A Study on the Role of Similarity Measures in Visual Text Analytics

Frizzi San Roman, Rosane Minghim and Maria Cristina F. de Oliveira
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Brasil
frizzi.sanroman@ucsp.edu.pe, rminghim@icmc.usp.br, cristina@icmc.usp.br

***Abstract*—Text Analytics is essential for a large number of applications and good approaches to obtain visual mappings of text are paramount. Many visualization techniques, such as similarity based point placement layouts, have proved useful to support visual analysis of documents. However, they are sensitive to data quality, which, in turn, relies on a critical preprocessing step that involves text 'cleaning' and in some cases term detecting and weighting, as well as the definition of a similarity function. There has been limited discussion on the effect of these important similarity calculations in the quality of visual representations. In this work we studied the effect of different text similarity measurements on the quality of visual text mappings. We focused mainly on two types of distance functions, those based on the well-known text vector representation and on direct string comparison measurements, comparing their effect on visual mappings obtained with point placement techniques. We find that both have their value but, in many circumstances, the recently introduced incremental vector space model (iVSM) is the best solution when discrimination is important. Based on the results of our evaluation we offer recommendations on the application of different text similarity measurements for Visual Text Analytics tasks.**

***Keywords*-Visual Text Analytics, Visual Text Mining, Vector Space Model, High-dimensional Data Visualization, Multidimensional Projections.**

I. Introduction

Producing visualizations from textual documents requires a pre-processing step in which similarity evaluation plays a fundamental role. Often, a Vector Space Model (VSM) [1] that considers the frequency of relevant words is created, over which cosine distance approximates text dissimilarity. Little is known about how this pre-processing affects the outcome of text visualization techniques.

The VSM poses many limitations for visualization purposes, as it fails to capture semantics implicit in the relationships among words and terms. Moreover, in building a meaningful VSM several pre-processing operations require parameter settings that may affect the outcome considerably. Resulting models are typically described by very high-dimensional feature spaces, which suffer from drawbacks globally referred to as 'the curse of dimensionality' [2] that result in low discrimination power by most techniques.

*M.Sc. Dissertation. The authors acknowledge the support of FAPESP (2010/12294-0 and 2010/03100-8) and CNPq (305079/2009-3 and 133874/2010-9).

VSM models may be avoided altogether by using direct string comparison functions [3]. Adding documents to a collection does not impact the underlying model, since it suffices to compare the new document with the existing ones. Many such measures have been defined, for different purposes and applications. Again, there is little record on how their choice affects text analytics, visual or otherwise, and the question remains on how they compare with cosine distances calculated over the VSM.

In this work we focused on assessing how the choice of a (dis)similarity function affects the output of content-based visualization techniques. We consider visualizations that lay out documents as points on a plane based on their similarity, to verify how the choice of a similarity function affects their quality in terms of discriminating groups of text files with highly related content. This work investigates these issues, reporting on the following questions:

1) are string distance measures suitable for text visualizations based on similarity? which measures may be considered and how their choice affects the visualizations?
2) how do string distances compare with the traditional cosine distance computed over the VSM and iVSM regarding visualization quality?

II. Related Work

The VSM with *tf-idf* measure of terms deemed relevant is the typical input representation to most text visualization and text clustering techniques. Visualizations may be derived directly from such representations, e.g. as in various *Multidimensional Scaling* (MDS) approaches [4], [5], [6]. Hierarchical similarity-based layouts have also been proposed and illustrated for visualizing textual documents, e.g. the Neighbor-Joining tree [7].

The *Incremental Board - incBoard* [8] and the *Incremental Space* [9] also produces visualizations of text collections. They are, by design, more suited for handling dynamic collections to which documents are added gradually. These techniques inspired the *Incremental Vector Space Model* (iVSM) introduced and described in [10], [11], [12]. The iVSM has been proposed to represent text documents of an incremental collection as vectors in a high-dimensional space. As in the original VSM, each dimension represents the *tf-idf* frequency of a relevant term in the collection. As the documents in a dynamic collection are not known, *a priori*, an initial representation of the unknown collection is approximated from the VSM

constructed for a similar known collection (e.g., news, or scientific papers). This approximate initial representation is called a 'language model', and it provides the relevant terms in the collection, their frequency and the number of documents in which they occur. The iVSM model is constructed by continuously updating the language model, updating the term frequency and document frequency counting of terms, as new documents are added to the collection (or existing documents are removed).

