An Ecology-based Index for Text Embedding and Classification

Alessio Martino, Enrico De Santis, Antonello Rizzi
Department of Information Engineering, Electronics and Telecommunications
University of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome, Italy
{alessio.martino, enrico.desantis, antonello.rizzi}@uniroma1.it

***Abstract*—Natural language processing and text mining applications have gained a growing attention and diffusion in the computer science and machine learning communities. In this work, a new embedding scheme is proposed for solving text classification problems. The embedding scheme relies on a statistical assessment of relevant words within a corpus using a compound index originally proposed in ecology: this allows to spot relevant parts of the overall text (e.g., words) on the top of which the embedding is performed following a Granular Computing approach. The employment of statistically meaningful words not only eases the computational burden and the embedding space dimensionality, but also returns a more interpretable model. Our approach is tested on both synthetic datasets and benchmark datasets against well-known embedding techniques, with remarkable results both in terms of performances and computational complexity.**

***Index Terms*—Text Classification, Natural Language Processing, Granular Computing, Support Vector Machine, Explainable Artificial Intelligence, Supervised Learning, Embedding Spaces.**

I. Introduction

One of the most relevant problems in automated Pattern Recognition approached through Machine Learning is the 'representation', that is finding the right method to represent real-world objects, often heavily structured, in such a way that they can be elaborated by standard Machine Learning algorithms. This is even more true when dealing with Natural Language Processing and related text mining techniques, due to the structured nature of text as source of *meaning* produced by a really complex machinery – the brain. In turn, 'meaning' emerges in a not yet well-understood way from its underlying complex and hierarchical structure. In fact, in textual data we can find a growing granulation starting from atomic objects, such as letters grouped in words and words suitably grouped in sentences that, in turn, are grouped in paragraphs, documents and corpora.

Within this setting, Granular Computing (GrC) can be conceived as a suitable toolbox convenient also for solving text mining problems such as text classification. In fact, in GrG, thanks to the notion of 'information granule', systems (and related data) can be perceived at different levels of specificity (detail), depending on the complexity of the problem [1] and the objectives of the analysis. Moreover, 'information granules', which arise in the process of data abstraction, allow discovering regularities in data, climbing the so-called pyramid of knowledge [2], thus transforming raw data in knowledge and, hopefully, in wisdom.

One of the appealing characteristics related to GrC paradigm is the possibility of building classifiers, as input-output processing systems, that are *interpretable* and *comprehensible*. In other words, not only GrC allows building a system where inputs are mathematically mapped to outputs (interpretability), but the adoption of information granules, as abstract but interpretable entities, also increases the comprehensibility of the model, since information granules can be semantically associated to well-defined concepts [3]. These features are in sound with that corpus of practices and leading methodologies recently studied and developed by the scientific community, together with industry and government entities (such as DARPA in USA) under the term "explainable AI" (or XAI) umbrella [4]. Consequently, the possibility of working with XAI models, together with user knowledge about clear rationales behind the machine's decision-making process, opens to automatic knowledge discovery paradigms.

Hence texts, as structured objects, need a way to be embedded in a well-suited algebraic space before feeding any automatic learning procedure. Traditionally, the essence of such algebraic space, capturing some kind of co-occurrence between words and contexts, is built on the top of Distributional Semantics (DS). DS is grounded, in turn, on the distributional hypothesis, that is the similarity of meaning correlates with the similarity of distribution. After all, American linguist Z.S. Harris sustained [5] that 'words that are used and occur in the same contexts tend to purport similar meanings'. Based on this concept, a traditional representation of text documents is known as Bag of Words (BOW). The BOW paradigm discards the information related to the order of words and suggests to represent a text document as an array of occurrence frequencies of vocabulary words, leading to the possibility of defining a (semantic) dissimilarity measure between documents or even between words. The vector representation of documents has been traditionally adopted in Information Retrieval with the so-called Vector Space Model [6], where document vectors are arranged in columns of the term-document matrix. There are a number of techniques to improve the power of this embedding. For example, instead of the raw word frequencies, a weighted version known as Term Frequency-Inverse Document Frequency (TF-IDF) [7],

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

[8], can be adopted, capturing the fact that words occurring roughly uniformly in the corpus tend to be less discriminatory. Hence, they will receive a lower weight. Even if the BOW embedding is highly adopted for its intrinsic simplicity, it has a number of drawbacks, such as the impossibility of capturing synonymy, the sparsity of the representation and the dimension of the feature space that is equivalent to the cardinality of the vocabulary words (often very large). A partial solution is offered by the Latent Semantical Indexing technique [9], [10], grounded on the SVD decomposition of the term-document matrix, where documents are embedded in a low dimensional latent feature space. Recently, many authors have proposed a different family of text embedding techniques based on shallow and deep (recurrent) neural networks, such as skip-gram and CBOW [11] and the more advanced BERT [12], that uses an attention mechanism and is able to embed even sentences.

In this paper, a novel lightweight embedding procedure is proposed which, similarly to BOW, TF-IDF and LSA, provides an explicit embedding space for text classification while, at the same time, leading to a drastically smaller feature space. The GrC paradigm acts as driving force behind a suitable choice of pivotal items on the top of which the embedding is performed by relying on statistical measures inherited from ecology. The resulting model is comprehensible, as common in GrC-based advanced pattern recognition systems, further pushing the boundary towards XAI.

The remainder of the paper is structured as follows: in Section II the proposed embedding procedure is described, along with few remarks on the following classification stage; in Section III the competing embedding techniques, datasets and computational results are shown; finally, Section IV concludes the paper, remarking future research endeavours.

II. Proposed Methodology

In order to embed text documents towards real-valued embedding spaces, a procedure for extracting meaningful information granules (i.e., words) thanks to a sensitivity-vs-specificity integrated evaluation is investigated. This approach, inherited by a classical ecological method originally proposed in [13], was developed in order to spot 'signature species' in a given environment. The logic at the basis of this method (INDVAL) is straightforward: a given species s is 'representative' and therefore useful for the recognition of a given environmental condition ec if it satisfies both of the following properties

1) s must be present only (or almost only) in the ec-positive objects
2) s must be present in all (or the great majority of) the ec-positive cases.

