Cognitive Computation
https://doi.org/10.1007/s12559-023-10132-9

Prototype Theory Meets Word Embedding: A Novel Approach for Text Categorization via Granular Computing

Enrico De Santis[1] · Antonello Rizzi[1]

Received: 14 July 2020 / Accepted: 6 March 2023

Abstract
The problem of the information representation and interpretation coming from senses by the brain has plagued scientists for decades. The same problems, from a different perspective, hold in automated Pattern Recognition systems. Specifically, in solving various NLP tasks, an ever better and richer semantic representation of text as a set of features is needed and a plethora of text embedding techniques in algebraic spaces are continuously provided by researchers. These spaces are well suited to be conceived as conceptual spaces in light of the Gärdenfors's Conceptual Space theory, which, within the Cognitive Science paradigm, seeks a geometrization of thought that bridges the gap between an associative lower level and a symbolic higher level in which information is organized and processed and where inductive reasoning is appropriate. Granular Computing can offer the toolbox for granulating text that can be represented by more abstract entities than words, offering a good hierarchical representation of the text embedded in an algebraic space driving Machine Learning applications, specifically, in text mining tasks. In this paper, the Conceptual Space Theory, the Granular Computing approach and Machine Learning are bound in a novel common framework for solving some text categorization tasks with both standard classifiers suited for working with $\mathbb{R}^n$ vectors and a Recurrent Neural Network (RNN) — an LSTM — able to deal with sequences. Instead of working with word vectors, the algorithms process more abstract entities (concepts), where patterns, in a first approach, are obtained through the construction of a symbolic histogram starting from a suitable set of information granules, representing a document as a distribution of concepts. For the RNN case, as a further novelty, a text is represented as a random walk over prototypes within the conceptual space synthesized over a suitable text embedding procedure. A comparison of the performance and a critical discussion are offered for both a neural embedding technique and the well-known LSA, showing how the conceptual level leads also to Knowledge Discovery applications.

Keywords Conceptual spaces · Granular computing · Text classification · Word Embedding · Conceptual embedding · Long Short Term Memory

Introduction

The human brain can be thought as the best pattern recognizer in the known universe. Since our early childhood, we have been observing patterns in the objects around us (e.g., flowers, toys, pets and faces). Learning patterns also reinforces, and is reinforced by, the acquisition of language. It is well known that most 5-year-old children are already able to recognize digits and letters [1]. At the same time, scientists, engineers and practitioners know that designing a general-purpose machine for Pattern Recognition (PR) able to contend in performances the brain remains, today, an elusive goal. PR, intended as a superordinate field, involve disciplines as Cognitive Sciences, Psychology, Artificial Intelligence (AI), and for some extent, Neuroscience, Linguistics and Philosophy. As a discipline which studies the ability of discovering and recognizing regularities in observations, PR can help to understand how human perception works and to discover the secrets behind the ability to gain new knowledge and to exploit it in appropriate ways. So forth, besides giving more insights into the study of human senses and the neural system, PR allows to build up automated systems adopted in medical diagnosis, industrial inspection, personal

✉ Enrico De Santis
enrico.desantis@uniroma1.it

Antonello Rizzi
antonello.rizzi@uniroma1.it

[1] University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy

Published online: 23 March 2023

identification, man-machine interaction, Natural Language Processing (NLP) tasks, etc.

From its own side, an open issue in Cognitive Science is the causal relation of low level phenomena occurring in the senses and the nerves onto higher levels of understanding and conceptual thinking [2, 3]. Interestingly, it is an open issue even in automated mechanical PR, even if we are dealing with sensors, actuators, processing units etc.

In fact, one of the main problems that affect both disciplines (namely automated PR and Cognitive Sciences), each one within its own phenomenology, is the "representation" [4, 5]. In other words, Cognitive Science, in studying the cognitive activities of humans and other animals, provides us with a set of explanatory theories and even a set of constructive prescriptions that can aid to design artifacts like robots, animats, and chess-playing programs, with the aim of accomplishing various cognitive tasks. A key issue is the representation of information or data, in such a way that the cognitive system can be modeled, starting from stimuli coming from biological or digital sensors to high-level processing capabilities, for example, to perform tasks on higher semantic levels. It can be affirmed, without pretense of completeness, that self-organizing phenomena of the physical world are relevant also for understanding cognitive processes [6]; hence, some properties of cognitive systems are in resonance with the ones attributed to Complex Systems. Even language and text production can be thought as activities of complexly organized brains [7] and semantic meaning can be hidden in the middle-ware of the complexity around us. The engineer Alfred Korzybski, the inventor of General Semantics, giving great importance to language and the use of words, in 1933 affirmed that thinking is a matter of multilevel order of abstraction and content is a declination of the structure with complex relationships [8]. The multilevel order of abstraction can be found even in the organization of most Complex Systems where emergent properties and vertical information processing generate new abstract levels dominated by its own semantic content.

From a computational point of view Granular Computing (GrC) is the umbrella term to cover any theories, methodologies, techniques, and tools that make use of *information granules* in complex problem solving [9, 10]. Information granules are atomic units [11] that naturally give rise to hierarchical structures: the same problem or system can be perceived at different levels of specificity (detail), depending on the complexity of the problem, the available computing resources and the particular needs to be addressed [9, 12]. Some authors (e.g., W. Pedrycz) conceive GrC as a conceptual and algorithmic platform supporting analysis and design of human-centric intelligent systems [13]. Zadeh, the scientist who made fuzzy logic great, considers GrC as a basis for computing with words, i.e., computation with information described in natural language [14, 15]. For example, the text in a book can be seen as an increasing granulation of the information content starting from the alphabet letters and ending with the aggregation of concepts and topics, passing through "mesoscopic" structures such as words, sentences, paragraphs, chapters and so on. In this regard, E. G. Altmann et al. affirm that [16]: "literary texts are an expression of the natural language ability to project complex and high-dimensional phenomena into a one-dimensional, semantically meaningful sequence of symbols. For this projection to be successful, such sequences have to encode the information in form of structured patterns, such as correlations on arbitrarily long scales".

Moreover, dealing with text, in automated PR systems the representation problem becomes more difficult because text is intrinsically structured at various levels, while classical PR problems solved by standard Machine Learning (ML) approaches need to work with the $\mathbb{R}^n$ vector space geometry. In general, the GrC approach allows designing automated PR problems able to deal directly with structured or unconventional input domains [17]. Hence, a challenging task is finding effective models and algorithms able to represent and process a set of samples coming from a structured domain.

The aim of the current work is twofold:

– from a more general point of view, it tries to investigate how to bridge the gap between some findings in Cognitive Science (the Conceptual Spaces and Prototype Theory [3, 18] and so forth) and the GrC approach in light of the problem of representation of text excerpts in text mining problems;
– from a specific point of view, the objective of the current study is to experiment some text embedding methods through a GrC approach, known as symbolic histograms [19, 20], in solving two specific text classification problems through some standard Machine Learning algorithms able to process n-tuple of real numbers or more structured objects, such as sequences.

It is well known that in PR applied to text data a traditional representation approach consists in embedding words or documents in a mathematical space with useful algebraic properties, such as a linear vector space, also known as feature space. 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) [21], grounded, in turn, on the *distributional hypothesis*: similarity of meaning correlates with similarity of distribution. After all, Wittgenstein claimed in his *Philosophical Investigations*, that "the meaning of a word is its use in the language" [22]. In other words, as the American linguist Z.S. Harris sustained: "words that are used and occur in the same contexts tend to purport similar meanings" [23] or paraphrasing the British J. R. Firth "a word is characterized by the company it keeps" [24]. In ML, specifically in document classification or even in Computer Vision [25], the approach is known as "bag of words (BoW)" — sometimes known as *surface form* — pointing out the fact that the text is represented as the frequency of occurrence of each word building a feature space for training a classifier [26], disregarding grammar and even the order of words. The methodology is heavily adopted in Information Retrieval (e.g., the so-called traditional Vector Space Model (VSM) [27]) and text mining but is well known that it has some limitation, such as (i) the orthogonality, (ii) the construction of the vocabulary that requires a careful design due to its size, (iii) the sparsity of the model and the lack of context due to the discarding all information brought by surrounding words.

Researchers have attempted to address the representations of natural language that are capable of capturing meaning through what they call *semantic spaces*, a set of language models that adopt the DS. For example, the Hyperspace Analogue to Language (HAL) [28] is a method for creating a simulation that exhibits some of the characteristics of a human semantic memory finding lexical co-occurrences by moving a window of length l over the corpus. HAL allows representing words as vectors.

In general, authors refer to the word-context models as explicit models, while some family of transformations of the underlying data structure leads to implicit representations [29]. Canonical co-occurrence models are simpler to implement and they work well within standard ML pipelines. However, they possess a number of drawbacks, for example, sparsity (a lot zeros due to Zipf's law) and high dimensions when dealing with huge corpora with large vocabularies. A simple frequency count, for example, does not embed intrinsically the fact that two words have the same meaning (synonymy) because they are treated as named entities, that is, they are symbols. Moreover, contexts can be similar too, or high-correlated. Furthermore, these raw representations can be very noisy.

In order to avoid some drawbacks, a number of implicit representations are provided in literature, some of which are known as dense representations, because they reach a non-sparse representation, often in a reduced feature space. The most adopted methodologies, in practice, use an implicit representation of features in a latent space where latent features are computed starting from the distributional models. For example, Latent Semantic Analysis (LSA), representing the text in a latent space through a set of linear algebraic transformations, aims at constructing a rich semantic space. LSA is obtained by means of (linear) matrix decomposition procedure known as Singular Value Decomposition (SVD), allowing dimensionality reduction (truncated SVD) and noise filtering. The dense embeddings produced by SVD sometimes perform better than the raw ones (grounded on PPMI matrices) on semantic tasks like word similarity. Various aspects of the dimensionality reduction contribute to improved performance. If low-order dimensions represent unimportant information, the truncated SVD may be able in removing noise. By reducing the input dimension, the truncation may also help the models to generalize better to unseen data. Due to interesting, and in some ways unexpected, properties, LSA has also been proposed as a cognitive model for human language use [30, 31]. Other techniques adopt other matrix factorization methods, such as the non-negative matrix factorization (NMF) or ML methods such as GloVE [32], which is based on a regression technique.

Recently, in technical literature there are some powerful neural approaches, for example, the *word2vec* algorithm [33, 34], which embeds the meaning of text in a similar way to HAL (windowing), but constructing a dense representation training a shallow Artificial Neural Network (ANN) — e.g., Skip-gram with negative sampling (SGNS). More recent approaches in neural language embedding adopt sophisticated Recurrent Neural Networks (RNN) bound with attention mechanisms for language modeling, such as the Bidirectional Encoder Representations from Transformers (BERT) [35] and related architectures. Another technique that uses an external corpus to build a semantic space is the Explicit Semantic Analysis (ESA) [36], where words are represented as vectors and each entry is a Wikipedia article. In other words, each Wikipedia article is a kind of concept and words are embedded in a "concept space". Hence, some attempts in embedding "meaning" and working with concepts are based on the so-called Bag of Concepts (BoG) [37] that, rather than identifying features directly with some surface form, utilizes some artifices to make practical the intuition that the meaning of a document can be approximated by the union of the meanings of terms appearing in the document itself. There are a number of practical implementations of BoC that uses concept vectors. They differ on how they construct the concept space, for example, adopting implicit or explicit representations, such as Word-net [38] like approaches or hyper-linked encyclopedic textual corpora.

In this paper, as concerns the textual conceptualization, we deal with a simple type of BoC useful for building a suitable feature space, where both traditional ML algorithms or advanced ones, such as RNN — for example, a Long Short Term Memory (LSTM) — can safely operate.