Alternatively, vector models may be derived with topic extraction techniques such as *Latent Semantic Analysis* [13] and *Latent Direchlet Allocation* (LDA) [14], usually producing lower-dimensional feature spaces. Topics are also often extracted to annotate similarity-based visualizations, based, for instance, on LDA [15] or on association rule mining [16] to derive topic-oriented views.

We are unaware of previous studies on how the choice of the similarity function affects the outcome of text visualizations. There are, however, studies that report comparisons of string distance functions in other application domains. Cohen et al. [17] compare the performance of several distance metrics for the tasks of matching and clustering lists of entity names. *SecondString* is an open-source Java toolkit that incorporates several string metrics for matching names and records, including some novel hybrids of well-known methods. Authors computed three evaluation measures, the non-interpolated average precision, the maximum F1 score and the interpolated precision at eleven recall levels. In general, the best results were obtained with the hybrid distances proposed by them.

Kempken et al. [18] compare the performance of selected distances to support retrieval of historical spelling variants in historical text documents. Experiments were conducted on a dataset of historical spellings manually collected from historical German documents, containing a list of word pairs. Distances were evaluated with the precision and recall measures, and the best performance was obtained with a stochastic distance.

III. Background: String Similarity Measures

String distance functions map a pair of strings X and Y to a real number r, where higher values of r indicate greater dissimilarity between X and Y. Opposite to distance functions, string similarity functions return higher values for r as X and Y are more similar. Pairwise distances may be generated from such functions taking the value $1 - r$ as measure of dissimilarity. In this section we introduce some string distance and similarity functions employed in this study.

One important class of string distance functions are the so-called *edit distances*, which return the minimum number of editing operations required to transform a string into the other. Typical editing operations are character insertion, deletion and substitution, and each one is assigned a cost. Some examples are, e.g., the Levenshtein distance [19] that assigns a unit cost to all edit operations, and the Needleman-Wunsch metric [20] that allows for variable adjustment of the gap cost, i.e., insert/delete operations and variable cost of substitutions. Two strings X and Y may also be considered as multisets of words (substrings or tokens), and several token-based measures are defined, as described in Table I. In Section V we compare these and other distance measures in generating (dis)similarity-based visualizations of text collections.

TABLE I
TOKEN-BASED MEASURES. FUNCTION $Q(\cdot)$ RETURNS THE NUMBER OF TOKENS IN THE INPUT STRING, $P(\cdot)$ RETURNS THE NUMBER OF CHARACTERS, $qG(\cdot)$ RETURNS THE NUMBER OF SUBSTRINGS OF LENGTH q, XY STANDS FOR A CONCATENATION OF X AND Y, AND $C(\cdot)$ RETURNS THE SIZE, IN BYTES, OF THE COMPRESSED INPUT STRING.

Name	Similarity	
Dice's Coefficient	$\frac{2 * Q(X' \cap Y')}{Q(X') + Q(Y')}$	(1)
Cosine	$\frac{Q(X' \cap Y')}{\sqrt{Q(X') * Q(Y')}}$	(2)
Matching Coefficient	$\frac{Q(X' \cap Y')}{\max\{Q(X'), Q(Y')\}}$	(3)
Overlap Coefficient	$\frac{P(X' \cap Y')}{\min\{P(X'), P(Y')\}}$	(4)
Q-gram	$\frac{2 * qG(X' \cap Y')}{qG(X') + qG(Y')}$	(5)
NCD	$\frac{C(XY) - min\{C(X), C(Y)\}}{max\{C(X), C(Y)\}}$	(6)
NCDs	$NCD(X,Y) + \frac{NCD(X,X) + NCD(Y,Y)}{2}$	(7)