Rather than individuating 'signature species' belonging to different environments, in [14] the INDVAL has been used to spot 'signature substructures' in structured data. Specifically, the INDVAL has been used for individuating relevant edges (chemical reactions) in graphs (metabolic networks) belonging to different organisms properly divided in groups (classes) thanks to the Linnaeus' taxonomy. Under the graphs viewpoint, the INDVAL has been formally defined as:

$$A_{i,j} = \frac{\text{\# graphs having edge } i \text{ in group } j}{\text{\# graphs having edge } i} \tag{1}$$

$$B_{i,j} = \frac{\text{\# graphs having edge } i \text{ in group } j}{\text{\# graphs in group } j} \tag{2}$$

$$I_{i,j} = A_{i,j} \cdot B_{i,j} \cdot 100 \tag{3}$$

By definition, since $A_{i,j} \in [0,1]$ and $B_{i,j} \in [0,1]$, then $I_{i,j} \in [0,100]$. The two supporting scores A and B have a straightforward interpretation: the maximum value of A is obtained when the i^{th} edge can be found only in patterns (graphs) belonging to class j, whereas the maximum value for B is obtained if all patterns of class j have edge i. Finally, the maximum INDVAL I corresponds to the maximum sensitivity and specificity for the i^{th} edge within group j: all patterns of class j have edge i and no patterns belonging to other classes have edge i. This approach has been successfully applied for the definition of GrC-based embedding spaces in the context of metabolic networks analysis and, alongside remarkable performances in classification, the resulting information granules (i.e., relevant edges) have been analysed by field-experts (i.e., biologists) and gave rise to further research and overall considerations on the subject matter [15].

A. Embedding Procedure

In the context of text classification, Eqs. (1)–(3) can be restated as:

$$A_{i,j} = \frac{\text{\# documents having word } i \text{ in group } j}{\text{\# documents having word } i} \tag{4}$$

$$B_{i,j} = \frac{\text{\# documents having word } i \text{ in group } j}{\text{\# documents in group } j} \tag{5}$$

$$I_{i,j} = A_{i,j} \cdot B_{i,j} \cdot 100 \tag{6}$$

More in detail, let us consider a dataset $\mathcal{S} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ of n documents and let $\mathcal{L}$ be the set of corresponding ground-truth labels for the classification problem at hand. Further, consider $\mathcal{S}$ to be split into three non-overlapping training, validation and test sets ($\mathcal{S}_{\text{TR}}$, $\mathcal{S}_{\text{VL}}$, $\mathcal{S}_{\text{TS}}$, respectively) with pairwise empty intersection and whose union returns $\mathcal{S}$. Finally, consider $\mathcal{L}$ to be split accordingly ($\mathcal{L}_{\text{TR}}$, $\mathcal{L}_{\text{VL}}$, $\mathcal{L}_{\text{TS}}$, respectively). Let $\mathcal{W}$ be the set of unique words in $\mathcal{S}_{\text{TR}} \cup \mathcal{S}_{\text{VL}}$, then one can figure $\mathbf{A}, \mathbf{B}, \mathbf{I} \in \mathbb{R}^{|\mathcal{W}| \times p}$ as a compact matrix representation of Eqs. (4)–(6) in which the three scores A, B and I are evaluated for each word in $\mathcal{W}$ against each of the p classes (groups) for the classification problem at hand.

The next step is to filter only relevant words for the embedding stage: given a threshold $T \in (0, 100)$, words in $\mathcal{W}$ having INDVAL score greater than (or equal to) T for at least one of the p classes are retained: the set of resulting words build up the alphabet $\mathcal{A}$, with $M = |\mathcal{A}|$ for the embedding stage. The latter is performed thanks to the symbolic histograms paradigm [14], [16] according to which each pattern (a document $\mathcal{D}$, in this case) is transformed into an M-length integer-valued vector $\mathbf{h}$ that counts, in position i, the number of times the i^{th} symbol from the alphabet appears in $\mathcal{D}$:

$$\mathbf{h} = [count(\mathcal{A}_1 \to \mathcal{D}), \ldots, count(\mathcal{A}_M \to \mathcal{D})] \qquad (7)$$

The above mapping is individually performed on each document belonging to $\mathcal{S}_{\text{TR}}$, $\mathcal{S}_{\text{VL}}$ and $\mathcal{S}_{\text{TS}}$, returning three instance matrices $\mathbf{S}_{\text{TR}} \in \mathbb{R}^{|\mathcal{S}_{\text{TR}}| \times M}$, $\mathbf{S}_{\text{VL}} \in \mathbb{R}^{|\mathcal{S}_{\text{VL}}| \times M}$, $\mathbf{S}_{\text{TS}} \in \mathbb{R}^{|\mathcal{S}_{\text{TS}}| \times M}$.

B. Classification

The embedding space spanned by the three instance matrices $\mathbf{S}_{\text{TR}}$, $\mathbf{S}_{\text{VL}}$ and $\mathbf{S}_{\text{TS}}$ can be equipped with algebraic operators such as the dot product or the Euclidean distance and any classification system can be used without alterations. However, it is possible to further shrink the set of meaningful words (symbols) by a suitable optimisation procedure, possibly leaded by a genetic algorithm [17]. In a general sense, let $\mathcal{H}$ be the set of hyperparameters for the classifier at hand and let $\mathbf{w} \in \{0,1\}^M$ be a boolean vector, acting as a selection mask. Hence, the genetic code reads as $[\mathcal{H}, \mathbf{w}]$.

Each individual from the evolving population strips columns from $\mathbf{S}_{\text{TR}}$ and $\mathbf{S}_{\text{VL}}$ corresponding to 1's in $\mathbf{w}$. The filtered $\mathbf{S}_{\text{TR}}$ trains the considered classification model using the hyperparameters written in the $\mathcal{H}$ portion of the genetic code. Its performance π is then evaluated on the filtered version of $\mathbf{S}_{\text{VL}}$ and serves as (part of) the fitness function F:

$$F = \alpha \cdot \pi + (1-\alpha) \cdot \kappa \qquad (8)$$

where κ takes into account the sparsity of the feature selector $\mathbf{w}$ and $\alpha \in [0,1]$ is a user-defined parameter weighting the two contributions. At the end of the evolution, the best individual is retained and evaluated on the filtered version of $\mathbf{S}_{\text{TS}}$.

This optimisation procedure not only takes into account an automatic tuning of the classifier hyperparameters, but also allows to select a suitable subset of features deemed useful by the classifier itself. Having a small, yet informative, subset of resulting features (alphabet symbols) is crucial in GrC-based classifiers, as it fosters the interpretability of the trained model.