In doing so, as stated above, we adopt GrC as a general toolbox, while the road-map of the proposed approach is grounded by a specific approach mediated from Cognitive Psychology and in general from Cognitive Science, that is the "Conceptual Space" [3]. The theory of Conceptual Spaces is a modern extension of Prototype Theory developed by Rosch [39, 40]. P. Gärdenfors affirmed that the problem of representation in Cognitive Science, thus the problem of the vertical information processing where stimuli and senses data become high-level thinking and concepts, is due to the lack of a middle level between the Sub-conceptual Representations based on *associations* and the Symbolic Representations where rational thinking operate. This level is the *Conceptual Level*, a bridge where information is organized in a smooth space and where the notion of prototype and similarity (intended as a mathematical distance) allows to deal with concepts and properties (as a particular instance of concepts) in representing real-world objects. Concepts are particular "natural" regions of the Conceptual Space [3].

The proposed methodology foresees first the embeddings of words in a given corpus through either (i) the neural word embedding technique — *word2vec* — that is based on the association between words and contexts computed through a neural technique or (ii) the classical LSA. The aim here is to build a semantic space — a Conceptual Space — were words coded by vectors are embedded.

The space of word vectors is, thus, partitioned in "natural" regions (Voronoi regions) through a clustering algorithm, where regions are intended as a semantically homogeneous containers around its prototype. Once constructed the Conceptual Space, each word in a given document takes part in a new representation, known as a symbolic histogram.

Symbolic histograms [17] is an embedding technique, where a pivotal role is played by a set of meaningful and recurrent substructures in the original data space, often adopted for representing other structured objects lying in a non-metric structured space, such as graphs, sequences, strings, and images. In the current approach each document in a given corpus is represented as a symbolic histogram.

Specifically, concepts are represented by symbols (i.e., prototypes). In this sense, the vectors correspond to subsymbols [41] that are transformed into symbols through a process characterized by information loss.

In other words, a documents is represented as a probability distribution on a set of alphabet symbols — we will call representatives of concepts among the Conceptual Space — used as feature vector for feeding a classification algorithm. Specifically a comparison will be offered among Random Forest (RF), a Support Vector Machine (SVM) and an advanced RNN model able to deal with sequences (LSTM). In the last case, as further novelty, instead of a classical features space where features are concepts, the RNN processes sequences of concepts, that is, ultimately, a new representation of a document. By the way, Wiggins argues [42] that learning is not only a matter of acquiring static co-occurrences, unless it includes generalization and the ability of processing sequences of events or even sequences of concepts.

In light of the Conceptual Space Theory this approach adds a *middle layer* in the representation/embedding of text in documents. Hence, starting from a *sub-conceptual layer* where associations dominate the representation (neural embedding or LSA), the construction of the alphabet — obtained at training time — is based on a conceptual organization of the underlying *associative layer*, where are elicited a set of (read a small number of) prototypes that, in turn, offer a symbolic level used to build the embedding representation by symbolic histograms. The proposed embedding allows representing documents in a smaller feature space in term of dimension compared to BoW approaches, providing a good performance for further recognition tasks. Moreover, the new feature space constructed on the top of the granulation of the semantic information contained in the word embedding model is a classical real-valued feature space, allowing the adoption of standard ML algorithms (as mentioned earlier). This is a strong point of the proposed approach. It is worth to note that the proposed methodological framework opens the way to knowledge discovery applications and, in general, to the Explainable AI paradigm [43, 44]; a fact not so obvious for the modern neural architectures used in the NLP context.

The paper is organized as follows.

In "Related Works" a brief overview of related works is reported. In "Background: Prototypes and Conceptual Spaces" the Conceptual Space Theory and the Prototype Theory are outlined. In "Methods" is presented the adopted approach and the problem framing. The description of the data sets for the experiments and the main results are provided in "Experiments". Lastly, conclusions are drawn in "Conclusions".

Related Works

The symbolic histograms technique within the GrC model is widely adopted in many PR tasks [17], such as online handwriting recognition [45] or protein classification [46]. This technique is heavily adopted when dealing with unconventional structured data, such as graphs, for example, performing frequent substructures mining in graphs seriation [47, 48] and classification methods [19, 49]. In the specific field of text mining and text categorization GrC is found very promising [50, 51]. Concerning Knowledge Discovery applied to text mining problems, authors in [52] deal with concept formation and concept relationships identification through constructing a granules' network. An automatic text categorization system is proposed in [53] considering a document as an ordered sequence of words, proposing a system able to automatically mine frequent terms, considering as a term not only a single word, but also a sub-sequence of a few consecutive words (i.e., *n*-grams). The categorization system is tailored to process sequences of atomic elements (i.e., encoded words) by means of an embedding procedure based on clustering and adopting the symbolic histograms technique.

Many authors have adopted the BoC terminology referring to some technique for dealing with more general representations of words or sentences rather than the BoW model. In [37] authors adopt a particular technique within the BoC paradigm called Random Indexing, training a SVM with good results. Random Indexing is even used in [54] along with the Holographic Reduced Representation, previously proposed in cognitive models, which can encode relations between words. In [55] authors propose the cross-language concept matching (CLCM) technique, which relies on Wikipedia inter-language links to convert concept vectors from the Spanish to the English language space. They synthesize a classifier of text documents, represented as vectors in spaces of Wikipedia concepts, and provide an analysis of its suitability for classification of Spanish biomedical documents when only English documents are available for training. An approach, called Mined Semantic Analysis, is proposed in [56]. The study tries to address and mitigate problems arising in concept space models, such as the limitation to direct association between words and concepts, affecting the ability of models to transfer the association relation to other implicit concepts which contribute to the meaning of these words. The particular BoC paradigm is able to build concepts through concept rich encyclopedic corpora, even exploring the "see also" link graph in Wikipedia. A different declination of the BoC technique is provided in an interesting investigation [57] in line with the current research work, where authors creates concepts through clustering word vectors generated from *word2vec* and using the frequencies of clusters' representatives to compute document embedding vectors. They propose a suitable weighting scheme, such as the concept frequency-inverse document frequency. Through these data-driven concepts, the method allows semantically similar words to be preserved effectively in a suitable document proximity measure. A related BoC approach is proposed in [58] solving an emotion estimation task from text excerpts, characterized even by youth slang, an ambiguous and difficult task when using existing dictionaries, such as thesaurus. In an interesting work [59] authors try to outperform the lack of concept overlapping in some text mining tasks,resulting in a data sparsity problem, proposing an efficient vector aggregation method, grounded on a neural embedding model, able to generate fully continuous BoC representations.

Background: Prototypes and Conceptual Spaces

Humans are extremely efficient at learning new concepts. Cognitive Science is interested in how to model concept learning starting from the ability of humans to learn concepts from a few examples. On the other side, ML, along with the data-driven approach, uses its own models to learn from examples. The main approaches in modeling concept learning are the one known as "symbolic" and the one known as "associationist" [3]. The symbolic approach starts from the assumption that cognitive systems can be described as Turing machines. Hence, cognition is a matter of symbol manipulations. Within the associationists paradigm associations between different kinds of information elements carry the main burden of representation [60]. The Swedish cognitive scientist P. Gärdenfors sustains that connectionism — the ANN approach — is a special case of associationism [3]. However, the same author admits that there is no unique correct way of describing cognition. There are phenomena that neither the symbolic representation nor the associationist appears to offer appropriate modeling tools. He proposes the "Conceptual Spaces", as the framework placed in the middle of the two main approaches, that is the most appropriated for modeling concept learning and representation. The theory of conceptual spaces, due to its versatility and capability even when in dealing with high-dimensional spaces, has been extended together to the 3-way formal analysis to investigate phenomenal consciousness, within a quantum framework [61]. By the way, the three approaches mentioned can be seen as three levels of representations of cognition with different scales of resolution or "granulation". Conceptual Spaces are able to geometrize the thought, because world objects are embedded in a geometric space where the notion of distance, region and prototype can be used to model concepts [62]. Actually, the embedding of real-world objects, through a series of suitable measures on them, is a normal procedure in automated PR systems. Measurable properties in automated PR and ML are called "features", while in Conceptual Space theory they are called "quality dimensions". However, neither with the symbolic approach (as an example, the first-order logic) nor with the associanist/connectionist approach, it is easy to deal with similarities [3]. While the associationist approach suffers for the black-box problem — think to ANN — the symbolic approach seems not working at the appropriate abstraction level, for example, lacking in creative induction, new knowledge creation and basically being not able to perform conceptual discoveries. Moreover, the symbolic approach lacks in automatic management of semantic and meaning. On the contrary, in Conceptual Spaces induction can be derived "naturally" from the metric properties of the underlying algebraic space, allowing what is known in automated PR and ML as "generalization capability". That is, the capability of generalizing predictions on unseen data. By the way, P. Gärdenfors asserts that the symbolic level is not completely non-significant and it depends strictly on the underlying conceptual level [3].

An important distinction, useful in the context of the current work and due to Palmer [63], is about *intrinsic* and *extrinsic* representation. The former, is valid when the representing relation has the same inherent constraints as its represented relation. For example, in the isomorphism between the dimension "age" and the "height" of a bar in a chart, the structure of the represented relation (age) is intrinsic in the representing relation (height). In contrast, extrinsic representations must be accompanied by a rule that specifies how the representation is to be interpreted; such a rule provides the "meaning" of the representation. On the symbolic level, atomic concepts are not modeled, just named by the basic symbols. Even if complex concepts can be constructed through compositions of logical or syntactical rules, they remain extrinsically represented. In DS the BoW model considers intrinsically words as named entities, that is, as symbols with no further relational structure and the frequency count for representing documents is a symbol count. This leads to the synonymy problem.

Within the Conceptual Space theory the geometric characteristics of the quality dimensions are utilized to introduce a spatial structure for properties:

Criterion P: A Natural Property is a Convex Region in Some Domain A subset C of a conceptual space S is said to be convex if, for all points x and y in C, all points between x and y are also in C [3]. It's worth to note that *Criterion P* assumes that a notion of "betweenness" among objects is provided when each concept is represented as a point in a given space [3]. Convexity, for example, is mantained for the color naming and the three-dimensional representation of the color space. It is worth to note that properties defined by the *Criterion P* are a special case of concept.

Studying the phenomenology of colors and its perceptual representation in Cognitive Psychology E. Rosch and collaborators defined the Prototype Theory providing us with a model of categorization [39, 40]. The main idea in this theory is that within a category of objects, like those instantiating a property or a concept, certain members are judged to be more representative of the category than others. That prototype representation of a category is generally taken to be a generalization or abstraction of a class of instances falling into the same category [64]. In cognitive linguistics a prototype is a typical instance of a category and other elements are assimilated to the category on the basis of perceived similarity to the prototype [65].

The appealing feature of Conceptual Space lies in the underlying algebraic structure, that can be metric. This means that are fulfilled all or some properties of metric spaces [66]. A natural partition of such spaces is the Voronoi tessellation, a particular tessellation of the space based on a simple rule. If $p_1, p_2, ..., p_n$ are prototypes of a space S, the Euclidean distance $d_E(p, p_i)$ among a point p and the prototypes p_i can be defined. If we now state that p belongs to the same category as the closest prototype p_i, it can be shown that this rule will generate a partitioning of the space, the so-called Voronoi tessellation [67]. Not every distance metric (e.g., Manhattan or in general the Minkowski distance for some values of its parameter) generates a set of regions that fulfill the convexity property, however, for the Euclidean distance this property holds. Among the many methods used to compute Voronoi cells [67], the clustering algorithm k-means can help, in an unsupervised fashion, to compute centroidal Voronoi regions, where centroidal points are the centroids of the regions [68]. Hence, centroids are isomorphic to prototypes of some Conceptual Space. Thereby, depending on the nature of the space S (i.e., the nature of dimensions), the Conceptual Space becomes a semantic space (here the term semantic is used in a weak interpretation). In this way, the Voronoi tessellation provides a constructive geometric answer to how a similarity measure, together with a set of prototypes, determines a set of categories [3]. The Conceptual Spaces have been adopted even in trying to pragmatically untie the knot of semantics, intended as the relationship between an expression and an extralinguistic reality, within the riverbed of the cognitive semantics. The last assumes that the referents of words are identified with conceptual structures in people's minds. However, semantics is a huge field of study where numerous discipline converges, such as Semiology, Semiotics, Linguistics, Psychology, Pragmatics, Communication, and Philosophy of Language. In Linguistics and, specifically, in Computational Linguistics the meaning, and in general the semantic content of a word or expression, assumes a specific way of being related to a context, which is empirical and measurable. For example, it is common to refer to space generated by the BoW model as a semantic space, specifically, a mathematical space grounded on the DS.