IV. Methodology: Study Set-Up

The goal of our work was to investigate how the different string comparison functions affect the quality of layouts output by point-placement techniques applied to textual collections. As far as visualizations are concerned, assessing quality of point-placement layouts is a difficult issue, as an analysis depends on the tasks the layout is meant to support. We believe important tasks are related with the layout's capability of preserving meaningful text clusters, i.e., to which extent it favors data grouping and group segregation; alternatively data analysts may desire layouts capable of preserving the original distances, i.e., the original dissimilarity relationships, as much as possible.

Some objective quality measures may be applied to compare different layouts in this context. We consider the *Silhouette Coefficient* [21], that attempts to quantify the quality of clusters identifiable in the high-dimensional text space or in a layout derived from it, and the *Neighborhood Hit* curve [5], which attempts to quantify to which extent a layout preserves known text classes.

- The *silhouette coefficient* SC of a cluster is computed as the average of the silhouette coefficient computed for its individual points. The silhouette of a particular data point p_i, belonging to a cluster C_i is computed according to Equation 8:

$$SC_{p_i} = \frac{(b_i - a_i)}{max(a_i, b_i)} \tag{8}$$

  where a_i is the average distance from p_i to all the other data points in C_i and b_i is the minimum average distance from p_i to the other clusters, obtained after computing the average distance from p_i to all the data points in a

cluster C_j, for all $j \neq i$. The silhouette coefficient takes values in the range $[-1, 1]$. Negative values indicate that a_i is greater than b_i, whereas the opposite is desirable – notice that SC assumes its maximum value, equal to 1, when $a_i = 0$

- The *Neighborhood Hit* (NH) measure is a graph that conveys information on the capability of the layout to preserve the class structure of the data. The NH value for an individual data point is computed by counting number of neighbors on the projected layout that belong to its same label or class. The graph is obtained by averaging the NH measure computed for all individual data points in the layout, for a varying number of neighbors to the point, from 1 to a maximum.

We compared layouts obtained with two point-placement techniques representative of distinct approaches to generating layouts of multidimensional data points, namely the *Least Square Projection* [5] and the *Neighbor-Joining tree* [7].

LSP attempts to generate a layout that preserves neighborhood groupings in the feature space. It first obtains a subsample of the data points, called control points, that is hopefully representative of its overall spatial distribution, and then computes neighborhoods for this sample points. The control points are projected first with a precise technique, and their projected coordinates, plus the neighborhoods, provide information to build a linear system model that is solved to obtain the projected coordinates of all data points. LSP takes as input parameters a pairwise distance matrix computed for the collection, the number of control points, and the number of neighbors to consider in defining neighborhoods.

The Neighbor-Joining (NJ) tree is inspired on algorithms for building phylogenetic trees in Biology. It builds a tree that describes ancestrality relations between species, given a matrix of pairwise distances between them. Then, a tree layout algorithm is employed to display the resulting hierarchy. NJ takes as input a pairwise distance matrix of the collection and requires no additional parameters. Whereas LSP shows a global view that attempts to convey meaningful groups of texts that have similar content, the branches and sub-branches in the tree view allow a user to infer levels or degrees of similarity between the texts.

We conducted studies with textual data sets from scientific papers and from news articles, their content is summarized in Table II.

TABLE II
TEXT DATA SETS

Name	Description	General Content	# docs	# classes
CBR-ILP-IR	case based reasoning, inductive logic programming and information retrieval	scientific papers	574	3
news2011	RSS news feeds (AP, CNN, Reuters and BBC)	news	1,771	23
ReutersNews	subset from Reuters21578 collection (newswire articles)	news	3,988	7

We computed 15 distinct pairwise dissimilarity matrices for the above datasets, using the following string distance or similarity functions[1]: Block, Jaccard, Cosine, Euclidean, JaroWrinkler, Dice Coefficient, Levenshtein, Matching Coefficient, SmityWaterman, Jaro, QGram, Soundex, NeedlemanWunch, Monge and Overlap Coefficient. The choice of these particular functions was based on a survey of existing alternatives for string comparison.