III. Experiments

A. Datasets Description

In order to show the effectiveness of the proposed embedding procedure, three synthetic datasets of progressively harder text classification have been manually built:

TOY: 57 scientific paper abstracts for three different topics: Anatomy, Information Theory, String Theory.

ABS2: 460 scientific paper abstracts for four different topics: Anatomy, Information Theory, String Theory, Semiconductors.

ABS4: 575 scientific paper abstracts for five different topics: Anatomy, Information Theory, Smart Grids, String Theory, Semicondutors.

These three datasets show perfectly balanced classes. Alongside these sets, which will mainly be useful for addressing the knowledge discovery phase of the proposed system, the following benchmark datasets have been used as well for a thorough investigation and comparison:

REUTERS8: collection of documents appeared on the Reuters newswire in 1987. The documents were assembled and indexed with categories by personnel from Reuters Ltd.. The adopted splitting is the "ModApte" split[1] on 7674 documents and 8 classes.

WEATHERREPORTS: dataset included in the standard MATLAB® suite that contains text description from weather reports. Only the top 10 classes have been retained, leading to a dataset composed by 24176 documents.

TDT2: corpus coming from different sources (e.g., newswires, TV and radio shows) for semantic classification. Only the top 30 classes are retained for a total of 9394 documents.

20NEWS: dataset gathered from newsgroups, grouped by topic. There are 20 classes, for a total of 18774 documents.

At odds with their synthetic counterpart, these four datasets show unbalanced classes. All datasets except for TDT2 and 20NEWS, for which preprocessed versions are freely available[2], went through a common pre-processing phase consisting of tokenisation, uppercase-to-lowercase conversion, punctuation removal, stop-words removal and stemming/lemmatisation. The $\mathcal{S}_{\text{TR}}$, $\mathcal{S}_{\text{VL}}$ and $\mathcal{S}_{\text{TS}}$ splits have been built in a label-aware stratified fashion with a ratio of 50% in $\mathcal{S}_{\text{TR}}$ and 25% in both $\mathcal{S}_{\text{VL}}$ and $\mathcal{S}_{\text{TS}}$.

B. Competing Embedding Techniques

The proposed INDVAL-based strategy has been benchmarked against three well-known explicit matrix-based embedding strategies, already mentioned in Section I.

BOW: also known as Term Frequency (TF) [18], BOW consists in evaluating the raw counts of unique words in the corpus (vocabulary) against each document in the corpus. The embedding via BOW leads to a matrix, say $\mathbf{B}$, with as many rows as documents and as many columns as items in the vocabulary, where $\mathbf{B}_{i,j}$ scores the number of times the j^{th} word from the vocabulary appears in the i^{th} document. For the sake of consistency with the INDVAL technique, the vocabulary is built by considering the set of unique words in $\mathcal{S}_{\text{TR}} \cup \mathcal{S}_{\text{VL}}$.

TF-IDF: alongside $\mathbf{B}$, TF-IDF considers another (row) vector, say $\mathbf{t}$, whose i^{th} item is given by $\log(n/c_i)$, where c_i is the number of documents in which the i^{th} item from the vocabulary appears. Multiplying $\mathbf{B}$ and $\mathbf{t}$ returns the embedding matrix.

LSA: obtained by means of SVD decomposition. This means that given a distributional representation through the generic matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ with m being the dimension of the vocabulary – i.e. the BOW matrix $\mathbf{B}^T$, it can be decomposed into the product of three matrices: $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. However, since the decomposition scales as the number of documents, a reduced version of the SVD (truncated SVD) can be provided: $\hat{\mathbf{M}} = \mathbf{U}_{(m\times k)}\mathbf{\Sigma}_{(k\times k)}\mathbf{V}^T_{(k\times n)}$, where $k \ll rank(\mathbf{M})$. In LSA, the matrix $\mathbf{V}$ is discarded and the new reduced representation is given by: $\hat{\mathbf{M}}' = \mathbf{U}_{(m\times k)}\mathbf{\Sigma}_{(k\times k)}$. The row vectors of $\hat{\mathbf{M}}'$ are distributional vectors with latent semantic dimensions

[1] https://link.springer.com/content/pdf/bbm:978-3-642-04533-2/1.pdf

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

$SD = \{d_1, d_2, ..., d_k\}$, while the row vectors of $\mathbf{U}$ represent the target terms. Sometimes, even the singular value matrix $\mathbf{\Sigma}$ is discarded, simplifying the representation. The above-described transformation provides a dense representation that, due to the lower number of features compared to the plain BOW, generally reduces the noise in data. While, for example, in sparse representation, `automobile` and `car` (that are synonyms) are represented in two distinct dimensions, LSA may capture the synonymy modelling the relationship of the similarity between a word with `car` as a neighbour and a word with `automobile` as a neighbour. Moreover, the reduced representation may avoid overfitting providing a better generalisation capability. Due to its interesting and (in some ways) unexpected properties, LSA has also been proposed as a cognitive model for human language use [19].

C. Experimental Setup

Two classification systems have been considered for comparison: ν-SVM [20] and ℓ_1-SVM [21]. The former minimise the 2-norm of the separating hyperplane, whereas the latter minimise the 1-norm of the separating hyperplane. Two version of ν-SVMs are considered: the former is equipped with the radial basis function kernel, therefore the set of hyperparameters $\mathcal{H}$ of the genetic code reads as $\mathcal{H} = \{\nu, \gamma\}$, where $\nu \in (0, 1]$ is the regularisation term and $\gamma \in (0, 100]$ is the kernel shape parameter, whereas the latter is equipped with the linear kernel, hence $\mathcal{H} = \{\nu\}$.

In the fitness function (Eq. (8) – to be minimised), π has been defined as

$$\pi = 1 - \overline{J} = 1 - ((J + 1)/2) \tag{9}$$

where J is the informedness [14], [16], in turn defined as:

$$J = \text{Sensitivity} + \text{Specificity} - 1, \quad J \in [-1, +1]$$

The affine normalisation in Eq. (9) ensures that $\overline{J} \in [0, 1]$, hence $\pi \in [0, 1]$. κ is defined as the ratio of selected symbols:

$$\kappa = \frac{\|\mathbf{w}\|_0}{|\mathbf{w}|} \tag{10}$$

and since $\kappa \in [0, 1]$, they can be fairly combined in Eq. (8).