Methods

The approach presented in details hereinafter is an attempt of systematizing the theory of Conceptual Spaces with a specific declination of the BoC paradigm built upon the background of the GrC approach. The overall processing pipeline is composed by several steps where information extracted from the text is granulated, and information granules are adopted, in turn, in constructing a new embedding space grounded on the symbolic histogram technique. The main objective is to find an economic representation of documents as BoC for classification purposes, hence for text categorization. Following the scheme proposed in Fig. 1, given a corpus of documents, the first step is to perform the embedding of words in an algebraic space, called in the following Conceptual Semantic Space (CSS). The embedding of words

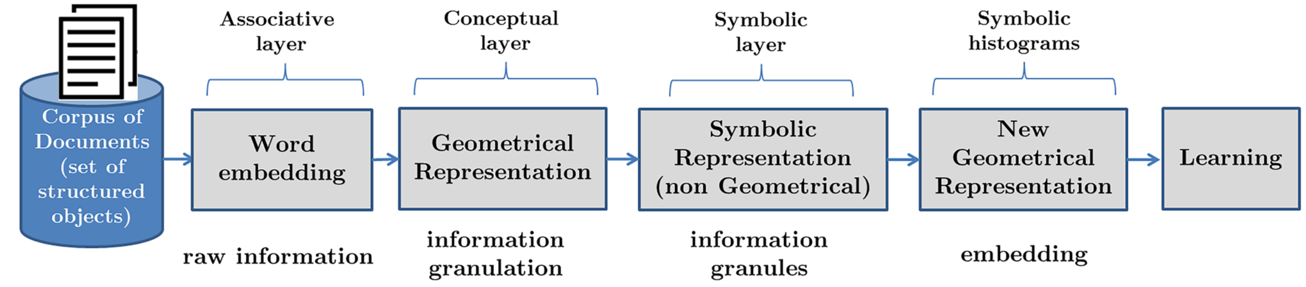


Fig. 1 Information processing scheme

can be performed through various methodologies outlined in "Introduction''. In this work the LSA and the neural word embedding through the *word2vec* algorithm are performed.

The word embedding step is grounded on the co-occurrences (collocates) of words obtained through a context window of suitable length. In the case of LSA, the word vectors within the reduced latent space are obtained on the top of a BoW model with TF-IDF weighting, where contexts are documents. Hence, this layer fits with the "associative layer" [3]. The word vectors generate a vector semantic space endowed with the standard Euclidean norm, thus, it is defined a dissimilarity measure based on the Euclidean distance [69]. In the case of neural embedding through the *word2vec* algorithm, word vectors are directly obtained by the training procedure, ready to be further processed. Instead of using directly the word vectors, a Voronoi tessellation is computed, where each region coincides with a concept whose instances are linked by semantic relations. The Voronoi tessellation is obtained by computing the representatives — the prototypes — through a clustering algorithm. The *k*-means algorithm used in the following, but in principle other clustering algorithms can be adopted. This step embodies the "conceptual layer" that is the layer interposed between the "associative" and the "symbolic" one. Figure 2 depicts an example of CSS obtained for a corpus of scientific paper abstracts ("Abstracts" data set hereinafter), with four classes ("Anatomy", "Information Theory", "String Theory", "Semiconductors") of which a deep description will be given in the experiment section. The CSS in Fig. 2 is synthesized by a Voronoi tessellation in $k = 8$ regions, where the prototypes are highlighted by crosses. Dots represent words embedded (initially in a

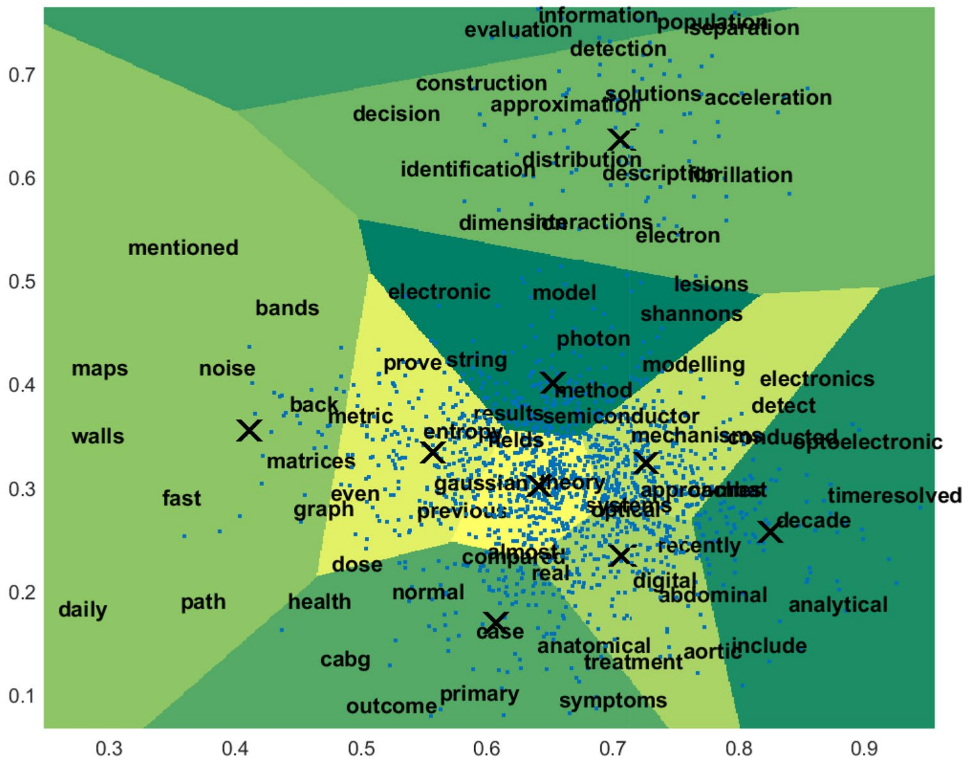


Fig. 2 Centroidal Voronoi regions of the Conceptual Semantic Space for the Abstracts data set obtained through the *k*-means. Dots are words computed with the *word2vec* algorithm (word embedding) and projected in a bi-dimensional space through PCA. Crosses are the prototypes for each region. In this explanatory example the number of conceptual regions are $k = 8$

100-dimensional space) through the *word2vec* algorithm. Principal Component scores are computed for dimensionality reduction with the aim of data visualization.

Accordingly, the conceptual semantic layer is the ground for a symbolic representation of documents, namely each word is abstracted by its concept computed measuring the semantic similarity of the vector representation of words and the prototypes on the underlying CSS. Thus, documents are represented as discrete probability distributions on concepts.

Prototypes are intended, therefore, as symbols of a suitable alphabet $\mathcal{A}$ of concepts used for the symbolic representation.

Let $\mathcal{H} = \left\{D_1, D_2, ..., D_L\right\}$ be a corpus with L documents, where each document D, $D = \left\{w_1, w_2, ...w_{|D|}\right\} \in \mathcal{H}$, hence D is a collection of words $w_i, i = 1, 2, ..., |D|$ in a vocabulary $\mathcal{V}$. The prototype $c_j \in \mathcal{A}, j = 1, 2, ..., k$, abstracting a concept of a region $\mathcal{R}_j, j = 1, 2, ..., k$ of the Conceptual Space P, defines what we can call a symbol of a suitable alphabet $\mathcal{A}$. It is worth to note that the parameter k defines the level of granulation of the CSS. Each document can be suitably represented by some statistics on the alphabet symbols $c_i \in \mathcal{A}$, namely, if the prototypical region pertaining the partition obtained by word embedding vectors is a "concept", the document is represented as a "bag of concepts". In the limit where the number of prototypes (aka the cardinality of the alphabet $\mathcal{A}$) equals the number of words in the corpus of documents $\mathcal{H}$, the standard BoW model is recovered. It is worth to note that the symbol c_i is obtained by a suitable mapping $\mathcal{M}$ from the underlying word vector $\mathbf{w}_i \in \mathcal{W}$, obtained through the word embedding, and concepts in $\mathcal{A}$, that is $\mathcal{M} : \mathcal{W} \rightarrow \mathcal{A}$, where $\mathcal{M}(\mathbf{w}_i) = c_j, j = 1, 2, ..., k$, for the i-th word within a document.

Figure 3 depicts the word clouds for a CSS $\mathcal{P}$ partitioned in $k = 8$ semantic regions. The thickness of each word is proportional to the similarity (based on the Euclidean distance) to the prototype computed as the centroid of the conceptual region.

Moreover, in Fig. 4 the symbolic histograms for four documents pertaining the classes "Anatomy", "Information Theory", "String Theory", and "Semiconductors" of the Abstracts data set are reported. The length of a bar represents the number of occurrences of each one of the (ten) symbols (prototypes) for a given class.

The symbolic histogram representation allows naturally to embed documents in a vector space giving the way for classification or regression ML algorithms. However, there are possible other representations. Instead, it is possible to build a *centroidal* prototype for each document simply computing the average of word vector representations of prototypes. In other words, instead of having a prototype derived from a count histogram, we have an average value of word vectors prototypes associated to each word in a document. This alternative will be introduced more formally below. Another quite different representation of documents adopting prototypes is conceiving a document as a sequence of

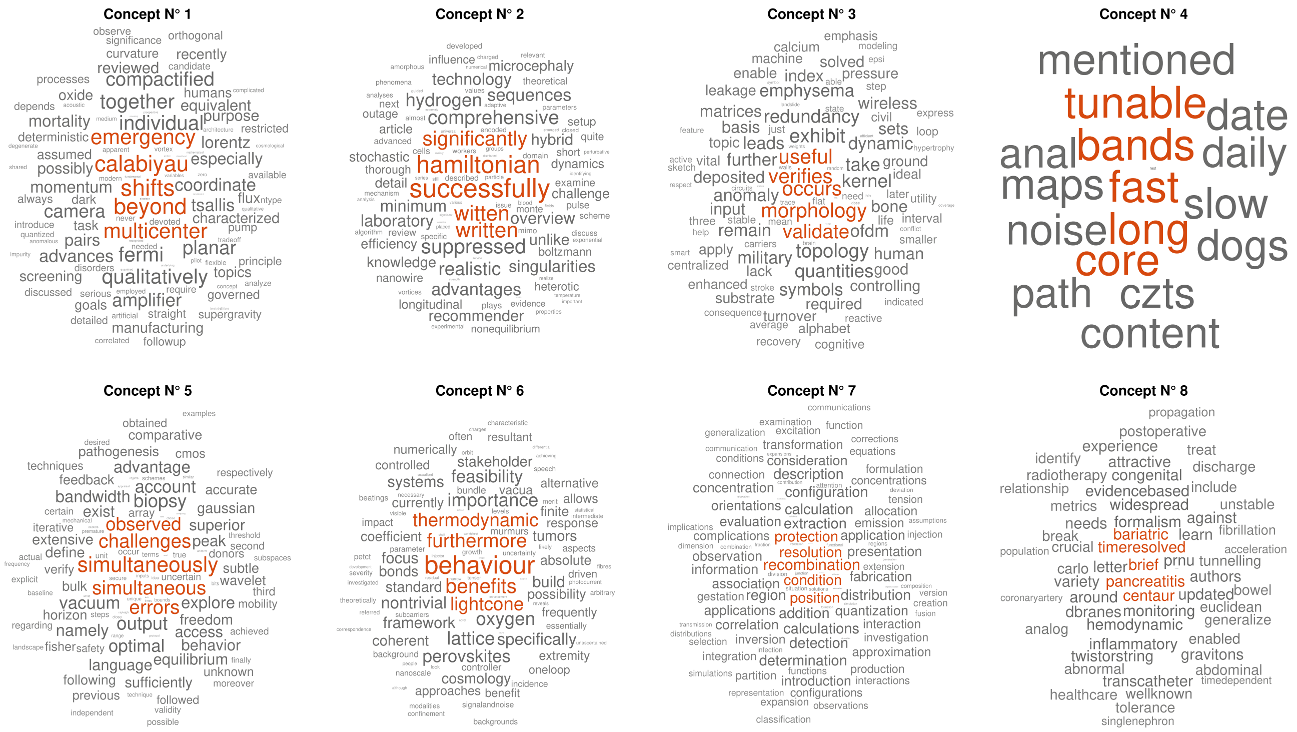


Fig. 3 Concept cloud for each one of the $k = 8$ conceptual regions for the Abstracts data set. The thickness of each word is proportional to the similarity (Euclidean distance) to the prototype computed as the centroid of the conceptual region