After inputting the distance matrices to LSP (considering two distinct configurations for the number of control points and neighborhood size) and to the NJ-tree, resulting layouts were compared to identify the functions with the best results on the CBR-ILP-IR data, by conducting a subjective evaluation of the visual quality of the resulting layouts and also comparing their corresponding NH curves. Based on this preliminary analysis, the five best performing string measures were further investigated, namely Cosine Similarity, Dice's Coefficient, Matching Coefficient, Overlapping Coefficient and QGram.

In all cases some text-preprocessing has been applied, which varied depending on the test case, due to the nature and objective of differente preprocessing measures. Luhn's cutting thresholds, stopwords removal and Porter stemming were employed when appropriate. For each case, the Results section specifies the pre-processing steps taken.

In a subsequent step, we compared the five previous string measures, plus *Normalized Compression Distance* (NCDs) [3], with the conventional approach for generating similarity-based layouts from text, namely the Cosine similarity applied over a VSM vector representation of the collection. Finally, we also included in the comparison the Cosine similarity applied over iVSM. Precision results are shown in the following Section, processing times are given in Table III.

TABLE III
PROCESSING TIMES, IN SECONDS, FOR COMPUTING THE DISSIMILARITY MATRIX, FOR THE DIFFERENT STRING DISSIMILARITY FUNCTIONS

Measure	CBR-ILP-IR	News2011	ReutersNews
Cosine Distance	750	41	2,331
Dice's coefficient	715	41	2,344
Matching's coefficient	1,588	73	4,761
Overlap's coefficient	758	41	2,319
Qgram Distance	16,744	1,215	52,877
NCDs	1,350	10,038	63,109

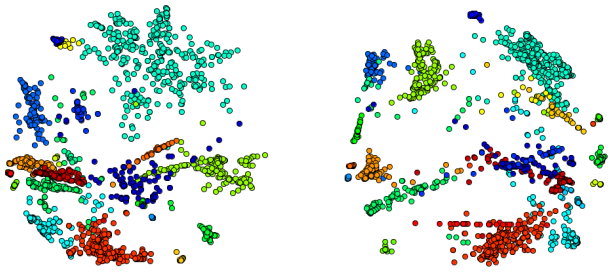
V. RESULTS

Due to space limitations, in this paper we only describe in detail the results relative to one of the data sets studies, namely News2011. The reader is referred to [11], [12] for additional details and information.

Fig. 1 and Fig. 2 show the layouts obtained with LSP and with NJ using as input dissimilarity matrices computed employing the cosine distance over the VSM and iVSM representations, respectively. The LSP input parameters (Fig. 1) were set to 177 control points and to 15 nearest-neighbors. Fig. 2 shows the corresponding NJ tree layouts, created with the NJ implementation by [22], which is faster than the

[1] http://sourceforge.net/projects/simmetrics

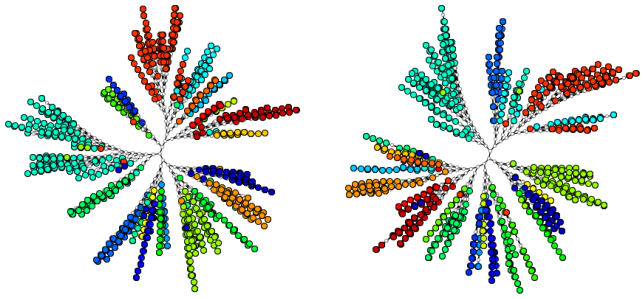
original one [7][2]. In the visualizations each circle represents a document and color maps the document class. One may visually assess the degree of class separation inspecting the spatial distribution of colors in the LSP layouts, or the distribution of colors in the branches and sub-branches of the NJ-tree layouts.



