Prompting the data transformation activities for cluster analysis on collections of documents

Discussion Paper
Prompting the data transformation activities for cluster analysis on collections of documents

Tania Cerquitelli, Evelina Di Corso, Francesco Ventura and Silvia Chiusano

Politecnico di Torino - Dipartimento di Automatica e Informatica
Corso Duca Degli Abruzzi, 24 - 10129 Torino - ITALY
{name.surname}@polito.it

Abstract. In this work we argue towards a new self-learning engine able to suggest to the analyst good transformation methods and weighting schemas for a given data collection. This new generation of systems, named SELF-DATA (SELF-learning DAta TrAnsformation) relies on an engine capable of exploring different data weighting schemas (e.g., normalized term frequencies, logarithmic entropy) and data transformation methods (e.g., PCA, LSI) before applying a given data mining algorithm (e.g., cluster analysis), evaluating and comparing solutions through different quality indices (e.g., weighted Silhouette), and presenting the 3-top solutions to the analyst. SELF-DATA will also include a knowledge database storing results of experiments on previously processed datasets, and a classification algorithm trained on the knowledge base content to forecast the best methods for future analyses.
SELF-DATA's current implementation runs on Apache Spark, a state-of-the-art distributed computing framework. The preliminary validation performed on 4 collections of documents highlights that the TF-IDF and logarithmic entropy weighting methods are effective to measure item relevance with sparse datasets, and the LSI method outperforms PCA in the presence of a larger feature domain.

Keywords: Text mining; parameter-free technique; Big data framework.

1 Introduction

Data-driven analysis is a multi-step process, in which data scientists tackle the complex task of configuring the analytics system to transform data into actionable knowledge. Since collections of textual data are usually characterized by a high variability, the knowledge extraction process is challenging and requires a lot of expertise. The text mining is focused on studying algorithms to find implicit, previously unknown, and potentially high-quality information from a large collection of documents. Text mining activities include: (i) grouping data/documents with similar properties or similar content [3], (ii) topic detection [1], and (iii) document summarizations [5].

Effectively performing the text mining process on textual data requires a multi-step process involving different algorithms and for each one different specific parameters should be manually set by the end-user. Furthermore, different methods exist and the selection of the optimal ones is guided by the expertise of the analyst. Innovative and scalable solutions need to be devised to suggest to the analyst how configure the mining process to relieve the end-user of the burden of selecting proper methods for the overall cluster analysis process [2].

This paper proposes SELF-DATA (SELF-learning DAta TrAnsformation), a distributed engine to suggest to the analyst a good configuration of the mining process to cluster a collection of documents into cohesive and well-separated groups. SELF-DATA relies on different building components that characterize the data distribution of collections of documents and *learn* how configure the analytics process through self-tuning and self-learning strategies. Based on previous processed datasets, SELF-DATA will also be able, in the future, to suggest to the analyst the complete configuration of the mining process after the characterization of the data distribution of a new unexplored dataset. SELF-DATA's current implementation runs on the Apache Spark [8] framework, supporting parallel and scalable processing for analytics activities. A preliminary set of experiments have been performed on four textual data collections to show the potential of some components of SELF-DATA. The experiments highlighted the ability of some of SELF-DATA's components to automatically identify a good weighting schema, a good transformation method, and good values for specific-algorithm parameters.

This paper is organized as follows. Section 2 presents the SELF-DATA envisioned architecture and a brief description of its main building components, while Section 3 discusses the preliminary performed experiments. Finally, Section 4 draws conclusions and presents future developments of this work.

2 The SELF-DATA engine

SELF-DATA (SELF-learning DAta TrAnsformation) is a distributed engine whose aim is to suggest to the analyst a good configuration of the whole mining process to cluster a textual data collection into correlated groups of documents with a similar topic. The components of the SELF-DATA envisioned architecture, as well as the interactions between such components, are shown in Figure 1. It relies on 5 main components exploited in 2 phases, i.e., *learning* and *prediction* phases.

Learning phase. During the learning phase, SELF-DATA learns good configurations of the whole mining process for specifics data distributions of textual data collections from previous processed datasets. To this aim SELF-DATA relies on 4 main components able to (i) automatically configure the complete clustering activity (e.g., selecting the good weighting schema and transformation method, automatically setting the specific-algorithm input parameters) by exploiting self-tuning strategies; (ii) characterize different data distribution through the computation of various indices; (iii) store results of steps (i) and (ii)