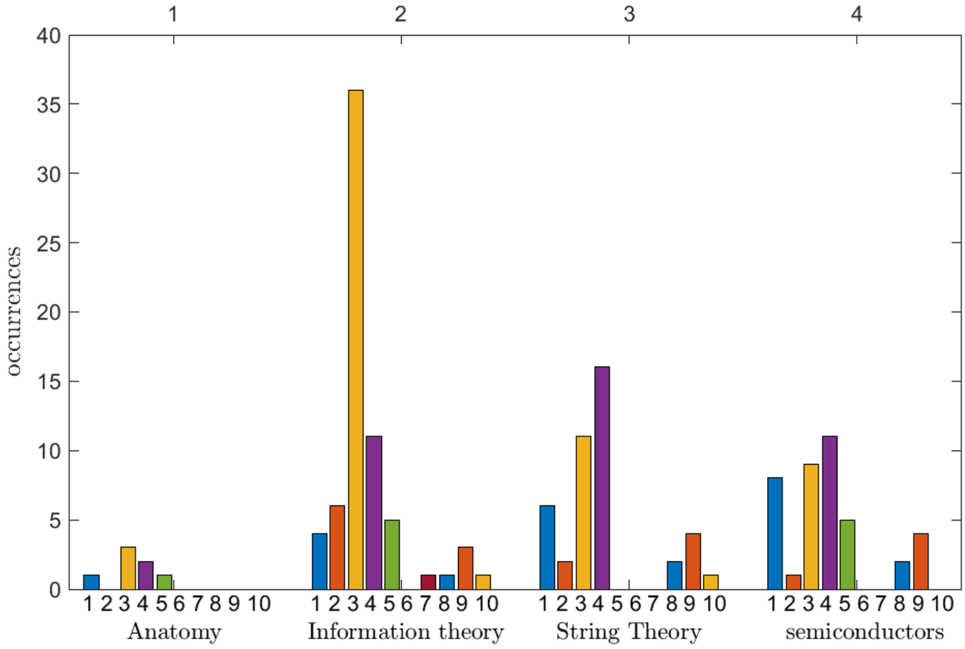


Fig. 4 Symbolic histogram for four documents pertaining the classes "Anatomy", "Information Theory", "String Theory", "Semiconductors" of the Abstracts data set

words, hence as a sequence of the prototypes associated to words. Namely, a document is represented by a sequence of concepts, where concepts are semantic abstraction of words. Words, in this setting, are fine-grained representation, while concepts pertain to coarser one. This new representation gives the way for sequence-based ML algorithm, such as the deep learning-based LSTM. Interestingly, this sequence-based representation allows framing a document as a random walk of concepts, instead of a random walk of words.

Classification Problem Framing with Symbolic Histograms

A general classification problem instance is defined as a triple of disjoint sets, namely training set ($\mathcal{S}_{tr}$), validation set ($\mathcal{S}_{vs}$), and test set ($\mathcal{S}_{ts}$). Given a specific parameters setting, a classification model is built based on $\mathcal{S}_{tr}$ and it is validated at training stage on $\mathcal{S}_{vs}$. The generalization capability of the optimized model (the one synthesized by the whole training procedure) is finally measured on $\mathcal{S}_{ts}$. Hence, given a corpus $\mathcal{H} = \left\{D_1, D_2, ..., D_L\right\}$ composed by L documents D, we have

$$\mathcal{H} = \left\{\mathcal{S}_{tr} \cup \mathcal{S}_{vs} \cup \mathcal{S}_{ts} | \mathcal{S}_{tr} \cap \mathcal{S}_{vs} = \emptyset, \mathcal{S}_{tr} \cap \mathcal{S}_{ts} = \emptyset, \mathcal{S}_{vs} \cap \mathcal{S}_{ts} = \emptyset\right\}. \tag{1}$$

The CSS $\mathcal{P}$ is conceived as a hard partition of order k, as a collection of k disjoint and non-empty clusters, $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k\}$. In this study the partition is obtained through the well-known k-means algorithm [70, 71]. Each cluster $\mathcal{C}_i \in \mathcal{P}$ is synthetically described by a representative or *prototype* element, which we denote as $\mathbf{c}_i = R(\mathcal{C}_i)$; let $R(\mathcal{P}) = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_k\}$ be the set of representatives of the partition $\mathcal{P}$.

The definition of a cluster representative is well defined for vector feature spaces equipped with an algebraic structure, where it can be simply computed as the average vector in a set of real-valued vectors, i.e., $\mathbf{c}_i = \frac{\sum_{\mathbf{w}_i \in \mathcal{C}_i} \mathbf{w}}{|\mathcal{C}_i|}$.

Alternatively, the representative of $\mathcal{C}_i$ can be computed as the element $\mathbf{c}_i$ that minimizes the sum of distances (MinSOD) [72]:

$$\mathbf{c}_i = \arg\min_{\mathbf{w}_j \in \mathcal{C}_i} \sum_{\mathbf{w}_k \in \mathcal{C}_i} d(\mathbf{w}_j, \mathbf{w}_k). \tag{2}$$

In this case the representative is an object of the cluster, that is $\mathbf{w}_j \in \mathcal{C}_i$. Note that computing the MinSOD does not require an algebraic structure, demanding just the definition of a dissimilarity measure. From this point of view, the MinSOD representative is much more general, and can be applied in any data domain.

Finally, each prototype $\mathbf{c}_j$ identifies a centroidal Voronoi region $\mathcal{R}_j$ through the Euclidean distance $d_j = \left\|\mathbf{w}_i - \mathbf{c}_j\right\|_2$, with $\mathbf{w}_i \in \mathcal{W}$, where $\mathcal{W}$ is the set of word vocabulary vectors.

Embedding Words

As concerns the word embedding procedure, a comparison will be offered between the word embedding obtained through LSA, by means of the SVD decomposition, and the neural embedding, by means of the *word2vec* algorithm — see "Introduction''. In particular the two techniques allow new ways to represent each word $w \in \mathcal{V}$ through suitable vectors $\mathbf{w} \in \mathcal{W}$, just considering a mapping $\Phi : \mathcal{V} \rightarrow \mathcal{W}$, from the set of vocabulary words to word vectors, i.e., $\Phi(w) = \mathbf{w}$.

The Symbolic Histogram Construction

In abstracting a concept, hence a prototype for a word of a given document, it is necessary to associate a prototype to each word of a given document. Hence, given a document $D = \left\{ w_1, w_2, ...w_{|D|} \right\} \in \mathcal{H}$ as a collection of words w_i, and its vector representation $\Phi(w_i) = \mathbf{w}_i$ first, the nearest cluster prototype $\mathbf{c}^* \in R(P)$ is individuated according to the following expression:

$$c(\mathbf{w}) = \mathbf{c}^*_{\mathbf{w}} = \arg\min_{\mathbf{c}_j \in R(P)} d(\mathbf{w}, \mathbf{c}_j). \tag{3}$$

The construction of the symbolic histogram is performed as follows. An array $\mathbf{I}_{w_i} = [\delta_1, \delta_2, ..., \delta_k]^T$ of indicator functions is constructed, where:

$$\delta_j = \begin{cases} 1 & \text{if } c(\mathbf{w}_i) = \mathbf{c}_j, i = 1, 2, ..., |D| \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Finally, the symbolic histogram for a document d is provided by:

$$\mathbf{h}^D = \sum_{i=1}^{|D|} \mathbf{I}_{w_i}. \tag{5}$$

Alternatively, instead of constructing a symbolic histogram as an array of counters, it is possible to represent the document $D \in \mathcal{H}$ as the average of the associated centroids $c(\mathbf{w}_i)$, for each word $w \in D$, that is:

$$\mathbf{h}^D_{avg} = \sum_{i=1}^{|D|} \frac{c(\mathbf{w}_i)}{|D|}. \tag{6}$$

At this point, each document in the corpus has an associated symbolic histogram, hence a vector of Integers or Real-valued numbers, depending on the specific rule adopted. In other words, documents are embedded in a bag of concept vector space.

Classification Layer

Once obtained the new representation, that is, the new vector space (through the symbolic histograms) or the new sequence of concepts, a learning layer can be designed depending on the problem at hand. In this work it is faced a classification problem comparing three different classification algorithms, namely SVM with Gaussian Kernel [73, 74], Bagged Tree RF [75, 76] and LSTM [77–79]. The first two learning algorithms are suited for working with Real-valued patterns, while LSTM is conceived for learning with a representation grounded by sequences of objects. Specifically, in the current approach LSTM is fed by the sequences of prototype vectors $c(\mathbf{w})_i$ obtained through Eq. 3 corresponding to the sequence of words w_i pertaining a given document D. These classification algorithms belong to three big and heterogeneous families of learning algorithms, namely kernel-based, where learning is conceived as a convex optimization problem (SVM), random tree-based (RF), and deep learning-based, specifically RNNs. Hence, this choice guarantees the diversity of the learning paradigms applied to the proposed method. It is worth noting that RF algorithms are based on the bootstrap technique (some samples will be used multiple times) and the observations that are out of the bootstrap sample are called out-of-bag (OOB). This technique allows estimating the importance of variables (features) through a suitable procedure described, for example, in [80].

Experiments

Data Sets

As concerns text data for experiments, the “Reuters-21578” data set and the “Abstracts” data set have been used. Reuters-21578 is a benchmark data set for document classification consisting in 8 classes. The collection of documents appeared on the Reuters news-wire in 1987. The documents were assembled and indexed with categories by personnel from Reuters Ltd. [81]. The adopted splitting is the “ModApte” split [82] on 7674 documents and 8 classes. The “Abstracts” data set is a collection of 575 abstracts of scientific papers belonging to 5 classes (‘Anatomy’, ‘Information theory’, ‘Smart Grid’, ‘String Theory’, ‘Semiconductors’), collected by authors. Some statistics on the experimented data sets are reported in Tables 1 and 2. The former provides some general information about the data set, while the latter reports some statistic per class, such as the mean and standard deviation of document length per class.