The set of hyperparameters in the genetic code for ℓ_1-SVMs reads as $\mathcal{H} = \{C, C_-, C_+\}$, where $C \in (0, 10]$ is the regularisation term and $C_-, C_+ \in (0, 10]$ are two additional weights in order to adjust C in a class-aware fashion. It is worth stressing that the two additional weights are not mandatory for ℓ_1-SVMs to work, yet they can be useful in case of unbalanced classes. Since ℓ_1-SVMs automatically perform feature selection during training due to the 1-norm minimisation (i.e., sparse hyperplane coefficient vector), the $\mathbf{w}$ portion of the genetic code has been discarded and the genetic algorithm takes care of optimising only the hyperparameters[3]. κ sees the hyperplane coefficient vector rather than $\mathbf{w}$ in its definition, see Eq. (10).

For all classifiers, a value of $\alpha = 0.5$ has been considered in the fitness function (Eq. (8)) in order to give the same importance to sparsity and performances. The genetic algorithm driving the model synthesis has been configured to host 100 individuals for a maximum number of 100 generations with a rigid early-stop criterion if the average change in the fitness function over $1/3^{\text{rd}}$ of the generations is below or equal to 10^{-6}; the elitism is set to the best 10% individuals per generation, the selection follows the roulette wheel heuristic, the crossover operator generates new offsprings in a scattered (uniform) fashion, the mutation acts in a flip-the-bit fashion for Boolean genes ($\mathbf{w}$) and adds to real-valued genes ($\mathcal{H}$) a random number drawn from a zero-mean Gaussian distribution whose variance shrinks as generations go by. Software setup includes MATLAB® R2019b and its toolboxes for text analysis and optimisation, with LibSVM and LibLINEAR as external dependancies for ν-SVM and ℓ_1-SVM, respectively.

As regards the embedding procedure parameters, namely the threshold T for INDVAL and the number of components k for LSA, the following values have been set:

- $T = 20$ for TOY, ABS2 and ABS4; $T = 10$ for REUTERS8 and TDT2; $T = 5$ for 20NEWS and WEATHERREPORTS
- $k = 30$ for TOY; $k = 250$ for ABS2 and ABS4; $k = 1000$ for REUTERS8; $k = 4000$ for WEATHERREPORTS and 20NEWS; $k = 3500$ for TDT2.

Values for k have been estimated by plotting the normalised cumulative eigenspread and selecting the value where the curve flattens. Values for T have been estimated using this simple, yet effective, heuristic: recall $\mathbf{I} \in \mathbb{R}^{|\mathcal{W}| \times p}$ be the matrix containing the INDVAL scores for each word in $\mathcal{W}$ against each of the p classes. For each word (row), consider the maximum INDVAL score amongst the p columns and plot the distribution of the resulting INDVALs. For the sake of example, Fig. 1 shows the distribution for TOY and REUTERS8. For TOY, the vast majority of the words have maximum INDVAL lower than 20, whereas for REUTERS8 the elbow disappears at $T \simeq 10$, so we do not expect to have highly discriminant words in these respective ranges. The same analysis has been carried on the remaining five datasets, returning the above thresholds.

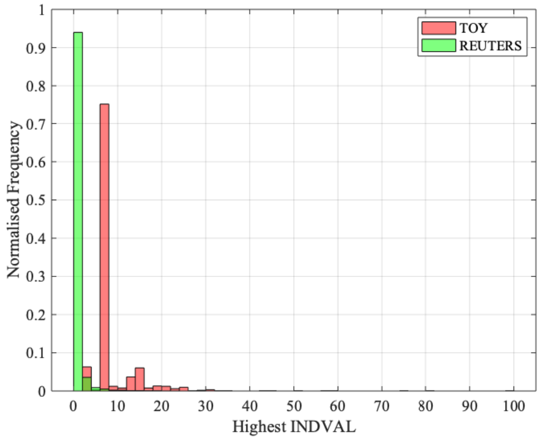


Fig. 1. Distribution of the highest INDVAL scores.

[3] Despite this small set of hyperparameters can also be optimised via lighter heuristics such as grid search or random search, a genetic algorithm has been employed for the sake of comparison with the ν-SVMs case.

D. Computational Results on Synthetic Datasets

Fig. 2–4 show the results for TOY, ABS2 and ABS4, respectively, when using both ℓ_1-SVM and ν-SVM (both linear and non-linear). SVMs are trained (and optimised) in a one-vs-all fashion: in this way, each class has its own feature selection mask. Heatmaps are normalised row-wise (i.e., for each class). Results include the accuracy on the test set and the resulting number of relevant features after feature selection, whereas Table I shows the starting embedding space dimensionality.

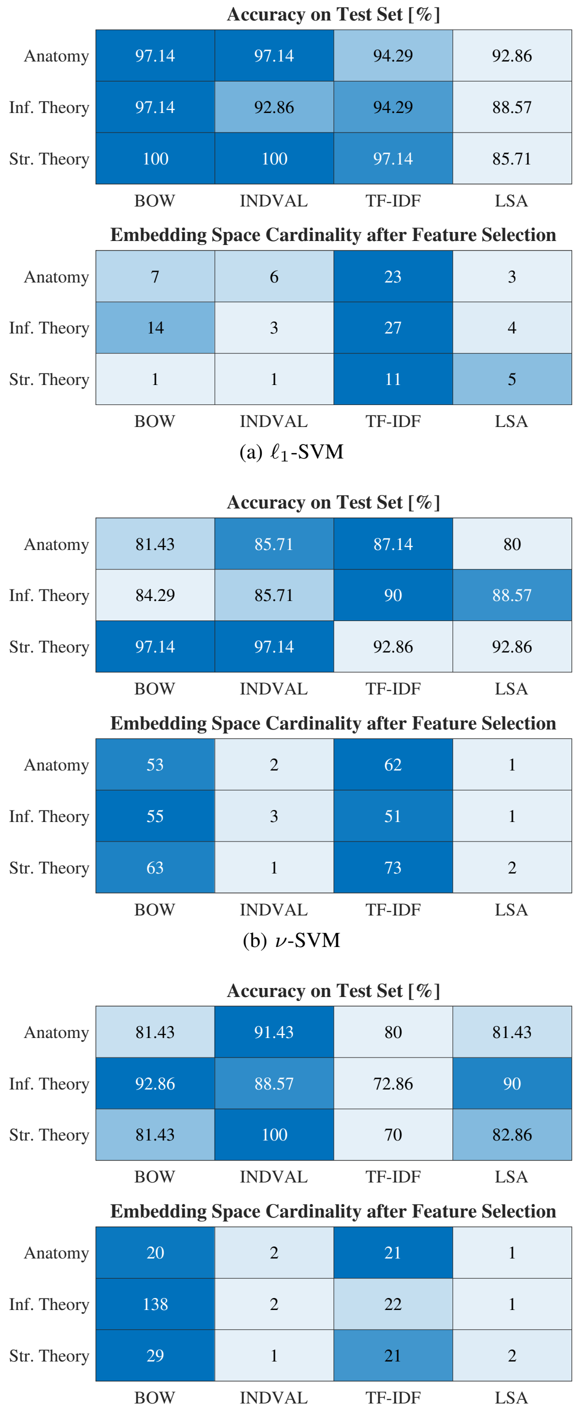
TABLE I
STARTING EMBEDDING SPACE DIMENSIONALITY (SYNTHETIC DATASETS).

