in the form of adapted test corpora with their answer key for instance the20 newsgroup[1] and the Reuters-21578 (Lewis, 2004) corpora. In these corpora, the texts have been assigned manually to categories and automatic algorithms are evaluated on their ability to reproduce the human categorization.

Topic detection and Tracking (TdT) was a well worn track in the TIDES[2] evaluation program sponsored by the DARPA. TdT (Allan *et al*., 1998) was mainly concerned with tracking the first occurrence of new events in streams of newspaper stories. It had five sub-tracks:

- *story segmentation* aimed at evaluating the ability of systems to detect changes between topically cohesive sections,
- *topic tracking* evaluated the ability of systems to keep track of stories similar to a set of example stories,
- *topic detection* evaluated the ability of systems to build clusters of stories that discussed the same topic,
- *first story detection* evaluated the ability of systems to detect the first occurrence of a new story,
- *link detection* evaluated the ability of systems to discover if two stories were topically linked.

By definition, TdT is incremental in nature, i.e., incoming news are taken into account with respect to already classified news. Although the *topic detection* track appears similar to the objective of theme detection systems considered here, it was conceived also as an incremental task in the TdT campaign. For a given source file of stories, the system could look ahead only by a specified number of 'days' before making a final decision[3]. In incremental clustering, the algorithm processes stories in a

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

[2] Translingual Information Detection, Extraction and Summarization.

[3] http://www.nist.gov/speech/tests/tdt/tasks/detect.htm

given time window. It compares each story to already formed cluster (if any) and decides whether to merge a story with an existing cluster (if their similarity exceeds a certain threshold) or to create a new cluster. It then looks forward to the next stories and iterates the whole procedure. Thus the *topic detection* track in the TdT campaign was inherently a classification task (once the first clusters have been formed). Systems had to make binary decisions. Evaluation metrics for classification or categorisation tasks have already been designed, especially when the answer key (*gold standard*) exists.

2. Descriptive data analysis

We are interested in the problem posed by the evaluation of descriptive data analysis systems, where the aim is precisely to discover the "important" topics in a corpus of texts, without relying on any prior knowledge. The main technique used for this type of task is clustering. Descriptive data analysis systems usually perform a global clustering (clustering over the entire data set) in a "once-and-for-all" approach. Also, such systems do not have the obligation of discovering only "new" topics. They aim to provide a synthetic and suggestive view of the entire data by proposing one or several partitions (clusters) of this data. Clustering methods addressing this task are based mostly on unsupervised approaches. The aim of clustering is to form groups of "similar" objects (Kaufman & Rousseeuw, 1990). Objects here can be text units (words, phrases, *n*-grams) or documents. Similarity is defined as the measure of shared contexts by two units. Context can be defined in any number of ways : co-occurrence in the same document, in the same paragraph, in a fixed text window. Few methods base clustering on shared linguistic contexts, either on identical grammatical function within noun phrases (Grefenstette, 1997) or on terminological variation relations (Ibekwe-SanJuan & SanJuan 2004).

As of now, there exists no consensus on how to evaluate such systems, no commonly accepted benchmarks and test corpora against which new systems can be evaluated. This is due to the numerous inherent difficulties in evaluating the output of such systems. Indeed, many questions arise, each of which has garnered a certain amount of research effort and has received varying answers. Among the major evaluation problems are the following:

1. *Determining the optimal number of clusters.* How can the optimal number of clusters be determined ? Based on what criteria ? The situation is further complicated when some algorithms (k-means) need the user to specify the number of clusters beforehand whereas other clustering algorithms do not require this a prior specification. In this case, evaluation is necessarily faced with differing number of clusters, a situation that is deemed risky and difficult in the literature (Yeung, 2001). To date, most studies on cluster evaluation have made as a pre-requisite that competing methods produced the same number of clusters (Yeung, 2001 ; Zamir & Etzioni, 1998). This could handicap some methods but the effect of this requirement on the performance of the algorithms has received little or no attention.
2. *Evaluating the clusters' content*: How can the well-formedness of clusters be determined ? How should cluster content evaluation be done when competing algorithms do not have exactly the same input ? Indeed some algorithms (numerical ones) work best on the "*bag-of-words*" approach. Other algorithms are tailored down for more complex units (terms, phrases, n-grams). Thus, an inherent difficulty in cluster content evaluation will be how to compare one-word units to longer units which may contain them (lexical subsumption). This may entail taking into account the grammatical role of the lone word in the longer term and perhaps weighting this relationship according to the role concerned. Take for instance three clusters *c1*, *c2*, *c3* produced by three competing algorithms A, B and C. Supposing that cluster *c1* was produced by a numerical method clustering only lone words. Thus cluster *c1* has only one-word units among which the word "*cancer*". The other two algorithms, B and C can cluster longer units. They have respectively the multi-word terms "*lung cancer*" in *c2* (method B) and "*cancer research*" in *c3* (method C). An evaluation metric focused on evaluating the similarity of cluster contents across the different methods has to determine which pair of clusters is closer. Intuitively, we would like to see clusters "*c1 – c2*" closer because in the two cases, "*cancer*" is the noun focus (the grammatical head of the phrase). This intuition has to be formalised as a measure which can be computed and generalised to take into account different situations. For instance, a more complex situation arises when the comparison concerns two multi-word terms sharing some common words. For instance, if cluster *c2* contains the term "*bran incorporation*", *c3* the term "*raisin incorporation*" and another cluster *c4* contains the term "*wheat bran incorporation*", which pair of clusters is closer ? In other words, can we formalise our linguistic intuitions based on lexical association or lexical subsumption in terms of semantic distance ? Does the substitution of a modifier word as in "*bran incorporation ↔ raisin incorporation*" induce less semantic distance than the addition of a modifier word as in "*bran incorporation → wheat bran incorporation*" ? Is this a good question to ask ? There is no easy answer and no clear way to perform this type of comparison. Here again, the current practice in the literature is to impose the same input on all competing methods in other to facilitate the evaluation but at the expense of penalising methods which work better either on the "bag-of-words" approach or on multi-word terms. In a recent experiment, SanJuan & Ibekwe-SanJuan (2006) compared their clustering system based on multi-word clustering by linguistic relations, to existing clustering algorithms based on statistical criteria (co-occurrence of words or phrases). In order to perform this comparison with the same input, the authors had to adapt the representation of multi-word terms to the vector space model needed by numerical algorithms (k-means and hierarchical clustering). This adaptation further led to a special term weighting scheme taking into consideration the grammatical function (head or modifier) of words in a

term and the position of each modifier word with regard to the head word. This weighting scheme enabled the authors to bridge the gap between the clustering principles used by the competing algorithms. Indeed, TermWatch, the authors system, clusters terms based on linguistic relations (variation relations) whereas numerical clustering methods cluster text units based on document co-occurrence information. It is not yet clear how this numerical encoding of internal term structure affected the performance of the algorithms and thus the evaluation metrics. This has to be investigated further.

3. *Topological structure of the clusters*. Given the uses of the results of descriptive data analysis, most clustering systems are also equipped with visualisation tools. Indeed, visualisation is often strategic for the goal of the clustering as it enables the user to understand the hidden patterns in the data (Chen, 2006). It could be a tool for strategic decision making. Visualisation tools enable the mapping of the discovered structure in a 2D or 3D space. Depending on the particular algorithm used, the exploration of the graphical results may be extremely complex to a layman. Information visualisation is a research area on its own and has numerous unsolved research issues like comprehensibility of the image, cognitive load, interpretation, to cite only issues from the user's point of view. These problems will have to be addressed if the evaluation is focused on the topology of the structure. They pose difficult challenges for evaluation among the problems enumerated so far.