Specifically, in Table 1 the total number of documents (# docs), the dimension of the vocabulary before pre-processing ($|\mathcal{V}|$), the dimension of the vocabulary after the pre-processing ($|\mathcal{V}|_{pre}$), the number of classes (# class) and the average length of documents in terms of tokens (words) ($|\bar{D}|$) with standard deviation in brackets are reported. In Table 2 the class names (class), the average length of documents in term tokens

Table 1 Data set statistics (in brackets it is reported the standard deviation)

# docs.	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{V}\rvert_{pre}$	# class.	$\lvert\bar{D}\rvert$
Reuters-21578				
7674	23585	20768	8	67.649 (68.080)
Abstracts				
575	8722	6585	5	65.464 (31.687)

Table 2 Data set statistics per class (in brackets it is reported the standard deviation)

class	$\|\bar{D}\|$	# docs.
"Reuters-21578"		
'acq'	118.240(113.866)	2292
'crude'	187.297(155.367)	374
'earn'	65.642(76.740)	392
'grain'	183.765(182.954)	51
'interest'	118.672(154.255)	271
'money-fx'	161.867(173.211)	293
'ship'	149.924(85.788)	144
'trade'	234.460(179.296)	326
"Abstracts"		
'Anatomy'	96.400(43.593)	115
'Information theory'	146.148(52.508)	115
'Smart Grid'	148.843(51.823)	115
'String Theory'	84.809(44.047)	115
'Semiconductors'	126.565(63.512)	115

(words) ($|\bar{D}|$) with standard deviation in brackets and the number of documents per class (# docs) are reported. In Fig. 5 the statistics on document lengths per class and for each data set are reported, while in Fig. 6 the histograms of the length of documents for both data sets are depicted. This information will be useful for setting the sequence length parameter for experiment with the LSTM algorithm.

In Fig. 7 the class distributions for both the data sets are reported. We note that the "Reuters-21578" data set has a heavy skewed class distribution leading to a strong unbalanced data set, making challenging the classification task, while the "Abstracts" data set classes are equally distributed.

Experimental Settings

As concerns the performance measures of the classifiers, several metrics for the multi-class case are adopted.

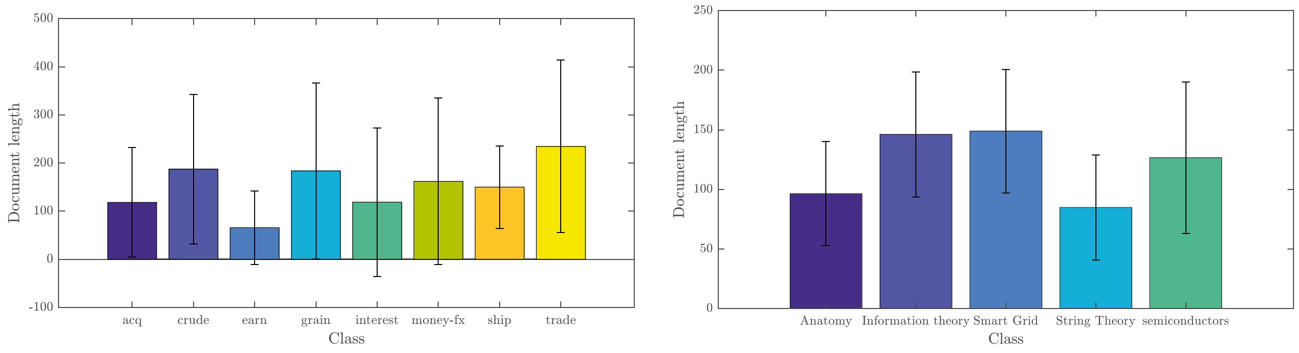
Specifically, considering the i-th class ($i = 1, 2, ...|C|$) of the available data sets it is possible to define:

- TP_i (true positive): number of patterns belonging to the i-th class and correctly classified by the system;
- FN_i (false negative): number of patterns belonging to the i-th class whose class is incorrectly assigned to the i_th class predicted by the system;
- FP_i (false positive): number of patterns not belonging to the i-th class whose class is incorrectly by the system.
- TN_i (true negative): number of patterns belonging to the i-th class and correctly classified by the system.

Starting from these metrics a set of derived indicators for each class can be computed, such as the $Accuracy_i$, $Precision_i$ and $Recall_i$ together with other global figures of merit, such as the *Informedness* and the Cohen's *Kappa*. Besides these metrics, the global classification performances can be assessed in two ways: (i) macro-averaging that is the average of the same measure calculated for each class, (ii) micro-averaging that is the sum of counts to obtain cumulative *TP*, *FN*, *TN*, *FP* and then calculating the performance measure. Macro-averaging treats all classes equally while micro-averaging favors classes characterized by a relative higher number of patterns [83].

The final metrics adopted in the current study are (the higher, the better):

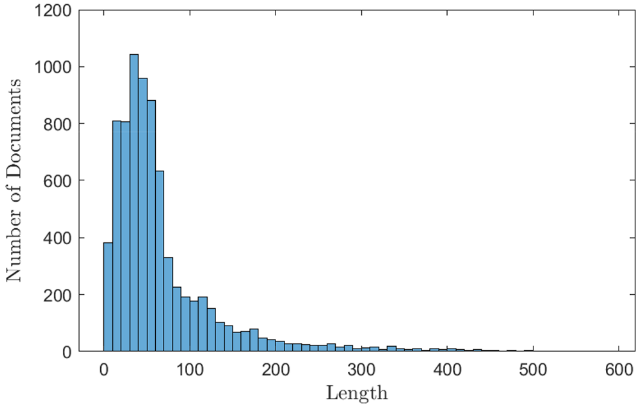
- the average Accuracy (Acc.) in [0,1], that is the average per-class effectiveness of a classifier;
- the Precision (P) in [0,1], that is the fraction of relevant instances among the retrieved instances by the classifier;
- the Recall (R) in [0,1], that is the fraction of the total amount of relevant instances that were actually retrieved by the classifier;
- the Informedness (Inf.) in [0,1] — known as J-index — is the maximum distance between the bisector diagonal line of the Receiver operating characteristic (ROC) [84] dia-



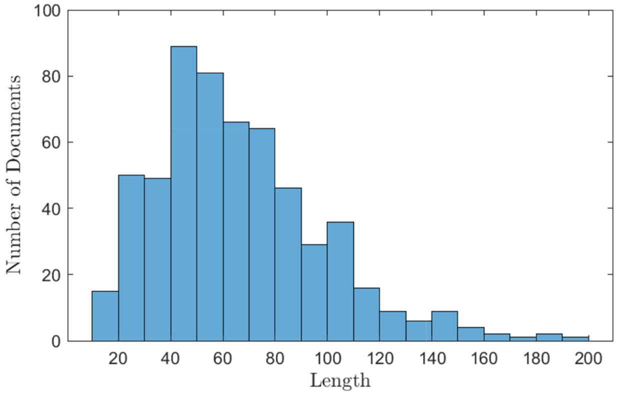
(a) "Reuters-21578" data set.

(b) "Abstracts" data set.

Fig. 5 Document length per class



(a) “Reuters-21578“ data set.



(b) “Abstracts” data set.

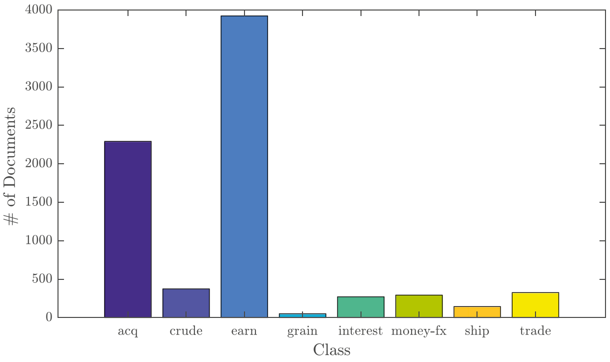
Fig. 6 Document length histogram

gram and the ROC curve estimated. It indicates the probability of an informed decision compared to a chance;

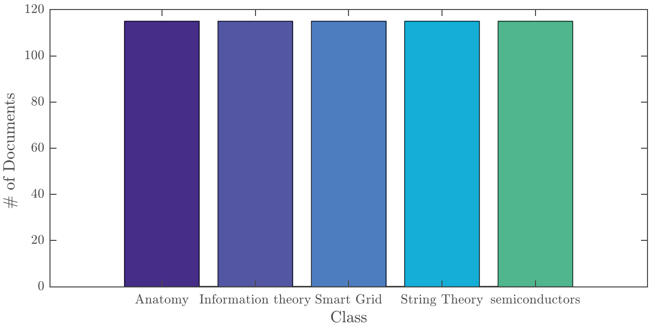
- the Cohen’s kappa (kappa) in [0,1] that, considering the classification task as a rating process, measures inter-rater reliability (sometimes called inter-observer agreement) [85];
- the macro F1 score (Fmacro) in [0,1] that is the unweighted mean of the F1 scores calculated per class, where $F1 = \frac{2TP}{2TP+FP+FN}$ [83];
- the micro F1 score (Fmicro) in [0,1] the same expression as Fmacro, but using the total number of TP, FP and FN, instead of computing these scores for each class [83].

The experimental settings are organized as follows.

In order to assess the proposed approach, two main sets of experiments are provided. The first set aims at comparing the three learning algorithms (namely, LSTM, C-SVM, RF) adopting both the *word2vec* and the LSA embedding, for both “Abstracts” and “Reuters-21578” data sets. The embedding is computed on the given corpus. In this case the cardinality of the alphabet $\mathcal{A}$, that is the number of clusters k or the number of concept regions, is left to vary in the integer range [2,1002] (see Fig. 8), while a snapshot of the performance, for $k = 502$, is provided in Table 3 for the “Reuters-21578” data set and, for $k = 202$, in Table 4 for the “Abstracts” data set. The specific choice of the granularity level k has been made simulating an arbitrary setting where we have no information about the variability of the performance as function of the granularity level. In other words, this setting simulates the case in which the performance cannot be computed for an increasing set of granularity levels due to, for example, computational and time constraints. The best granularity level, instead, is considered in the second set of experiments. In fact, the second set of experiments — reported in Tables 5 and 6 — allows evaluating and comparing the proposed



(a) “Reuters-21578” data set.



(b) “Abstracts” data set.

Fig. 7 Class distribution for the experimented data sets

Table 3 Classification performances for a given granularity level *k* for LSTM, C-SVM and RF (“Reuters-21578” data set)

Reuters-21578 data set			
	LSTM	C-SVM	RF
word2vec			
k	502	502	502
Acc.	0.9469(0.0000)	0.9495(0.0000)	0.9322(0.0000)
P	0.8294(0.0009)	0.8568(0.0000)	0.8856(0.0000)
R	0.8305(0.0002)	0.8339(0.0002)	0.6985(0.0002)
Inf.	0.8219(0.0002)	0.8251(0.0001)	0.6861(0.0002)
Kappa	0.7571(0.0000)	0.7647(0.0000)	0.6902(0.0000)
Fmicro	0.9469	0.9485	0.9322
Fmacro	0.8255	0.8448	0.7766
LSA			
k	502	502	502
Acc.	0.9092(0.0004)	0.9545(0.0000)	0.9325(0.0000)
P	0.7148(0.0004)	0.8921(0.0007)	0.9101(0.0001)
R	0.7274(0.0010)	87321(0.0001)	0.7020(0.0003)
Inf.	0.7125(0.0013)	0.8653(0.0001)	0.6895(0.0003)
Kappa	0.5851(0.0069)	0.9545(0.0003)	0.6916(0.0004)
Fmicro	0.9092	0.9545	0.9325
Fmacro	0.7188	0.8817	0.7930

methodology with a baseline approach. Specifically, for the mentioned learning algorithms the best level of granulation *k* in terms of performances is compared with a classical approach where the feature vectors representing documents are obtained either from the TF-IDF representation or from the LSA representation. In other words, the features for the classification task are the TF-IDF weighted words count or the weights related to the latent variables, respectively. In the end, taking advantage of the implicit features weighting offered by the RF algorithm, a task of knowledge discovery is performed. Hence, a threshold filtering is adopted in order to select and show the most important concepts within the concept region that, in turn, allowed reaching a good classification performance. Before discussing the main results, it is worth to deal with the text pre-processing steps and the parameter setting of the adopted learning algorithms.