	BOW	**INDVAL**	**TF-IDF**	**LSA**
TOY	1200	45	1200	30
ABS2	4848	24	4848	250
ABS4	5618	29	5618	250

Due to the intrinsic randomness in the overall procedure, results herein presented have been averaged across five different $\mathcal{S}_{\text{TR}}-\mathcal{S}_{\text{VL}}-\mathcal{S}_{\text{TS}}$ splits. In order to ensure a fair comparison, the same splits have been fed to all classifiers. A first comparison regards the dimensionality of the embedding space (Table I), with the INDVAL strategy greatly outperforming the three competitors on ABS2 and ABS4, with TOY being the only exception since the suitable number of components for LSA is below the number of relevant words at $T = 20$. After the feature selection phase (Fig. 2–4, bottom panels), the INDVAL strategy also leads to the smallest set of meaningful words for all datasets and for all classifiers: LSA has comparable results with the INDVAL strategy in this regard, yet features for LSA do not correspond to meaningful words. As for classification performances (Fig. 2–4, top panels), the INDVAL strategy outperforms the three competitors when using RBF ν-SVMs, whereas it results to be the least performing strategy on ABS2 and ABS4 when using ℓ_1-SVMs or linear ν-SVMs: this is coherent with current knowledge on large-scale classification [22], where linear classification is particularly suited for high dimensional and sparse data (e.g., BOW, TF-IDF).

A second aspect regards the knowledge discovery phase. Indeed, recall that the INDVAL has the potential to spot relevant words within the corpus. In order to discard spurious selections due to intrinsic randomness in the procedure and focus the analysis only on words that classifiers persistently consider important, let us consider only words that survived the feature selection phase for all of the five $\mathcal{S}_{\text{TR}}-\mathcal{S}_{\text{VL}}-\mathcal{S}_{\text{TS}}$ splits. Furthermore, let us consider ℓ_1-SVMs since they greatly outperform ν-SVMs in terms of selected features, albeit their slight performance decay. Nonetheless, similar knowledge discovery results hold for ν-SVMs as well. Results for TOY:

- class 'Anatomy': TF-IDF, BOW and INDVAL selected only one word (`disease`)
- class 'Information Theory': BOW selected `shannon`, `patient` and `disease`; TF-IDF and INDVAL selected only `shannon`



Accuracy on Test Set [%]

	BOW	INDVAL	TF-IDF	LSA
Anatomy	97.14	97.14	94.29	92.86
Inf. Theory	97.14	92.86	94.29	88.57
Str. Theory	100	100	97.14	85.71

Embedding Space Cardinality after Feature Selection

	BOW	INDVAL	TF-IDF	LSA
Anatomy	7	6	23	3
Inf. Theory	14	3	27	4
Str. Theory	1	1	11	5

(a) ℓ_1-SVM

Accuracy on Test Set [%]

	BOW	INDVAL	TF-IDF	LSA
Anatomy	81.43	85.71	87.14	80
Inf. Theory	84.29	85.71	90	88.57
Str. Theory	97.14	97.14	92.86	92.86

Embedding Space Cardinality after Feature Selection

	BOW	INDVAL	TF-IDF	LSA
Anatomy	53	2	62	1
Inf. Theory	55	3	51	1
Str. Theory	63	1	73	2

(b) ν-SVM

Accuracy on Test Set [%]

	BOW	INDVAL	TF-IDF	LSA
Anatomy	81.43	91.43	80	81.43
Inf. Theory	92.86	88.57	72.86	90
Str. Theory	81.43	100	70	82.86

Embedding Space Cardinality after Feature Selection

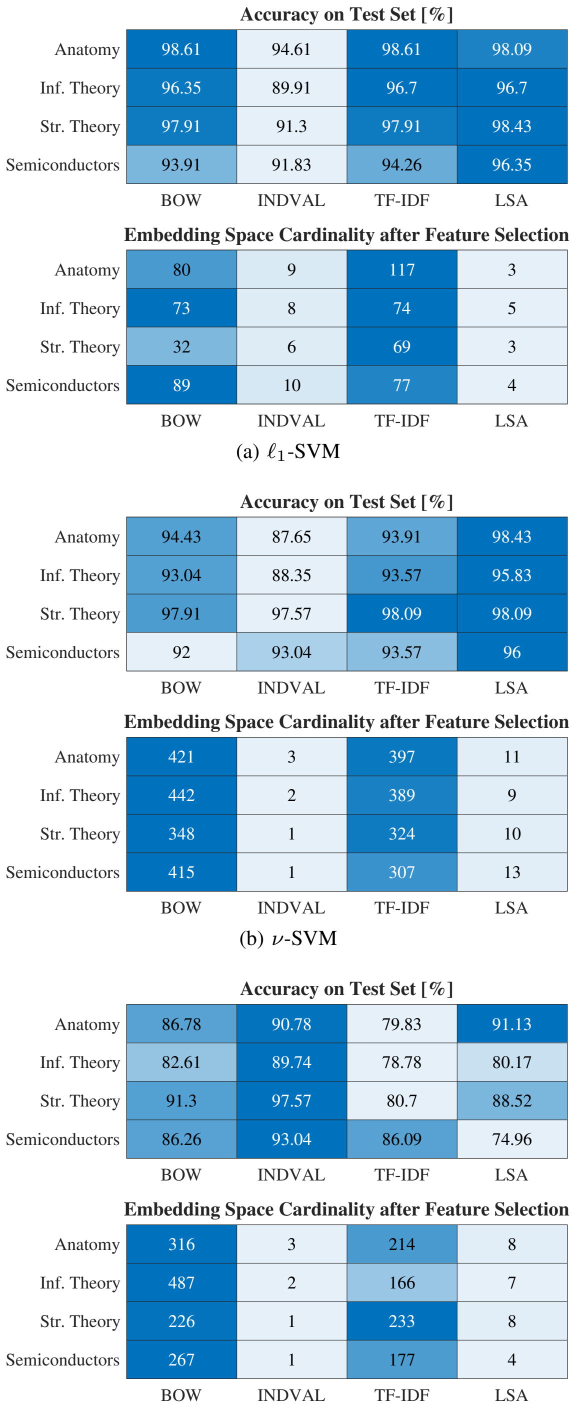
	BOW	INDVAL	TF-IDF	LSA
Anatomy	20	2	21	1
Inf. Theory	138	2	22	1
Str. Theory	29	1	21	2