4. *What metrics what purpose ?* In many areas of research on human-computer interaction, metrics have been designed for evaluating different aspects of the systems. Such metrics exist for evaluating clustering output but each metric focuses on a particular aspect. Some metrics come from neighbouring research fields like the "precision and recall measure in the information retrieval field, the "mutual information" measure in the lexical linguistics. Other probabilistic measures have also been applied to cluster evaluation. There is need to classify these metrics according to their uses for cluster evaluation, following the exact type of evaluation problem addressed and the objective of the metric.

Below, we recall some existing methods and measures for cluster evaluation and suggest ways to move forward by embedding cluster evaluation in a task-oriented framework.

3. Cluster evaluation: state-of-the-art

Cluster evaluation generally falls under one of these two frameworks: evaluation of the quality of the partitions vis-à-vis some internal cluster properties (intrinsic evaluation); task-oriented evaluation where the clustering output is embedded in an application or confronted to a real world truth like the existence of a gold standard (extrinsic evaluation).

3.1. Intrinsic cluster evaluation

This is also called “internal criteria”. Internal criteria is used to measure the intrinsic quality of the clusters especially in the absence of an external ideal partition (extrinsic evaluation). The focus of the evaluation is on some internal properties of the clusters which judge their mathematical or statistical well-formedness. Such measures are for instance *cluster separation* (isolation) and *cluster homogeneity* (compactness). *Cluster separation* evaluates how well the content of a cluster is separated or different from another. It tests the ability of a method to form groups of similar objects, which is the principle of clustering, thus its ability to maximise the intra-cluster similarity while minimising the inter-cluster similarity.

Other intrinsic measures may try to determine the *optimal number of clusters* or measure the *stability of the partitions* with respect to sub-sampling (Ben-Hur, 2002).

Intrinsic cluster evaluation does not address the usefulness of the clusters with regard to a real application or task. They do not tackle the question “Are the clusters any good for the task which the user is performing ?" Intrinsic evaluation is concerned with the statistical and mathematical properties of the algorithm and consequently of its output.

Intrinsic criteria has as advantage the fact that it can bypass the necessity of having an external ideal solution but its major inconvenience is that evaluation is based on the same information from which the clusters were derived. Studies on intrinsic cluster evaluation are quite useful in themselves in that they drive the improvement of clustering algorithms but they do not satisfy entirely the evaluation paradigm. Some measures of the actual usefulness of the clustering output is needed.

3.2. Extrinsic cluster evaluation

Two evaluation methodologies can be distinguished in this framework: evaluation embedded in a particular application or evaluation against a target partition (*gold standard*). Needless to say that the second methodology is the easier of the two to set up.

3.2.1 Evaluation against a target partition

In this case, there exists many metrics allowing the comparison of cluster contents[4] across different methods with a target partition (Milligan & Cooper, 1985 ; Jain & Dubes, 1988). One of the oldest measures used is the "Rand Index" and its enhanced version, the "Adjusted Rand index" (Hubert & Arabie, 1985). The Rand index measures the degree of agreement between two partitions. However, it was shown that this measure, in its original form, had some flaws. For instance, the expected value of the Rand index of two random partitions does not take a constant value like zero (Yeung 2001). Also, Pantel & Lin (2002) observed that computing the degree of agreements and disagreements between the proposed partitions and the target partition could lead to unintuitive results. For instance, if the target partition has 20 equally-sized

[4]This is the only aspect evaluated.

clusters with 1000 elements each, treating each element as its own cluster will lead to a misleading high score of 95%. Milligan & Cooper (1985) recommended the use of Adjusted Rand index even when comparing clusters at different levels of the hierarchy. However, SanJuan & Ibekwe-SanJuan (2006) also observed that the Rand index and the adjusted Rand Index had the following flaws:

- they are computationally expensive since they require $|\Omega|^2$ comparisons which is problematic when $|\Omega|$ is large,
- they are too sensitive to the number of clusters when comparing clustering outputs of different size (Wehrens et al., 2003),
- the adjusted Rand Index supposes the generalized hypergeometric distribution as the model to ensure that two random partitions do take a constant null value. This type of distribution is not always fitted by that of items in the target partition.