Table 4 Classification performances for a given granularity level *k* for LSTM, C-SVM and RF (“Abstracts” data set)

Abstracts data set			
	LSTM	C-SVM	RF
word2vec			
k	202	202	202
Acc.	0.8527(0.0022)	0.9478(0.0000)	0.9275(0.0001)
P	0.8662(0.0016)	0.9492(0.0000)	0.9296(0.0001)
R	0.8547(0.0021)	0.9478(0.0000)	0.9275(0.0001)
Inf.	0.8179(0.0033)	0.9348(0.0000)	0.9094(0.0002)
Kappa	0.5397(0.0216)	0.8370(0.0000)	0.7736(0.0017)
Fmicro	0.8527	0.9478(0.0000)	0.9275
Fmacro	0.8552	0.9485(0.0000)	0.9283
LSA			
k	202	202	202
Acc.	0.8721(0.0034)	0.9302(0.0001)	0.9362(0.0000)
P	0.8686(0.0040)	0.9334(0.0000)	0.9400(0.0000)
R	8728(0.0034)	0.9305(0.0001)	0.9362(0.0000)
Inf.	0.8407(0.0053)	0.9130(0.0002)	0.9203(0.0000)
Kappa	0.6003(0.0033)	0.7815(0.0014)	0.8007(0.0002)
Fmicro	0.8721	0.9301	0.9362
Fmacro	0.8702	0.9313	0.9378

Regarding the pre-processing steps, words in documents are lowercased and stop words are eliminated using a stop words list.

As concerns the LSTM algorithm, it is preceded by a word embedding layer (through the *word2vec* algorithm) — namely a *concept embedding* layer — with a dimension of the word vectors equal to 100. The LSTM layer foresees 180 cells followed by a fully connected layer and a softmax layer. The classification layer computes the cross entropy loss for multi-class classification problems with mutually exclusive classes. The maximum number of epochs is set to 50, while the initial learning rate is set to 0.005. In the case of SVM, a C-SVM with multiple kernels is used. A set of hyper-parameters are optimized with the Bayesian optimization technique and 5-fold cross validation. Specifically, the hyper-parameters optimized are the multi-class coding (One-versus-All and One-versus-One), the Box Constraint, the scale of the kernel, the type of kernel function (*Gaussian*, *linear*, *polynomial*), the polynomial order and the binary variable indicating whether standardize data or not. For the ensemble learning, an ensemble of boosted classification trees is experimented, hence with trees as weak learners. Even in this case, a Bayesian hyper-parameters optimization has been chosen and 5-fold cross validation is performed. In particular the optimization of the training algorithm (*Bag*, *Subspace*, *AdaBoostM1*, *AdaBoostM2*, *GentleBoost*, *LogitBoost*, *LPBoost*, *RobustBoost*, *RUSBoost*, *TotalBoost*) and the number of learning cycles [80, 86–88]. The OOB performance is measured for establishing the predictor importance.

Where not specified, for robustness purposes, the optimizations of hyper-parameters or the simple learning routines (for example, in LSTM) are repeated three times and performance results are averaged. The data set splitting for the training set $\mathcal{S}_{tr}$ and test set $\mathcal{S}_{ts}$ is 80%, 20%, respectively, both for C-SVM and RF. In the case of LSTM the data set is split in training set $\mathcal{S}_{tr}$, validation set $\mathcal{S}_{vs}$, test set $\mathcal{S}_{ts}$ with the following percents: 50%, 25%, 25%, respectively.

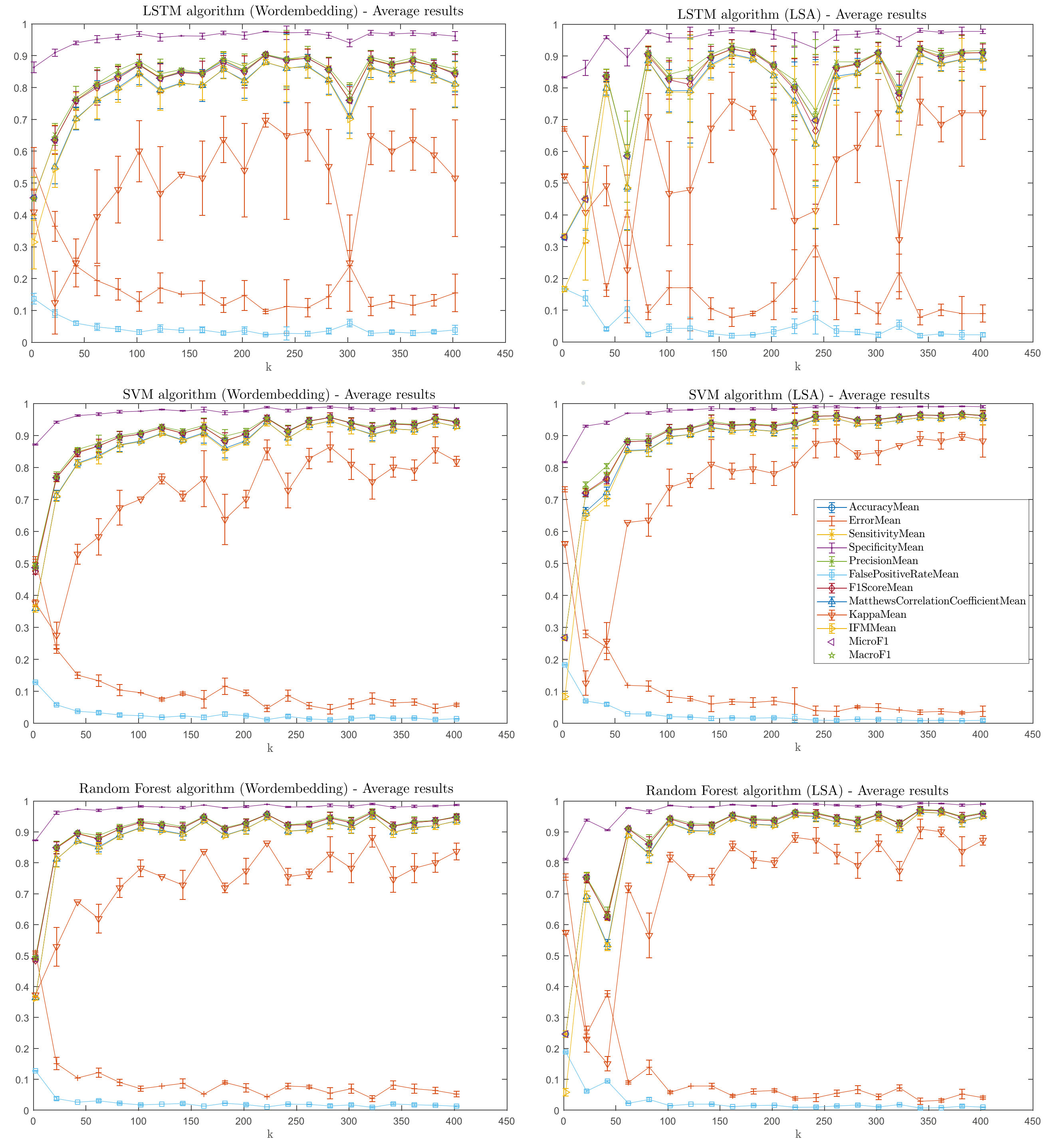


Fig. 8 Classification performance varying the number of concept regions from 2 to 402 ("Abstracts" data set) for the LSTM, SVM, RF algorithms, for both *word2vec* and LSA techniques

Results and Granularity Assessment

Fixing the concept granularity value to $k = 202$ — see Table 4 — where the three classification algorithms are compared both with the *word2vec* embedding and the LSA embedding in building the concept space over the "Abstracts dataset" (that is balanced), best classification performances in term of Accuracy (0.94) are obtained with C-SVM with *word2vec*. The second best performances are obtained with RF and C-SVM (Accuracy 0.93) with the LSA embedding. LSTM reaches lower performances on both embeddings with an Accuracy of 0.85 for *word2vec* and 0.87 for the

Table 5 Performances comparison over the "Reuters-21578" data set between LSTM, C-SVM and RF for both *word2vec* and LSA embeddings. Results of the best granulation level *k* are compared with the plain solution given by a sequence formed by the word vectors related to words in the given text for LSTM, and the TF-IDF weighting scheme for the other classifiers

Reuters-21578 data set

	LSTM		C-SVM		RF	
	word2vec	***sequence***	***word2vec***	Term Frequency	***word2vec***	Term Frequency
	best	plain	best	plain	best	plain
k	622	x	662	x	542	x
Acc.	0.9551(0.0000)	0.9094(0.0002)	0.9657(0.0000)	0.9667(0.0000)	0.9505(0.0000)	0.9332(0.0000)
P	0.8396(0.0001)	0.7449(0.0049)	0.9198(0.0001)	0.9386(0.0000)	0.9207(0.0000)	0.9130(0.0004)
R	0.8350(0.0005)	0.6669(0.0022)	0.8697(0.0005)	0.8942(0.0000)	0.7684(0.0003)	0.6865(0.0006)
Inf.	0.8277(0.0005)	0.6507(0.0025)	0.8639(0.0005)	0.8884(0.0000)	0.7598(0.0003)	0.6764(0.0006)
Kappa	0.7946(0.0003)	0.5628(0.0049)	0.8434(0.0000)	0.8475(0.0000)	0.7738(0.0001)	0.6944(0.0000)
Fmicro	0.9551(0.0000)	0.9094(0.0000)	0.9657(0.0001)	0.9667(0.0000)	0.9505(0.0000)	0.9332(0.0000)
Fmacro	0.8356(0.0000)	0.6961(0.0000)	0.8939(0.0001)	0.9159(0.0000)	0.8376(0.0000)	0.7895(0.0000)
	LSA		***LSA***		***LSA***	
k	782	x	722	x	962	x
Acc.	0.9260(0.0000)	/	0.9621(0.0000)	0.9660(0.0000)	0.9356(0.0001)	0.9595(0.0000)
P	0.7812(0.0004)	/	0.9250(0.0002)	0.9249(0.0000)	0.7800(0.0002)	0.9150(0.0004)
R	0.7680(0.0001)	/	0.8952(0.0000)	0.9056(0.0000)	0.7061(0.0023)	0.8279(0.0001)
Inf.	0.7556(0.0001)	/	0.8886(0.0000)	0.9000(0.0000)	0.6945(0.0025)	0.8209(0.0001)
Kappa	0.6617(0.0002)	/	0.8267(0.0001)	0.8449(0.0000)	0.7055(0.0018)	0.8148(0.0003)
Fmicro	0.9260(0.0000)	/	0.9621(0.0000)	0.9670(0.0000)	0.9356(0.0000)	0.9595(0.0000)
Fmacro	0.7712(0.0000)	/	0.9095(0.0000)	0.9148(0.0000)	0.7979(0.0000)	0.8668(0.0000)

LSA embeddings. Interestingly, if we compute the figures of merits as the granularity of the concept space increases (varying *k*), by inspection of Fig. 8, it is found that with a very low number of concepts all classifiers achieve higher performances that, in turn, stabilize till the end of the experimented range ($k = 1002$). A similar behavior can be found analyzing the "Reuters-21578" data set. Here the granularity level is fixed to $k = 502$. Also in this case the performance in terms of classification capability increases quickly rising *k* (graphs not shown for the sake of brevity). However, in this case, examining the results reported in Table 3, the higher Accuracy value is attained by C-SVM (0.95 with LSA embedding), but even the LSTM obtains a good Accuracy (0.94 with *word2vec* embedding).