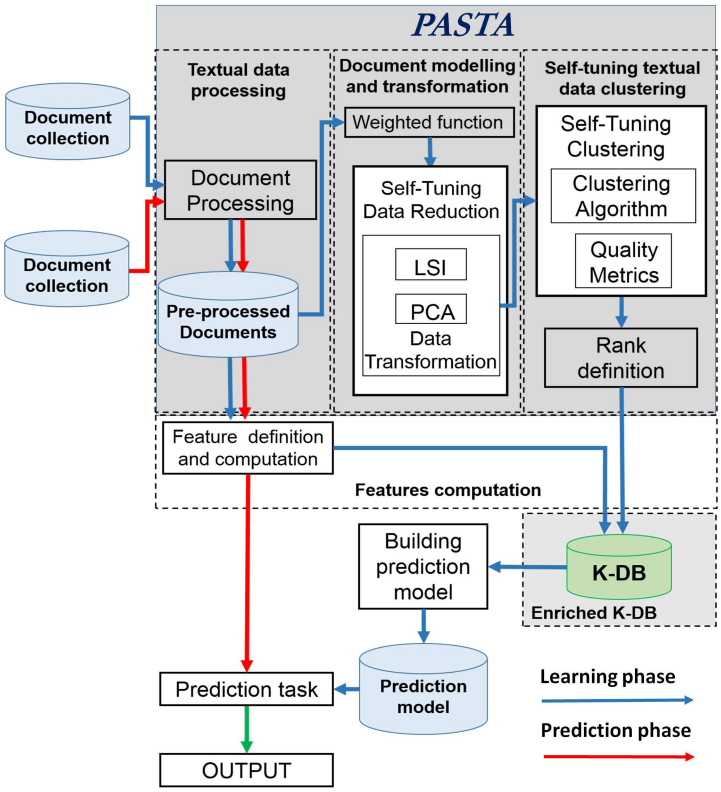

Fig. 1: The SELF-DATA architecture

into a knowledge base; and (iv) exploit the knowledge base to learn which subsets of data distribution indices with the corresponding values are mainly able to forecast the best configuration of the cluster analysis process of textual data collections. Specifically, the following components characterize the SELF-DATA learning phase.

– PASTA (PArameter-free Solutions for Textual Analysis) [4] is *a distributed self-tuning engine* including parameter-setting procedures to relieve the end-user of the burden of selecting proper values for the overall cluster analysis process on a collection of documents. The PASTA engine, reported in Figure 1, includes all the analytics blocks to make the overall analysis problem more effectively tractable and minimize user intervention. Specifically, PASTA includes three main building blocks: (i) *Textual data processing*, (ii) *Document modeling and transformation* including a self-tuning data reduction algorithm, and (iii) *Self-tuning textual data clustering*.
PASTA explores different suitable data weighting strategies, based on local (*Term-Frequency* (TF) and *Logarithmic term frequency* (Log)) and global

(*Inverse Document Frequency* (IDF)) weights and data transformation methods (e.g., PCA, LSI) before applying a clustering algorithm to gain better insights from the analysis. To streamline the analytics process PASTA includes two procedures to relieve the end-user of the burden of selecting proper values for algorithm-specific parameters. These procedures evaluate and compare solutions through different quality indices (e.g., Silhouette-based indices, rand index, f-measure) to identify a few good configurations (3-top) of the mining process that yield a good partition of the original dataset. PASTA's current implementation runs on the Apache Spark framework, supporting parallel and scalable processing for analytics activities.

- The *Feature computation* component is an engine capable of characterizing the data distribution of a collection of documents (corpus) through various indices including
  - # categories: the original number of topics in the dataset under analysis (if known);
  - Avg frequency terms: the average frequency of a term's occurrence in the corpus;
  - Max frequency: the maximum frequency of a term's occurrence in the corpus;
  - Min frequency: the minimum frequency of a term's occurrence in the corpus;
  - # documents: the number of documents in the corpus;
  - # terms: number of terms in the corpus, with repetitions;
  - Dictionary: the number of different terms in the corpus (without repetition);
  - TTR: the relation between the dictionary variety and the total number of terms in the corpus. It is calculated as a ratio between *Dictionary* and *# terms*;
  - Hapax %: the ratio between number of Hapax (number of terms with one occurrence) and the cardinality of the dictionary;
  - Guiraud Index: the ratio between the cardinality of the dictionary and the square root of *# terms*. It represents the *lexical richness*.
- **K-DB** is a knowledge base storing results of 3-top solutions identified through PASTA on previously processed datasets, including data distribution characterization. Specifically, for each processed dataset K-DB stores one record for each solution selected by PASTA. Each record includes the values of the various indices characterizing the data distribution, the selected weighting schema and transformation method, with the suggested input parameter value for either the LSI or PCA transformation method, and the suggested value for the input parameter of the clustering algorithm (i.e., the desired number of cluster).
- **Building a prediction model** addresses a classification problem trained on the knowledge base content to generate a prediction model to suggest to the analyst a good configuration/setting for the cluster analysis process on a given unexplored dataset. Many different approaches (e.g., data mining approaches, logic programming, statistics models) can be exploited for this complex and challenging task.