(a) VSM (b) iVSM

Fig. 1. LSP layouts generated for News2011 text datasets, using the VSM and iVSM representations and the cosine distance. Circle color maps document class.

In order to generate the visualizations, textual data has been preprocessed with stopwords removal, Porter's stemming and definition of Luhn's thresholds. We removed the usual stopwords, such as articles and prepositions, and also a few domain specific words when handling scientific papers or news, e.g., for papers added stopwords included 'press', 'proceedings', 'proc', 'vol' and 'year'. In generating the VSM models we set Luhn's lower cut to 10, and applied no upper cut threshold. For News2011, the starting language model has been computed from an existing collection with news from April 2006 (AP_BBC_CNN_Reuters.zip), available at the same site as the data set.

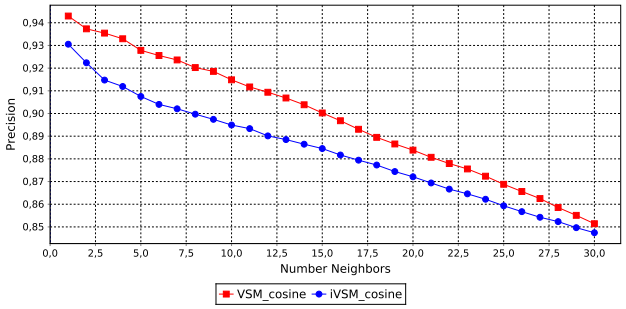


(a) VSM (b) iVSM

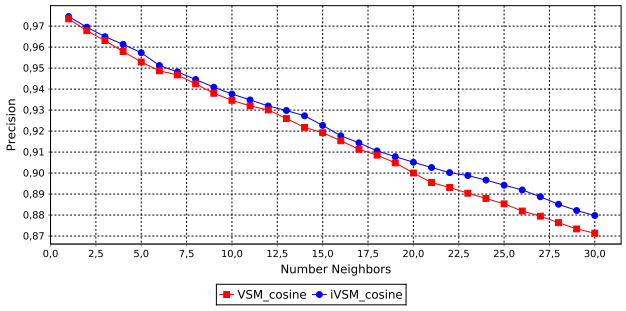
Fig. 2. NJ layouts generated for News2011 text datasets, using the VSM and iVSM representations and the cosine distance. Circle color maps document class.

Fig. 3 show the neighborhood preservation curves of the layouts depicted in the previous figures. One observes that the LSP with VSM does better, whereas both VSM and iVSM curves relative to the NJ layouts are very similar, although iVSM does slightly better.

[2]http://infoserver.lcad.icmc.usp.br/infovis2/NeighborJoiningTree



(a) LSP



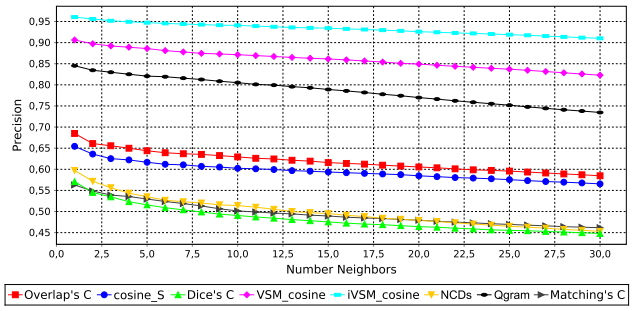
(b) NJ

Fig. 3. NH graphs of LSP and NJ layouts of News2011 built with the VSM and iVSM models and cosine similarity.

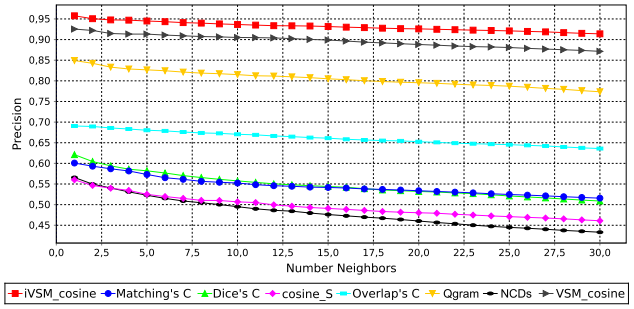
We also compared the neighborhood preservation capability of layouts obtained using distance matrices computed with distinct string similarity measures, plus the cosine similarity computed over the VSM and iVSM models.