If we consider the classification task as a rating process, Cohen's kappa coefficient is low for LSTM and RF, for both embeddings, but reaches the highest value for C-SVM with *word2vec* embedding. In terms of Fmicro and Fmacro C-SVM obtains its best results with the *word2vec* embedding. The best informed decision, taking into account the unbalance of the "Reuters-21578" data set is achieved by C-SVM with Informedness of 0.86 (LSA embedding). In general, as an expected behavior, we have a moderate variability of the classifiers' performances for both embeddings and data sets. The low Accuracy attained by LSTM for the "Abstracts" data set is likely to be addressed to the low granularity level and the short dimension of the data set, in terms of the number of documents and documents length. However, if we look at results for the best granularity level reported in Table 6 ("Abstracts" dataset), where the three classifiers are compared for both embedding types and with the TF-IDF features, LSTM obtains better performances with Accuracy 0.90 (*word2vec* embedding for best $k = 222$) and 0.92 (LSA embedding for best $k = 162$), outperforming the plain case (Accuracy 0.70), where sequences are directly generated without conceptualizing the corpus. In this particular setting, C-SVM with LSA embedding, for the best $k = 382$, outperforms both LSTM and RF. For C-SVM the LSA embedding adopted for constructing the conceptual space is found better than the TF-IDF case (Accuracy 0.98 and 0.96, respectively). It is worth to note that for both C-SVM and RF the results for the plain case (TF-IDF feature space) are good (RF attains an Accuracy of 0.97 for $k = 342$ and 0.95 with TF-IDF) in spite of the high dimensionality of the features space. This confirms the capability of both classifiers to work well in high-dimensional spaces. Both algorithms achieve high Accuracy and high Informedness for a similar granularity level above $k = 300$ in the LSA embedding case, while LSTM obtain even good performances (Accuracy 0.92) with the same embedding with a very low

Table 6 Performances comparison over the “Abstracts” data set between LSTM, C-SVM and RF for both *word2vec* and LSA embeddings. Results of the best granulation level *k* are compared with the plain solution given by a sequence formed by the word vectors related to words in the given text for LSTM, and the TF-IDF weighting scheme for the other classifiers

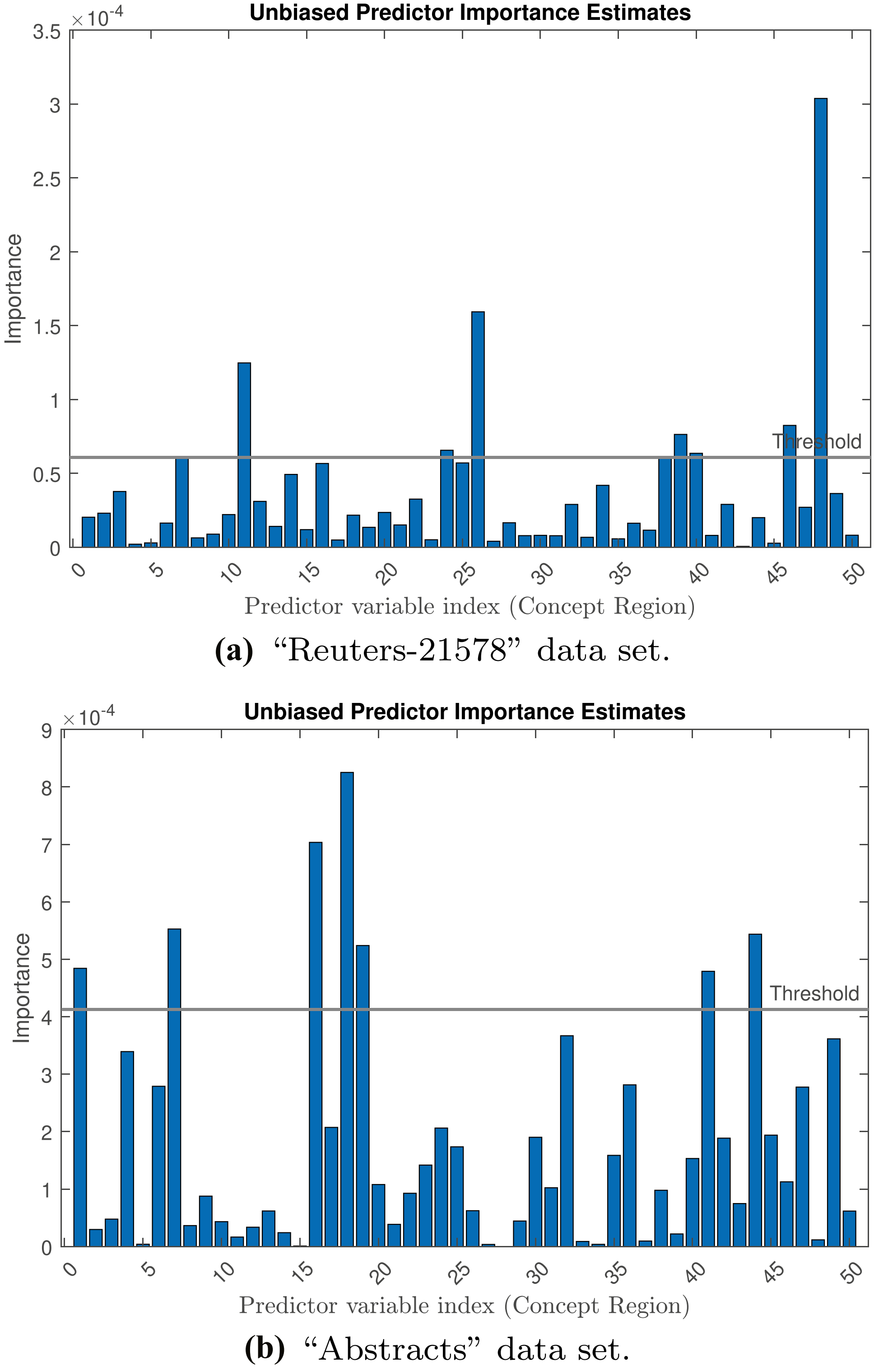
Abstracts data set						
	LSTM		C-SVM		RF	
	word2vec	***sequence***	***word2vec***	Term Frequency	***word2vec***	Term Frequency
	best	plain	best	plain	best	plain
k	222	x	282	x	322	x
Acc.	0.9031(0.0000)	0.7093(0.0151)	0.9565(0.0002)	0.9681(0.0000)	0.9623(0.0001)	0.9536(0.0003)
P	0.9042(0.0000)	0.7203(0.0156)	0.9566(0.0002)	0.9710(0.0000)	0.9658(0.0001)	0.9557(0.0003)
R	0.9037(0.0000)	0.7094(0.0152)	0.9565(0.0002)	0.9681(0.0000)	0.9623(0.0001)	0.9536(0.0003)
Inf.	0.8795(0.0001)	0.6367(0.0238)	0.9457(0.0004)	0.9601(0.0000)	0.9529(0.0002)	0.9420(0.0005)
Kappa	0.6972(0.0004)	0.2482(0.0618)	0.8641(0.0022)	0.9004(0.0002)	0.8822(0.0010)	0.8551(0.0032)
Fmicro	0.9031(0.0000)	0.7093(0.0000)	0.9565(0.0000)	0.9681(0.0000)	0.9623(0.0000)	0.9536(0.0000)
Fmacro	0.9033(0.0000)	0.7106(0.0000)	0.9565(0.0000)	0.9664(0.0000)	0.9640(0.0000)	0.9538(0.0000)
	LSA		***LSA***		***LSA***	
k	162	x	382	x	342	x
Acc.	0.9225(0.0009)	/	0.9826(0.0001)	0.9681(0.0000)	0.9710(0.0001)	0.9594(0.0000)
P	0.9316(0.0004)	/	0.9833(0.0001)	0.9715(0.0000)	0.9730(0.0001)	0.9617(0.0000)
R	0.9222(0.0009)	/	0.9826(0.0001)	0.9681(0.0000)	0.9710(0.0001)	0.9594(0.0000)
Inf.	0.9028(0.0014)	/	0.9783(0.0001)	0.9601(0.0000)	0.9638(0.0002)	0.9493(0.0000)
Kappa	0.7578(0.0084)	/	0.9457(0.0007)	0.9004(0.0002)	0.9094(0.0010)	0.8732(0.0002)
Fmicro	0.9225(0.0000)	/	0.9826(0.0000)	0.9681(0.0000)	0.9710(0.0000)	0.9594(0.0000)
Fmacro	0.9230(0.0000)	/	0.9828(0.0000)	0.9695(0.0000)	0.9719(0.0000)	0.9605(0.0000)

granularity ($k = 162$). Furthermore, considering the “Reuters-21578” data set on the same experimental setting, we have C-SVM that outperforms the other classifiers in terms of Informedness and Fscores in TF-IDF case, with a not so great separation from the LSA embedding with a granularity level of $k = 722$. For this data set even LSTM achieves good results in terms of Accuracy and Informedness, specially for the *word2vec* embedding (Accuracy 0.95, Informedness 0.82, Fmicro 0.95, Fmacro 0.83). Similar performances are obtained with RF that only in the *word2vec* embedding case outperforms the TF-IDF setting. In fact, for the LSA embedding, despite the high granularity level ($k = 962$) the TF-IDF setting attains a high Accuracy (0.95 v.s. 0.93) and a higher Informedness (082 v.s. 0.89). A similar behavior can be found for the Fmicro and Fmacro and for the kappa coefficient. Comparing the two data sets, that are very different in their structure and contents, the granularity level needed for attaining good results is found proportional to the complexity of the data set itself. In fact, the “Abstracts” data set consists of a set of short documents and each class is well separated in term of contents, at least at a semantic point of view. The “Reuters-21578” data set possesses more classes that are strongly unevenly distributed (see Fig. 7). For both data set, the conceptualization does not degrade the results, instead we obtain good performances with a lower granularity level. This means a low complexity of the feature space, that instead to be equal to the cardinality of the vocabulary (in the plain TF-IDF case) it matches the number of concepts adopted for representing the documents.

In the current experiments, there is no supremacy among the *word2vec* and the LSA in building the concept space. In fact, even if for both embeddings a smaller concept space attains good performances, there are classifiers that perform well for LSA and classifiers that do the best for the *word2vec* embedding. It is well known that both the embeddings possess suitable semantic characteristics, and the general performances depend on the entire processing chain and the particular hyper-parameter settings.

In general, the granulation of the word embedding space can be seen as a dimensionality reduction paradigm at the cost of inserting a new block in the downstream processing of texts before the classification task. However, this conceptualization block can be constructed once and for all even adopting richer corpora (e.g., Wikipedia), conversely to the one employed for the specific classification task. It is important to take care of the granulation parameter, significant for the classifier performances due to their attitudes in working with high or low dimensionality.

Fig. 9 Concept importance for the “Reuters-21578” (**a**) and “Abstracts” (**b**) data sets. The threshold value filters low importance concepts. The value is set as the half of the max value computed on concept importance



(a) “Reuters-21578” data set.

(b) “Abstracts” data set.

Towards the Explainable AI paradigm

The choice of the three particular classification algorithms depends on their specific characteristics. In fact, if from one hand SVM is known to be performing even with high dimensional feature spaces embedded in $\mathbb{R}^n$, LSTM is suitable with sequences and needs a further dense layer to be appropriate for classification tasks. RF, for its part, is suited for classification tasks where it is important also estimating the importance of features. In fact, RF offers the possibility to obtain a set of weights, as many as the number of features, that, in turn, in this study are a kind of superordinate word, we call concepts. Fixing an arbitrary threshold over these weights leads to estimate the strongest concepts related to the specific classification task. From a different point of view, this procedure can be seen as a concept filtering task, where only the strongest ones survive. This methodology can help to infer knowledge on the corpus, eliciting the Explainable AI paradigm.