Prediction phase. Given a new document collection, SELF-DATA first applies the document processing step and then computes the set of features to characterize data distribution. Given the values of the data distribution indices, the prediction model is exploited to forecast a good configuration of the cluster analysis process. This step suggests to the analyst a good weighting schema (both local and global) and a transformation method, and good values for specific-algorithm input parameters to effectively cluster the data collection into a few well-separated and cohesive groups of documents addressing a similar topic.

3 Preliminary development and results

Here we presented a preliminary implementation of the SELF-DATA system with its preliminary results.

Preliminary development. A preliminary implementation of some components of SELF-DATA has been developed.

The current implementation includes: (1) the feature computation building block able to characterize the data distribution through some basic indices (see Section 2 for further detail) and (2) an extended version of the PASTA component presented in [4]. PASTA includes:

- two local weights (TF and Log) and two global weights (IDF and Entropy) [6] to measure term relevance;
- two data transformation methods LSI and PCA;
- two self-tuning algorithms to automatically select the specific-algorithm parameters of data transformation method (input parameter is K, the number of relevant dimensions to consider) and the K-means algorithm (input parameter is k_{cls}, the desired number of clusters) respectively [4].

PASTA exploits various quality metrics to evaluate different configurations of the mining process.

Silhouette-based indices: purified and weighted Silhouette. The *Silhouette index* [7] measures both intra-cluster cohesion and inter-cluster separation. The *weighted silhouette index* WS [4], computed on a given collection of documents, is the ratio between (i) the sum of the percentage of documents in each positive bin weighted with an integer value (weights in $[w_{min} = 1 - w_{max} = 10]$, where the highest weight is assigned to the first bin [1 - 0.9] and so on) and (ii) the overall sum of weights. The higher the weighted silhouette index, the better the identified partition. The *purified silhouette index* PS [4] disregards documents appearing in singleton clusters to reduce the impact of these documents in the overall Silhouette index.

Classification technique to assess the robustness of a given document partition. PASTA builds a *classifier* using the same input features of the clustering algorithm, and the class label assigned by the clustering algorithm as

target. PASTA integrates the random forest method [6] to create the classification model. Quality metrics such as *accuracy and f-measure*[1] are then computed. The higher the classification metrics, the better the overall quality of the clusters.

Preliminary results. We performed preliminary experiments on four real collections of documents (see Table 1) to evaluate (i) the ability of the proposed indices to characterize and differentiate data distributions and (ii) the ability of the PASTA component to correctly identify good configurations for the cluster analysis.

Characterization of the data distribution. Table 1 shows the proposed data distribution indices to characterize data distribution of the considered datasets. The Guirand index is a good parameter to distinguish datasets with a very sparse data distribution (D1 and D2 in Table 1) from less sparse (D3 and D4) datasets. Datasets with high values for lexical richness are usually characterized by a high variability in the data dictionary, thus more sparse data distribution. The TTR index also allows to better differentiate sparse data distribution from more dense data distributions. In fact, the TTR value for D3 is one order of magnitude higher than the TTR values of the others.

Table 1: Datasets

	Wikipedia		Twitter	Reuters
ID	D1	D2	D3	D4
# categories	5	10	-	2
Avg frequency terms	17	23	2	12
Max frequency	8,642	19,437	86	2,605
Min frequency	1	1	1	1
# documents	989	2468	978	1,000
# terms	1,138,570	3,118,499	3,997	80,060
Dictionary	67,539	13,649	1,727	6,453
TTR	0.059	0.044	0.432	0.081
Hapax %	0.505	0.519	0.713	0.371
Guiraud Index	63.3	77.3	27.3	22.8

Selecting good configurations for the whole process of cluster analysis. For each dataset PASTA has been run eight times, once for each combination of a weighting function and a transformation method. Table 2 shows the 3-top solutions identified by PASTA for dataset D1 and by considering the LSI method and all weighting functions. The optimal solution is reported in bold.

For D1 PASTA identifies for K (i.e., number of relevant dimensions for LSI) 34, 37 and 112 dimensions to be considered for the cluster analysis. Given these numbers of dimensions, PASTA selects $k_{cls} = 10$ (input parameter for the K-means algorithm) as the optimal partition. We observed that PASTA usually

[1] It is the weighted harmonic mean of precision and recall

selects as optimal partition the experiment exploiting a low-medium number of dimensions (terms). The higher the K, the more variable the data distribution is and the more complex the cluster activity will be. Thus, the Silhouette-based indices (purified PS and weighted WS) tend to slightly decrease when a large number of terms featuring in each document is analyzed. Furthermore, the results reported in Table 2 highlight that the TF-IDF, Log-IDF and TF-Entropy weighting functions are able to better differentiate the weighting terms, thus identifying a larger number of clusters (associated with different topics in the same category) than the one discovered by Log-Entropy.