(c) RBF ν-SVM

Fig. 2. Results on TOY.

- class 'String Theory': TF-IDF, BOW and INDVAL selected only one word (`string`).

For the sake of completeness, the INDVAL strategy assigned (on average) $I = 100$ for `string` in class 'String Theory', $I \simeq 80$ for `shannon` in class 'Information Theory' and $I \simeq 47$ for `disease` in class 'Anatomy'. Whilst for a simple dataset such as TOY results are rather comparable also in terms of knowledge discovery, the same is not true for slightly more complex problems such as ABS2 and ABS4: Fig. 5 and 6



Accuracy on Test Set [%]

	BOW	INDVAL	TF-IDF	LSA
Anatomy	98.61	94.61	98.61	98.09
Inf. Theory	96.35	89.91	96.7	96.7
Str. Theory	97.91	91.3	97.91	98.43
Semiconductors	93.91	91.83	94.26	96.35

Embedding Space Cardinality after Feature Selection

	BOW	INDVAL	TF-IDF	LSA
Anatomy	80	9	117	3
Inf. Theory	73	8	74	5
Str. Theory	32	6	69	3
Semiconductors	89	10	77	4

(a) ℓ_1-SVM

Accuracy on Test Set [%]

	BOW	INDVAL	TF-IDF	LSA
Anatomy	94.43	87.65	93.91	98.43
Inf. Theory	93.04	88.35	93.57	95.83
Str. Theory	97.91	97.57	98.09	98.09
Semiconductors	92	93.04	93.57	96

Embedding Space Cardinality after Feature Selection

	BOW	INDVAL	TF-IDF	LSA
Anatomy	421	3	397	11
Inf. Theory	442	2	389	9
Str. Theory	348	1	324	10
Semiconductors	415	1	307	13

(b) ν-SVM

Accuracy on Test Set [%]

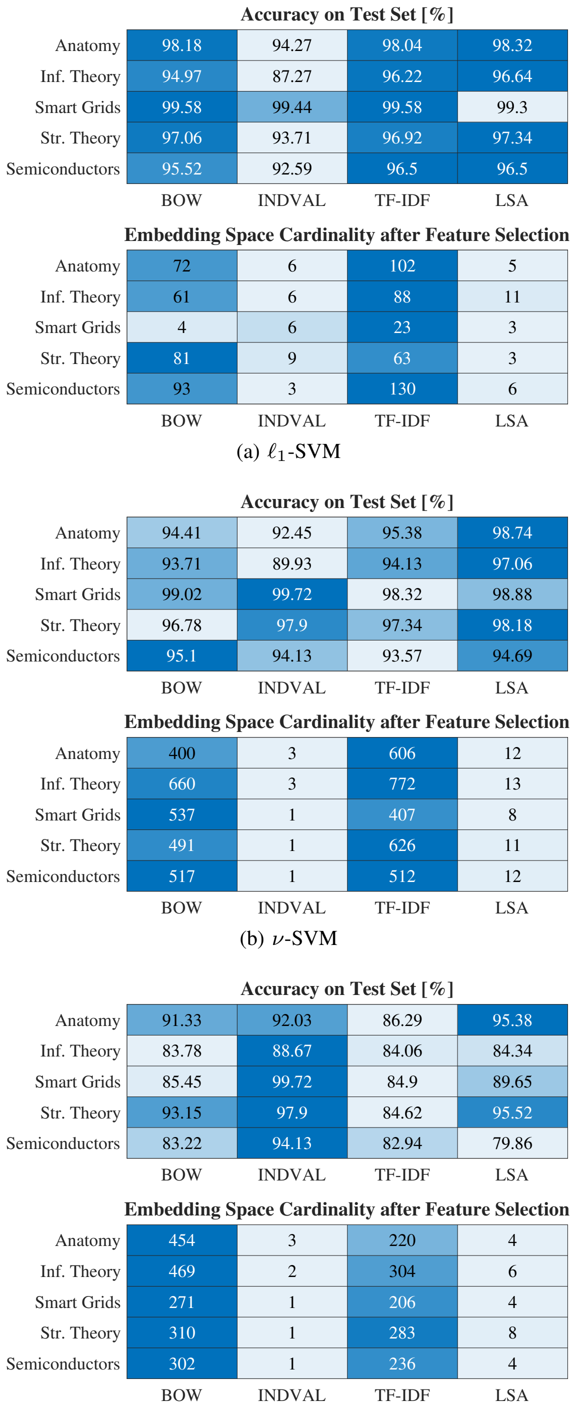
	BOW	INDVAL	TF-IDF	LSA
Anatomy	86.78	90.78	79.83	91.13
Inf. Theory	82.61	89.74	78.78	80.17
Str. Theory	91.3	97.57	80.7	88.52
Semiconductors	86.26	93.04	86.09	74.96

Embedding Space Cardinality after Feature Selection

	BOW	INDVAL	TF-IDF	LSA
Anatomy	316	3	214	8
Inf. Theory	487	2	166	7
Str. Theory	226	1	233	8
Semiconductors	267	1	177	4

(c) RBF ν-SVM

Fig. 3. Results on ABS2.