Results are shown in Fig. 4. We considered two configurations of LSP with 177 control points and 15 nearest-neighbors, and with 150 control points and 20 nearest-neighbors. The text preprocessing applied varied depending on the dissimilarity measure employed. In generating the VSM and iVSM models we applied general and domain specific stopword removal and no stemming. For VSM a lower Luhn's cut was set to 10 and no upper cut was adopted; for iVSM the thresholds were computed automatically, and the language model has been computed from the AP_BBC_CNN_Reuters.zip dataset. For the string distance matrices, pre-processing procedures also varied. General and specific stopwords were removed from the input strings when using the string-based Dice's Coefficient, Matching Coefficient, Overlap Coefficient and Qgram. No stopword removal was applied to the string-based Cosine and the NCDs distance. The choice of applying (or not) stopwords removal has been made after verifying which alternative produced the best NH curves.

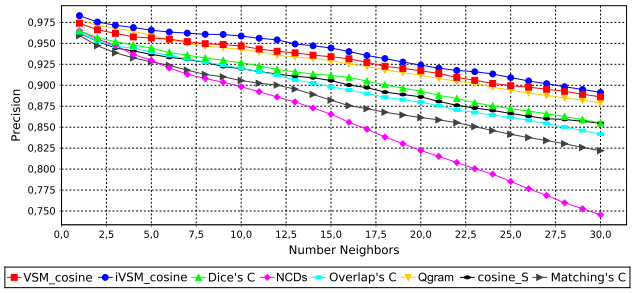
In the first LSP configuration, shown in Fig. 4a, best results regarding class segregation capability were obtained with cosine distance over the iVSM and VSM models and string-based Qgram, which all show curves with values above 0.73. The string-based Dice's Coefficient, Matching Coefficient and NCDs resulted in the worst performances (curves staying bellow 0.6). In the second LSP configuration, shown in Fig. 4b,



(a) LSP 1



(b) LSP 2

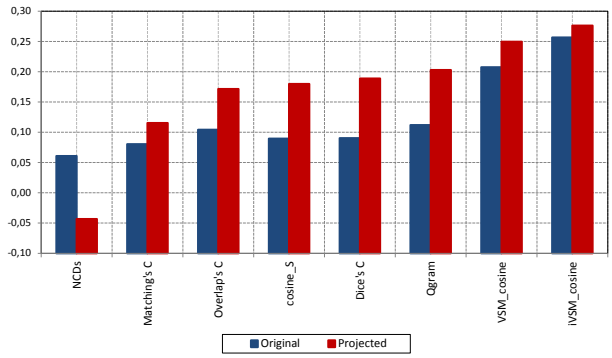


(c) NJ

Fig. 4. NH graphs of LSP and NJ layouts of News2011 obtained with 8 distinct distance matrices: 6 string functions plus the VSM and iVSM with cosine dissimilarity.

one notices that iVSM_cosine, VSM_cosine and Qgram kept the best performances. Note that in this configuration NH curves outperform slightly the ones in Fig. 4a. The worst results were returned by string-based Cosine (identified in the figures as cosine_S) and NCDs. Moreover, all NH curves produced by NJ (Fig. 4c) achieve similar precision values, above 0.75. Nonetheless, the best results are again by iVSM_cosine, VSM_cosine and Qgram.