Denoeud (2005) tested the ability of different measures in determining the distance between two partitions. The Jaccard measure appeared as the best for this task as it does not have the drawbacks of the Rand index.
This measure computes the number of pair of items clustered together by two algorithms divided by the total number of pairs clustered by one of the algorithms. However, it cannot take into account the specificities of a target distribution. For instance, in the case where the target partition has a very big class, the Jaccard measure will favour clustering algorithms that detect this big class against algorithms that try to fit the distribution of the smaller classes of the target partition (SanJuan & Ibekwe-SanJuan, 2006).

Pantel & Lin (2002) proposed the use of the editing distance to evaluate the distance between the proposed partitions and the target partition. The idea is to evaluate the cost of producing the target partition from the proposed ones. The editing distance is an old notion used to calculate the cost of elementary actions like "*copy, merge, move, delete*" needed to obtain one word (or phrase or sentence) from another. Here, the authors applied it to cluster contents and chose to consider three elementary actions: *copy, merge, move*. Their measure is formulated thus :
If *C* is a set of clusters produced by a clustering algorithm and *A* the clusters of the *gold standard,* an editing distance *dist(C, A)* is the number of operations required to transform *C* into *A*. Three types of operations are considered :

1. move an item from one cluster to another,
2. copy an item from one cluster to another
3. merge two clusters

If *B* is a base classification where every item forms a cluster, then the quality of a cluster can be calculated as :

$$1 - \left(dist\,(C,A) \,/\, dist\,(B,A)\right)$$ (Pantel & Lin, 2002)

The copy action allows to consider fuzzy clustering where clusters overlap (an element can belong to more than one cluster). Pantel and Lin's measure contains some deterministic behaviour with some inherent bias. To measure the distance between a clustering output and an ideal partition, these authors considered the minimal number of merges and moves that have to be applied to a clustering output in order to obtain the target partition. Considering two trivial partitions : one where all the items are in one cluster (complete partition) and the other where every item is its own cluster singletons (discrete partition), Pantel and Lin's measure supposes that the two trivial clusterings are at equi-distance from the target partition. Upon verification, this turned out not to be true. Their measure favours the complete trivial partition over the discrete one, therefore it favours algorithms that form fewer clusters, even of poor quality. A corrected version of this measure can be found in SanJuan & Ibekwe-SanJuan (2006). This study has brought to light the fact that the distribution of elements in the target partition can have possible influences on the metric used for evaluation and thus on the performance of certain algorithms. Thus, this aspect has to be taken into consideration when choosing metrics to evaluate systems against a gold standard.

Finally, as Yeung & Ruzzo (2001) observed, external criteria has the advantage of providing an independent unbiased assessment of the cluster but has as inconvenience the fact that they are hardly available. This is because of the labour-intensive nature of building a gold standard. Moreover, in the case of discovering data structures from a corpus of texts, there is no easy way of deciding what is a good cluster or which elements should belong to a particular cluster. Thus in practice, in many cases where unsupervised clustering is used, there will not be a gold standard for evaluating the quality of the output.

3.2.2 Task and activity-embedded evaluation

This framework addresses the crucial question of the usefulness of the discovered structures in a real world application. Evaluation is then embedded in the context of a task or an application like information retrieval (IR), question-answering (Q-A), science and technology watch (STW), text mining (TM). We first need to distinguish between two possible evaluation objects: *task* and *activity (application)*. This distinction is necessary because evaluation is not carried out on the same things and not in the same way.