Accuracy on Test Set [%]

	BOW	INDVAL	TF-IDF	LSA
Anatomy	98.18	94.27	98.04	98.32
Inf. Theory	94.97	87.27	96.22	96.64
Smart Grids	99.58	99.44	99.58	99.3
Str. Theory	97.06	93.71	96.92	97.34
Semiconductors	95.52	92.59	96.5	96.5

Embedding Space Cardinality after Feature Selection

	BOW	INDVAL	TF-IDF	LSA
Anatomy	72	6	102	5
Inf. Theory	61	6	88	11
Smart Grids	4	6	23	3
Str. Theory	81	9	63	3
Semiconductors	93	3	130	6

(a) ℓ_1-SVM

Accuracy on Test Set [%]

	BOW	INDVAL	TF-IDF	LSA
Anatomy	94.41	92.45	95.38	98.74
Inf. Theory	93.71	89.93	94.13	97.06
Smart Grids	99.02	99.72	98.32	98.88
Str. Theory	96.78	97.9	97.34	98.18
Semiconductors	95.1	94.13	93.57	94.69

Embedding Space Cardinality after Feature Selection

	BOW	INDVAL	TF-IDF	LSA
Anatomy	400	3	606	12
Inf. Theory	660	3	772	13
Smart Grids	537	1	407	8
Str. Theory	491	1	626	11
Semiconductors	517	1	512	12

(b) ν-SVM

Accuracy on Test Set [%]

	BOW	INDVAL	TF-IDF	LSA
Anatomy	91.33	92.03	86.29	95.38
Inf. Theory	83.78	88.67	84.06	84.34
Smart Grids	85.45	99.72	84.9	89.65
Str. Theory	93.15	97.9	84.62	95.52
Semiconductors	83.22	94.13	82.94	79.86

Embedding Space Cardinality after Feature Selection

	BOW	INDVAL	TF-IDF	LSA
Anatomy	454	3	220	4
Inf. Theory	469	2	304	6
Smart Grids	271	1	206	4
Str. Theory	310	1	283	8
Semiconductors	302	1	236	4

(c) RBF ν-SVM

Fig. 4. Results on ABS4.

show the selected words for the two datasets, respectively, in which the INDVAL ability of selecting a small subset of relevant words is striking. In ABS2, all competitors agree that the class 'String Theory' is the easiest to characterise, whereas for the other three classes it is possible to notice that not only the INDVAL strategy selects the smallest subset of relevant words, but the selected words perfectly fit with the positive class (e.g., disease and patient for 'Anatomy', information for 'Information Theory'). The same is not true for BOW and TF-IDF, in which also words related to negative classes are selected (e.g., information for 'Anatomy'), along with words that apparently do not characterise neither the positive nor the negative class (e.g., result for 'Anatomy' and conflict for 'Information Theory'). Similar observations hold for ABS4 (Fig. 6), in which 'Smart Grid' seems the easiest class to characterise (one word needed). The average INDVAL scores for ABS2 are: disease ($I \simeq 35$) and patient ($I \simeq 50$) for class 'Anatomy'; information ($I \simeq 54$) for class 'Information

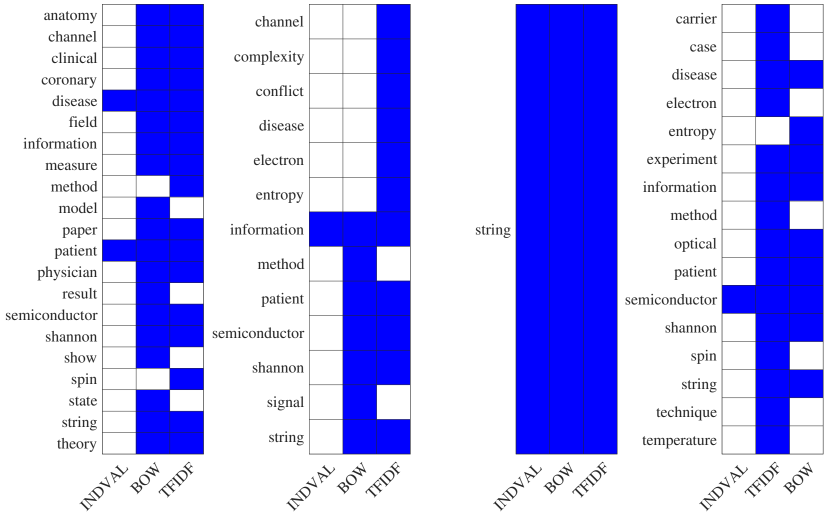


Fig. 5. Knowledge discovery results on ABS2. Left to right: Anatomy, Inf. Theory, String Theory, Semiconductors.

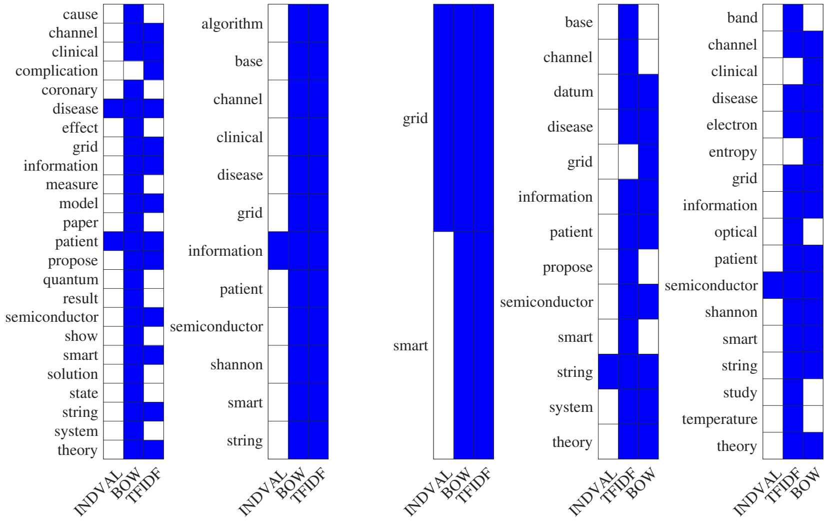


Fig. 6. Knowledge discovery results on ABS4. Left to right: Anatomy, Inf. Theory, Smart Grid, String Theory, Semiconductors.