Fig. 5 shows the Silhouette Coefficients (SC) computed for the dataset considering different distance functions, in the original (blue bars) and in the NJ-tree visual space (red bars). Distances in the NJ-tree are computed considering path lengths. As discussed in Section IV, SC values closer to 1.0 indicate highly cohesive and well separated clusters, according to the distance function considered. One observes how the choice of the distance function affects the grouping of elements based on similarity, in both the original and the



(a) CBR-ILP-IR

Fig. 5. Silhouette Coefficients of datasets, in the original feature space and in the NJ-tree visual space (different distance functions)

visual data spaces.

Ideally, a similarity-based layout should not degrade cluster quality, or even better it could actually improve it, favoring user perception of possibly meaningful structures. Indeed, the figures show that the NJ layout does improve cluster quality relative to the feature space in some cases, in terms of cohesiveness and separation, as measured by the SC. Inspecting the bar charts one notices that cluster quality in the feature space may be poor, and some distance functions are more effective than others in identifying better quality clusters.

For the News2011 data, we notice that all distance functions actually contributed to a projected layout with improved cluster quality. In fact, all distances produced low SC values in the feature space, and improve in the projected layouts, with the exception of layout obtained with the NCDs. SC value in the projected space is better for all functions, with the iVSM_cosine distance doing the best job in this matter.

It is worth noting that we did not consider the Silhouette Coefficient on the LSP projection because distance computation in 2D space is not necessarily a meaningful measure of cluster quality in the visual space, when cluster shapes vary largely, because clustering algorithms tend to favor identification of round-shaped clusters.

VI. Conclusions

In our experiments we observed that VSM and iVSM generated visualizations with the best class segregation capability. Similarity-based layouts of text collections obtained using both models were compared using Neighborhood Hit curves, for which values close to 1.0 reflect layouts with good class preservation capability. A global ranking summarizing the major findis is presented in Table IV. The iVSM outperformed, or otherwise stayed close, to the VSM in most cases. Since the VSM is not incremental, whereas many current datasets are, new additions to the collection force a global recalculation of the feature space and similarities. Given the observed results, we propose iVSM as a new incremental model based on VSM. Coupled with incremental MDS techniques, e.g., *incBoard* and

TABLE IV
RANKING OF NH CURVES OF LAYOUTS OBTAINED WITH STRING-BASED METRICS AND WITH THE COSINE SIMILARITY COMPUTED OVER VSM AND IVSM ON THE THREE DATASETS.

	CBR-ILP-IR			News2011			NewsReuters		
Ranking	LSP (1)	LSP (2)	NJ	LSP (1)	LSP (2)	NJ	LSP (1)	LSP (2)	NJ
1	iVSM	Dice's C	Overlap's C	iVSM	iVSM	iVSM	iVSM	iVSM	Cosine
2	Dice's C	Cosine	Qgram	VSM	VSM	VSM	VSM	VSM	Dice's C
3	Cosine	Overlap's C	NCDs	Qgram	Qgram	Qgram	Cosine	Cosine	iVSM
4	Qgram	iVSM	Cosine	Overlap's C	Overlap's C	Dice's C	Dice's C	Dice's C	Overlap's C
5	VSM	VSM	iVSM	Cosine	Matching's C	Cosine	Overlap's C	Overlap's C	VSM
6	Overlap's C	NCDs	Dice's C	NCDs	Dice's C	Overlap's C	Qgram	Qgram	Matching's C
7	Matching's C	Matching's C	VSM	Matching's C	Cosine	Matching's C	NCDs	Matching's C	Qgram
8	NCDs	Qgram	Matching's C	Dice's C	NCDs	NCDs	Matching's C	NCDs	NCDs

incSpace, it is well-suited for handling text streams and time-stamped document collections, with limited recalculations. Even though the incremental model employs a language model instead of constantly regenerating the word space, the iVSM behaved slightly better than the VSM in almost all cases.

Some string-based metrics also performed well in the comparisons, in particular Qgram, string based Cosine and Overlapping Coefficient. Their major advantage is not requiring intermediate text representations such as the vector models, althoug distance calculations are computationally expensive. A next step is to evaluate iVSM and string measures in a truly incremental setup, by applying them in displaying text streams with, e.g., *incBoard* or *incSpace*.