Table 2: Results for dataset D1

DId	Trans	Weight	K	k_{cls}	Accuracy	WA-F-Measure	PS	WS
D1	LSI	TF-IDF	34	6	0.99	0.99	0.3	0.058
			37	**10**	**0.96**	**0.96**	**0.3**	**0.062**
			112	12	0.91	0.91	0.2	0.039
D1	LSI	Log-IDF	**19**	**7**	**0.95**	**0.95**	**0.3**	**0.068**
			27	5	0.98	0.98	0.3	0.049
			76	6	0.97	0.97	0.2	0.043
D1	LSI	TF-Entropy	**38**	**10**	**0.91**	**0.91**	**0.3**	**0.056**
			44	16	0.89	0.89	0.3	0.052
			90	6	0.97	0.96	0.2	0.037
D1	LSI	Log-Entropy	**21**	**5**	**0.99**	**0.99**	**0.4**	**0.076**
			27	5	0.98	0.98	0.4	0.066
			84	6	0.97	0.97	0.2	0.048

We also compared the optimal configurations selected by PASTA when it explores LSI with respect to the ones obtained with PCA. Table 3 shows the best configuration selected by PASTA for both LSI and PCA and for all weighting functions. Results are obtained on D3 and D4 datasets. For datasets D1 and D2 the LSI method outperforms PCA (detailed results are omitted to lack of space), whereas for dataset D3 and D4 PCA yield a better final partition than LSI. Thus, we can conclude that the LSI method outperforms PCA in the presence of a larger data domain, and usually a more sparse data distribution.

4 Conclusions and future work

This paper presents a challenging vision for a self-learning engine capable of predicting how to configure the cluster analysis process of a collection of textual data. A preliminary implementation of some components has been presented and evaluated on four real collections of documents. We plan to develop the complete SELF-DATA system in a future work.

Table 3: Results for D3 and D4 datasets

DId	Trans	Weight	K	k_{cls}	Accuracy	WA-F-Measure	PS	WS
D3	LSI	TF-IDF	16	20	0.89	0.89	0.4	0.063
		Log-IDF	16	18	0.86	0.86	0.4	0.062
		TF-Entropy	13	20	0.9	0.9	0.4	0.061
		Log-Entropy	12	15	0.86	0.86	0.4	0.059
D3	PCA	TF-IDF	16	17	0.87	0.87	0.3	0.061
		Log-IDF	16	17	0.92	0.92	0.4	0.064
		TF-Entropy	13	20	0.88	0.88	0.3	0.064
		Log-Entropy	12	18	0.9	0.88	0.4	0.062
D4	LSI	TF-IDF	30	3	0.96	0.96	0.3	0.045
		Log-IDF	18	8	0.94	0.94	0.3	0.064
		TF-Entropy	29	7	0.92	0.92	0.3	0.059
		Log-Entropy	18	8	0.94	0.94	0.3	0.063
D4	PCA	TF-IDF	30	3	0.99	0.99	0.3	0.047
		Log-IDF	18	13	0.9	0.9	0.3	0.069
		TF-Entropy	29	3	0.98	0.98	0.3	0.052
		Log-Entropy	18	5	0.95	0.95	0.3	0.058

References

1. Luca Cagliero, Tania Cerquitelli, Paolo Garza, and Luigi Grimaudo. Twitter data analysis by means of strong flipping generalized itemsets. *Journal of Systems and Software*, 94:16–29, 2014.
2. Tania Cerquitelli, Elena Baralis, Lia Morra, and Silvia Chiusano. Data mining for better healthcare: A path towards automated data analysis? In *32nd IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2016, Helsinki, Finland, May 16-20, 2016*, pages 60–63, 2016.
3. Tania Cerquitelli, Silvia Chiusano, and Xin Xiao. Exploiting clustering algorithms in a multiple-level fashion: A comparative study in the medical care scenario. *Expert Syst. Appl.*, 55:297–312, 2016.
4. Evelina Di Corso, Tania Cerquitelli, and Francesco Ventura. Self-tuning techniques for large scale cluster analysis on textual data collections. In *Proceedings of the 32nd Annual ACM Symposium on Applied Computing, Marrakesh, Morocco, April 3rd-7th, 2017*, pages 771–776, 2017.
5. Oskar Gross, Antoine Doucet, and Hannu Toivonen. Language-independent multi-document text summarization with document-specific word associations. In *31st Annual ACM Symposium on Applied Computing*, Pisa, Italy, 2016. ACM, ACM.
6. Pang-Ning T. and Steinbach M. and Kumar V. *Introduction to Data Mining*. Addison-Wesley, 2006.
7. Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.
8. Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *NSDI'12*, 2012.