In a task-oriented framework, the object of the evaluation is to measure the performance of the systems in accomplishing a specific task (information retrieval, question-answering). What is being evaluated here is ability of independent clusters in helping the user accomplish his information seeking need. In the case of IR, the user would retrieve relevant documents thanks to the best cluster. In the case of Q-A, the user would hopefully be provided with the best answer to his question thanks to one or two clusters if the Q-A systems relies on clustering. A task-oriented framework evaluates simply part of the system's output, not necessarily all of it. The evaluation usually spans a short time, usually the time needed to run the system. A task-oriented evaluating can be carried out in an objective manner by applying the metrics used in the domain of the task. For instance, in an IR task, evaluation will be based on the precision-recall metric used for evaluating IR systems.

In an activity-oriented framework, evaluation is not only of the system's output but of its overall capacity in assisting a user in the course of his activity. An activity in this sense can be business intelligence, customer relation management (CRM) or STW, TM. In a STW activity for instance, the focus is on discovering the hidden structures in a set of texts and thereby identifying important topics and trend. The system can be evaluated at different stages of this activity. An activity-oriented evaluation is necessarily complex and spans a reasonable length of time (days or a week). It requires the close involvement of human actors in the evaluation process. Evaluation is usually of the entire system's output: cluster content (quality, homogeneity), number (recall) and topological structure (what interpretations can be made on the links between clusters ?). Elaborating an evaluation protocol generally involves a panel of users and is usually quite difficult to achieve in a systematic and satisfactory way. There are numerous biases and arbitrary judgments on several points (agreeing on existing clusters, their contents, their topological structure). Such an evaluation framework, because it involves a lot of human participation, aside from being fraught with problems, is not reproducible from one experiment to another. Yet, it remains the only framework that can address the "real-word usefulness" requirement.

4. Conclusion

The "Knowledge discovery in databases" (KDD) and data mining community has been addressing the "real-world usefulness" of results from data mining systems. Because of their obvious relationship, the "Knowledge Discovery in Texts" (KDT) and text mining community is also grappling with the same issue. KDD is defined as "*the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.* [...] *The discovered patterns should be valid on new data with some degree of certainty. We also want patterns to be novel (at least to the system and preferably to the user) and potentially useful, that is, lead to some benefit to the user or task. Finally, the patterns should be understandable, if not immediately then after some postprocessing*" (Fayyad, Piatetsky-Shapiro, and Smyth, 1996).

These requirements have led the KDD community to define quantitative measures for evaluating the discovered patterns from data mining systems. Such measures of usefulness include "*measures of certainty"* or "*measures of utility"*. A certainty measure evaluates the estimated prediction accuracy of the system in the face of new data. It can be applied to classification or categorisation tasks (decision tasks). Measures of utility concern for example the actual economic gain (in terms of money saved) resulting from the systems (accurate predictions or speed of response). The KDD process requires the discovered patterns to be novel and understandable for the user. However, novelty and understandability are much more subjective notions, thus difficult to measure. Ultimately, the measure of understandability of a pattern lies with the user. Likewise, the degree of novelty of discovered patterns can only be judged by the user. The system has no way of knowing if a pattern is already known by a user or not. All these requirements are gathered in a unique measure important to the KDD community, called *interestingness*. Interestingness gives "an overall measure of pattern value, combining validity, novelty, usefulness, and simplicity' (Fayyad, Piatetsky-Shapiro, and Smyth, 1996). Thus, a *pattern* discovered by a system is deemed to be knowledge if it "exceeds some interestingness threshold". This definition of "knowledge is entirely user oriented and domain specific and is determined by whatever functions and thresholds the user chooses." (Fayyad, Piatetsky-Shapiro, and Smyth 1996).

Since the KDD and KDT processes make heavy use of clustering as one of the mining techniques, cluster evaluation could look to these communities for solutions to the activity-embedded evaluation problem. Since these measures are steeped deep into the user's appreciation of the results, if effective, they can be a step forward in evaluating the output of unsupervised approaches for theme detection.