Theory'; string ($I \simeq 90$) for class 'String Theory'; semiconductor ($I \simeq 77$) for class 'Semiconductors'. Conversely, the average INDVAL scores for ABS4 are: disease ($I \simeq 37$) and patient ($I \simeq 52$) for class 'Anatomy'; information ($I \simeq 41$) for class 'Information Theory'; grid ($I \simeq 95$) for class 'Smart Grid'; string ($I \simeq 89$) for class 'String Theory'; semiconductor ($I \simeq 77$) for class 'Semiconductors'.

E. Computational Results on Benchmark Datasets

Tables II–III show the average results for ℓ_1-SVMs and ν-SVMs for the remaining four benchmark datasets, respectively. As for the above discussion on large-scale classification, RBF ν-SVMs have been tested on INDVAL only. Results on benchmark datasets are rather in line with those obtained on synthetic datasets. The INDVAL strategy again leads to the smallest embedding space (a few dozens against a few thousands). For ℓ_1-SVMs there are, however, non-negligible shifts in terms of average accuracy, e.g. for 20NEWS the INDVAL method scores 85% against 91% (LSA, the second least performing method), whereas for WEATHERREPORTS and REUTERS8 the accuracy shifts with respect to the second least performing methods are around 4%. TDT2 is the only dataset in which the INDVAL performs rather equally with respect to the three competitors. By using ν-SVMs, performances are clearly improved and perfectly in line with competing techniques.

IV. CONCLUSIONS

In this paper, we proposed a novel approach for building an embedding space for text classification. Conversely to common techniques, which use all of the available words in the corpus, this technique leverages on a statistical index which quantifies the relevance of each word within each problem-related class and, by thresholding these scores, one gets a drastically reduced set of meaningful words. Under the GrC viewpoint, the latter can be interpreted as meaningful information granules

TABLE II
Results on benchmark datasets (ℓ_1-SVMs). In brackets, the ratio of selected features.

	BOW	**INDVAL**	**TF-IDF**	**LSA**
WeatherReports	95.37% (88/14426)	90.17% (3/119)	95.41% (88/14426)	94.34% (59/4000)
reuters8	98.37% (106/20728)	91.83% (8/164)	98.56% (158/20728)	95.56% (28/1000)
20news	93.71% (344/61188)	85.11% (24/424)	94.64% (303/61188)	91.48% (228/4000)
tdt2	99.51% (88/36771)	98.39% (8/855)	99.59% (111/36771)	99.26% (17/3500)

TABLE III
Results on benchmark datasets (ν-SVMs). In brackets, the ratio of selected features.

	BOW	**INDVAL**	**INDVAL** (RBF)	**TF-IDF**	**LSA**
WeatherReports	96.59% (2281/14426)	94.63% (6/119)	95.28% (11/119)	96.54% (2261/14426)	95.56% (998/4000)
reuters8	97.97% (2964/20728)	96.36% (14/164)	95.33% (8/164)	98.14% (3514/20728)	97.29% (210/1000)
20news	97.17% (9255/61188)	95.93% (85/424)	96.09% (107/424)	97.11% (11041/61188)	95.62% (1069/4000)
tdt2	99.55% (5915/36771)	99.52% (91/855)	98.98% (115/855)	99.51% (6441/36771)	99.47% (992/3500)

and can be analysed a-posteriori (XAI). The embedding space can be optimised by means of an evolutionary metaheuristic (a genetic algorithm, in this work) in order to simultaneously tune the classifier (SVMs, in this work) and further reduce the set of meaningful granules. Our approach has been tested on three synthetic, yet realistic, datasets and four well-known benchmark datasets against three other explicit embedding techniques (BOW, TF-IDF and LSA): especially when using ν-SVMs, the proposed method has comparable performances with respect to the three competitors, whilst strikingly outperforming them in terms of complexity of the embedding space. This can be interpreted as a further demonstration of how the INDVAL method is able to spot meaningful words that carry pretty much the same information with respect to the entire dataset. Whilst in this work we considered words (1-grams) as atomic information granules, one can extend the proposed method to general n-grams by properly tweaking Eqs. (4)–(6) in order to explore higher levels of granularity as well.

However, a major (intrinsic) drawback regards the enumeration of all words (or n-grams, in a general sense) within the corpus for evaluating A, B and I: this stage can be computationally demanding for very large corpora (e.g., Wikipedia) and future research avenues can investigate the possibility of a distributed implementation of the proposed method for building the embedding space for large corpora.

References

[1] J. T. Yao, A. V. Vasilakos, and W. Pedrycz, "Granular computing: perspectives and challenges," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1977–1989, 2013.

[2] S. B. Ayed, H. Trichili, and A. M. Alimi, "Data fusion architectures: A survey and comparison," in *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2015, pp. 277–282.

[3] D. Doran, S. Schulz, and T. Besold, "What does explainable ai really mean? a new conceptualization of perspectives," in *CEUR Workshop Proceedings*, vol. 2071, 2018.

[4] S. T. Mueller, R. R. Hoffman, W. Clancey, A. Emrey, and G. Klein, "Explanation in human-ai systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable ai," *arXiv preprint arXiv:1902.01876*, 2019.

[5] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[6] 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, 1975.

[7] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[8] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513 – 523, 1988.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[10] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[13] M. Dufrêne and P. Legendre, "Species assemblages and indicator species: the need for a flexible asymmetrical approach," *Ecological monographs*, vol. 67, no. 3, pp. 345–366, 1997.

[14] A. Martino, A. Giuliani, V. Todde, M. Bizzarri, and A. Rizzi, "Metabolic networks classification and knowledge discovery by information granulation," *Computational Biology and Chemistry*, vol. 84, p. 107187, 2020.

[15] A. Martino, A. Giuliani, and A. Rizzi, "The universal phenotype," *Organisms. Journal of Biological Sciences*, vol. 3, no. 2, 2019.

[16] ——, "(hyper)graph embedding and classification via simplicial complexes," *Algorithms*, vol. 12, no. 11, 2019.

[17] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[18] M. McTear, Z. Callejas, and D. Griol Barres, *The conversational interface*. Springer, 2016, vol. 6, no. 94.

[19] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological review*, vol. 104, no. 2, p. 211, 1997.

[20] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[21] J. Zhu, S. Rosset, R. Tibshirani, and T. J. Hastie, "1-norm support vector machines," in *Advances in neural information processing systems*, 2004, pp. 49–56.

[22] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.