The approaches considered disregard any kind of semantic analysis of text. For instance, stemming in preprocessing impacts semantics in a not very predictable manner. Although this type of processing and dissimilarity calculation suffices for many applications, further investigation should be conducted on semantic-based distances, as semantics cannot be ignored in some text analytics applications. The impact of the language model also needs further study.

ACKNOWLEDGMENT

The authors would like to thank Roberto Dantas de Pinho for his assistance with the iVSM model and useful discussions. They are also grateful to FAPESP and CNPq for the financial support to this work.

REFERENCES

[1] G. Salton, A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, November 1975.

[2] S. Huang, M. Ward, and E. Rundensteiner, "Exploration of Dimensionality Reduction for Text Visualization," in *Proceedings of the Coordinated and Multiple Views in Exploratory Visualization*, 2005, pp. 63–74.

[3] G. P. Telles, R. Minghim, and F. V. Paulovich, "Normalized compression distance for visual analysis of document collections," *Computers & Graphics*, vol. 31, no. 3, pp. 327–337, June 2007.

[4] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow, "Visualizing the non-visual: spatial analysis and interaction with information from text documents," in *Proceedings of the 1995 IEEE Symposium on Information Visualization*, 1995, pp. 51–58.

[5] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz, "Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and its Application to Document Mapping," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 564–575, May 2008.

[6] F. V. Paulovich and R. Minghim, "HiPP: A Novel Hierarchical Point Placement Strategy and its Application to the Exploration of Document Collections," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1229–1236, November 2008.

[7] A. M. Cuadros, F. V. Paulovich, R. Minghim, and G. P. Telles, "Point Placement by Phylogenetic Trees and its Application to Visual Analysis of Document Collections," in *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, 2007, pp. 99–106.

[8] R. Pinho, M. C. F. de Oliveira, and A. A. Lopes, "Incremental board: a grid-based space for visualizing dynamic data sets," in *Proceedings of the 2009 ACM symposium on Applied Computing*, 2009, pp. 1757–1764.

[9] ——, "An incremental space to visualize dynamic data sets," *Multimedia Tools and Applications*, vol. 50, no. 3, pp. 533–562, December 2010.

[10] R. Pinho, "Espaço incremental para a mineração visual de conjuntos dinâmicos de documentos. (in Portuguese)," PhD Thesis, Instituto de Matemáticas e de Computação, Universidade de São Paulo, 2009.

[11] F. S. Roman, "Um estudo sobre o papel de medidas de similaridade em visualização de coleções de documentos (in Portuguese)," Master Dissertation, Instituto de Matemáticas e de Computação, Universidade de São Paulo, 2012.

[12] F. San Roman, R. D. Pinho, R. Minghim, and M. de Oliveira, "A study on the role of similarity measures in visual text analytics," in *Proceedings VISIGRAPP - 8th. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2013, pp. 429–438.

[13] T. K. Landauer, D. S. McNamara, S. Dennis, and W. Kintsch, *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, March 2003.

[15] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang, "TIARA: A Visual Exploratory Text Analytic System," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 153–162.

[16] A. A. Lopes, R. Pinho, F. V. Paulovich, and R. Minghim, "Visual text mining using association rules," *Computer and Graphics*, vol. 31, no. 3, pp. 316–326, 2007.

[17] W. Cohen, P. Ravikumar, and S. Fienberg, "A Comparison of String Distance Metrics for Name-Matching Tasks," in *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, 2003, pp. 73–78.

[18] S. Kempken, W. Luther, and T. Pilz, "Comparison of distance measures for historical spelling variants," in *Artificial Intelligence in Theory and Practice*, 2006, pp. 295–304.

[19] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[20] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, March 1970.

[21] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison-Wesley, 2005.

[22] J. G. S. Paiva, L. Florian, H. Pedrini, G. P. Telles, and R. Minghim, "Improved Similarity Trees and their Application to Visual Data Classification," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2459–2468, December 2011.