5. References

Allan J., Carbonell J., Doddington G., Yamron J., Yang Y. (1998). "*Topic Detection and Tracking Pilot Study: Final Report*." Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. 194-218.

Berry M.W (eds.) Survey of Text Mining. Clustering, classification and retrieval, Springer, NY, 2004, 244.

Castellanos M. (2004). HotMiner : Discovering hot topics from dirty texts, in Berry M.W (eds.) Survey of Text Mining. Clustering, classification and retrieval, Springer, NY, 2004, 123-157.

Chalmers M. Using a landscape metaphor to represent a corpus of documents. In *Spatial Information theory*, Frank A., Caspari I. (eds.), Springer Verlag LNCS 716, 1993, 377-390.

Chen C. (2006). CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57(3), pp. 359-377.

Cutting D. R., Karger D. R., Pedersen J., Tukey J. W. (1992). Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. Proceedings of the 15th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, 318-329.

Fayyad U., Piatetsky-Shapiro G., Smyth P. (1996). "From data mining to knowledge discovery in databases", *AI Magazine*, fall (1) , 37-54, 1996.

Feldman R., Fresko M., Kinar Y. (1998). Text mining at the term level. *In* Zytkow, J. M., Quafafou, M. (eds.), Principles of Datamining and knowledge discovery. Proceedings of the 2nd European symposium PKDD. Berlin-Springer, Nantes - France, 1998, 65-73.

Hearst M. A., Pedersen J. O. (1996). Re-examining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. 19th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, 18-22 August, ACM, 76-84.

Grefenstette G. (1997). SQLET:Short Query Linguistic Expansion Techniques, Palliating One-Word Queries by

Providing Intermediate Structure to Text. In *Proceedings of "Recherche d'Information assistée par ordinateur" (RIAO)*, pp. 500-509, 1997.

Hubert L., Arabie P. (1985) Comparing partitions. *Journal of Classification*, 193–218.

Ibekwe-SanJuan F., SanJuan E. (2004). Mining textual data through term variant clustering: the termwatch system. Proceedings of the Conference « Recherche d'Information assistée par ordinateur » (RIAO). Avignon", 2004, 487-503.

Jain A. K., Dubes R. C. (1998) Algorithms for Clustering Data, Prentice Hall.

Kaufman L., Rousseeuw P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. A Wiley-Interscience Publication.

Kodratoff Y., "Knowledge discovery in texts: A definition and applications", *In* Ras & Skowron (eds.) *Foundation of Intelligent systems*, Lecture Notes in Artificial Intelligence, n° 1609, Springer-Verlag, p. 16-29, 1999.

Kontostathis A., Galitsky L.M., Pottenger W.M., Roy S., Phelps D.J. (2004). A survey of emerging trend detection in textual data mining, *In* Berry M.W (eds.) Survey of Text Mining. Clustering, classification and retrieval, Springer, NY, 2004, 185-239.

Lewis D., Yang Y., Rose T.G, Li F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research, *Journal of Machine Learning Research*, 2004, 5, 361-397.

Milligan G.W., Cooper M.C. (1986). A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioural Research*, 21, 441-458.

Pantel P., Lin D. (2002). Document clustering with committees. Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'02, Tampere, Finland, 199-206.

SanJuan & Ibekwe-SanJuan (2006). Text Mining without document context, *Information Processing & Management*, special issue on Informetrics, To appear 2006.

Schiffrin R., Börner K. (2004). Mapping knowledge domains. *Publication of the National Academy of Science* (PNAS), 2004, 101(1), 5183-5185.

Yeung, K, Y., Ruzzo W, L. (2001). Details of the Adjusted Rand Index and clustering algorithms. Supplement to the paper "An experimental study on Principal Component Analysis for clustering gene expression data", *Bioinformatics,* 2001, 17, 763-774.

Zamir O., Etzioni O. (1998). Web document Clustering, A feasibility demonstration. *In ACM SIGIR Conference on Research and Development in Information Retrieval,* 1998, 46-54.