In Fig. 9(a), (b), as a bar chart, the weights related to the concepts and the thresholds (fixed to the 20% of the largest weight value), for the “Reuters-21578” and “Abstracts” data set, respectively, are depicted. For these specific experiments, for brevity purposes, the granularity of the concept space is fixed to $k = 50$ and it is constructed through the *word2vec* neural embedding. It is worth to note that the CSS is obtained through the k-means clustering algorithm, thus the prototype, being the average word vector for a given concept region, is a surrogate word vector. So forth, in order to find the existing word related to the prototype, the ℓ_2 norm distance is computed selecting the nearest word vector and the corresponding token. In Tables 7 and 8 the survived concepts together with the nearest five words obtained computing a ℓ_2 norm between the prototype and the respective word vectors are reported, normalizing for the highest similarity value that hold for the closest existing word vector to the prototype. In case of clusters with cardinality lower than five, all words within the cluster are shown. For the “Abstracts” data set, the best closest prototypes, for a given threshold value, are *holographic*, *concurrently*, *explicitly*, *explanation*, *succesfully*, *achieve*, *intrusiveness*, *leastsquares*, *pregnancy*, *furthermore*, *nonabelian*, *effectiveness*, *percutaneous*, *approximation*, *nanolasers*, *robustness*, *insulator*, *chalcogenide*, *rolling*, *infrared*. From Table 7, selecting, for example, a populated region, such as the fourth and considering the closest word (concurrently) to its prototype, we have *electrification*, *resistant*, *diameter*, *platform*, *industrial*. In general, we can find high semantic words that elicit roughly which term the algorithm estimates as important for the classification task; in fact, besides words with high semantic content, we can find verbs (for example, *achieve*) or other lexemes that are mostly used in papers’ abstracts. The same rationale can be found behind the results for the “Reuters-21578” data set illustrated in Table 7. Here, we can find less singleton or low-populated clusters, due to the dimension of the corpus. Even in this case it can be found a set of prototype words that span uniformly the conceptual space. Nevertheless, the richness of the semantic contents of prototypes and the underlying word cloud is attributable to the dimension and the heterogeneity of the corpus, since it is likely to lead to better representations for words, that in turn, leads to a performing CSS.

Table 7 Most important concepts obtained by filtering the Concept Regions through the feature importance estimation provided by the RF algorithm (“Reuters-21578” data set). There are reported the first five words for each region that exceed a threshold value — see Fig. 9(a)

“Reuters-21578” data set		
Word	**Norm. similarity**	**Concept Reg. Index**
“_overdraft*_”	1	2
“_gilt_”	0.97116	2
“_kwacha_”	0.95985	2
“_ours_”	0.95916	2
“_afterwards_”	0.95914	2
“_peseta_”	0.95878	2
“_tvx*_”	1	17
“_matthey_”	0.9753	17
“_behalf_”	0.97311	17
“_rmj_”	0.96955	17
“_cvn_”	0.96909	17
“_labelling_”	0.96849	17
“_mdc*_”	1	18
“_cbs_”	0.97853	18
“_mpb_”	0.97837	18
“_lpl_”	0.9771	18
“_uac_”	0.97703	18
“_magna_”	0.97675	18
“_fintech*_”	1	20
“_interactive_”	0.9439	20
“_intercompany_”	0.94195	20
“_integral_”	0.93706	20
“_intense_”	0.93507	20
“_intend_”	0.93033	20
“_veto*_”	1	27
“_sense_”	0.94648	27
“_accuse_”	0.94509	27
“_herrington_”	0.94407	27
“_gephardt_”	0.94021	27
“_imported_”	0.93945	27
“_ln*_”	1	34
“_nov_”	0.92031	34
“_ln_”	0.90159	34
“_ust_”	0.89955	34
“_eight_”	0.8988	34
“_sept_”	0.89351	34
“_libya*_”	1	38
“_indonesian_”	0.93859	38
“_libya_”	0.93804	38
“_iea_”	0.93717	38
“_quotas_”	0.93639	38
“_egypt_”	0.93517	38

Table 8 Most important concepts obtained by filtering the Concept Regions through the feature importance estimation provided by the RF algorithm ("Abstracts" data set). There are reported the first five words for each region that exceed a threshold value — see Fig. 9(b)

"Abstracts" data set

Word	Norm. similarity	Concept Reg. Index
"*holographic**"	1	1
"*holographic*"	0.9737	1
"*algorithm*"	0.96451	1
"*forward*"	0.96052	1
"*emphasis*"	0.96039	1
"*falling*"	0.95878	1
"*concurrently**"	1	4
"*electrification*"	0.9233	4
"*resistant*"	0.922	4
"*diameter*"	0.92197	4
"*platform*"	0.92186	4
"*industrial*"	0.92002	4
"*explicitly**"	1	6
"*respectively*"	0.90374	6
"*instantaneous*"	0.9037	6
"*role*"	0.90366	6
"*equilibrium*"	0.90349	6
"*vortex*"	0.90345	6
"*explanation**"	1	10
"*investigation*"	0.89886	10
"*production*"	0.89886	10
"*congestion*"	0.89669	10
"*information*"	0.89624	10
"*decision*"	0.89601	10
"*successfully**"	1	18
"*parathyroid*"	0.86238	18
"*careful*"	0.86218	18
"*marginal*"	0.86211	18
"*finding*"	0.86207	18
"*achieve**"	1	19
"*embed*"	0.92432	19
"*access*"	0.92391	19
"*outage*"	0.92324	19
"*lighting*"	0.92317	19
"*sufficiently**"	1	20
"*purely*"	0.90645	20
"*perform*"	0.90638	20
"*community*"	0.90625	20
"*alloy*"	0.90597	20
"*intrusiveness**"	1	22
"*undergraduate*"	0.92668	22
"*disturbance*"	0.91918	22
"*correlate*"	0.91901	22
"*continuity*"	0.9183	22
"*leastsquares**"	1	25
"*behavior*"	0.89503	25
"*importantly**"	1	26
"*transmission*"	0.95505	26
"*pregnancy**"	1	30
"*dispatch*"	0.89991	30
"*furthermore**"	1	31
"*chain*"	0.9178	31
"*nonabelian**"	1	36
"*matter*"	0.90482	36
"*effectiveness**"	1	37
"*limit*"	0.91467	37
"*percutaneous**"	1	38
"*randomness*"	0.92696	38
"*approximation**"	1	41
"*circulation*"	0.91257	41
"*nanolasers**"	1	42
"*contrast*"	0.91577	42
"*robustness**"	1	43
"*evidencebased*"	0.97102	43
"*insulator**"	1	44
"*series*"	0.91583	44
"*chalcogenide**"	1	46
"*hash*"	0.90352	46
"*rolling**"	1	48
"*explores*"	0.92888	48
"*infrared**"	1	50
"*higher*"	0.84745	50

Conclusions

The current study is an effort in providing a clear relationship between findings in the Conceptual Spaces theory and the problem of text representation in Pattern Recognition, specifically in NLP tasks involved in text mining. Text mining, as a particular application of Machine Learning techniques related to textual data, benefits from better representations of text as hierarchically organized set of features, where Granular Computing techniques can offer a wide range of tools for designing performing classification algorithms. Within this framework, where Granular Computing is bound to the Conceptual Spaces theory and Machine Learning, it is offered a comparison of some classification algorithms working on features constructed over the prototypes of a conceptual space obtained, in turn, over a suitable neural word embedding. Results show primarily that the conceptual layer placed in the middle between the associative layer and the symbolic layer (the symbolic histograms layer in this study) can be used for working with more abstract entities compared to words. These entities are a byproduct of the granulation of the conceptual space, that can be obtained with any algorithm in charge of embedding the text in a vector space. The three algorithms compared, two of them (SVM and RF) able to receive in input *n*-tuple of Real-valued numbers and the other (LSTM) working with input sequences, perform well for a large range of granulation levels of the conceptual space. Interestingly, depending on the nature of the textual data set, a low granulation level allows achieving good classification results, that at the stage of the current study depends only weakly from the specific algorithm. Moreover, the conceptual level together with the symbolic histograms technique can aid in Knowledge Discovery tasks, providing a framework for transforming black-box classifiers in gray ones, mining the strongest concepts that lead to a particular classification task, making a tiny step towards how a machine can represent meaning. Future works foresee the training of the conceptual space on exogenous corpora and, furthermore, the extension to *n*-gram prototypes where a suitable dissimilarity measure between sequences of vectors needs to be carefully designed in order to build up the symbolic histograms for the concepts embedding.

Funding Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of Interest The authors declare no competing interests.

References

1. Jain AK, Duin RPW. Introduction to pattern recognition. The Oxford Companion to the Mind, Second Edition; 2004.
2. Duin RPW, Pekalska E. Open issues in pattern recognition. In Computer Recognition Systems, pages 27–42. Springer; 2005.
3. Gärdenfors P. Conceptual spaces: The geometry of thought. MIT press; 2004.
4. Duin RPW, Roli F, de Ridder D. A note on core research issues for statistical pattern recognition. Pattern Recogn Lett. 2002;23(4):493–9.
5. Shea N. Representation in cognitive science. Oxford University Press; 2018.
6. Serra R, Zanarini G. Complex systems and cognitive processes. Springer Science & Business Media; 2013.
7. Cameron L, Larsen-Freeman D. Complex systems and applied linguistics. Int J Appl Linguist. 2007;17(2):226–39.
8. Korzybski A. Science and sanity (lancaster); 1933.
9. Yao JT, Vasilakos AV, Pedrycz W. Granular computing: Perspectives and challenges. IEEE Trans Cybern. 2013;43(6):1977–89.
10. Martino A, DeSantis E, Rizzi A. An ecology-based index for text embedding and classification. In 2020 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE; 2020.
11. Apolloni B, Bassis S, Malchiodi D, Pedrycz W. The puzzle of granular computing, volume 138. Springer; 2008.
12. Rizzi A, DelVescovo G. Automatic image classification by a granular computing approach. In 2006 16th IEEE signal processing society workshop on machine learning for signal processing, pages 33–38. IEEE; 2006.
13. Bargiela A, Pedrycz W. Toward a theory of granular computing for human-centered information processing. IEEE Trans Fuzzy Syst. 2008;16(2):320–30.
14. Zadeh LA. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. 1997;90(2):111–27.
15. Zadeh LA. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Computing-A Fusion of Foundations, Methodologies and Applications. 1998;2(1):23–5.
16. Altmann EG, Cristadoro G, Esposti MD. On the origin of long-range correlations in texts. Proc Natl Acad Sci. 2012;109(29):11582–7.
17. Lorenzo Livi, Guido DelVescovo, Antonello Rizzi, and Fabio Massimo FrattaleMascioli. Building pattern recognition applications with the spare library. arXiv preprint arXiv:1410.5263; 2014.
18. Song D, Bruza PD. Towards context sensitive information inference. J Am Soc Inform Sci Technol. 2003;54(4):321–34.
19. DelVescovo G, Rizzi A. Automatic classification of graphs by symbolic histograms. In 2007 IEEE International Conference on Granular Computing (GRC 2007), pages 410–410. IEEE; 2007.
20. Capillo A, DeSantis E, Mascioli FMF, Rizzi A. Mining m-grams by a granular computing approach for text classification; 2020.
21. Fabre C, Lenci A. Distributional semantics today; 2015.
22. Malcolm Norman. Wittgenstein's philosophical investigations. Philos Rev. 1954;63(4):530–59.
23. Harris ZS. Distributional structure. Word. 1954;10(2–3):146–62.
24. Firth JR. A synopsis of linguistic theory, 1930-1955. Studies in linguistic analysis; 1957.
25. Zhang Y, Jin R, Zhou ZH. Understanding bag-of-words model: a statistical framework. Int J Mach Learn Cybern. 2010;1(1–4):43–52.

26. McTear M, Callejas Z, Griol D. The conversational interface: Talking to smart devices. Springer; 2016.
27. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Commun ACM. 1975;18(11):613–20.
28. Lund K, Burgess C. Producing high-dimensional semantic spaces from lexical co-occurrence. Behav Res Methods Instrum Comput. 1996;28(2):203–8.
29. Levy O, Goldberg Y. Linguistic regularities in sparse and explicit word representations. In Proceedings of the Eighteenth Conference On Computational Natural Language Learning, pages 171–180; 2014.
30. Landauer TK, Dumais ST. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychol Rev. 1997;104(2):211.
31. Landauer TK, Foltz PW, Laham D. An introduction to latent semantic analysis. Discourse Process. 1998;25(2–3):259–84.
32. Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference On Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543; 2014.
33. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781; 2013.
34. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119; 2013.
35. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805; 2018.
36. Gabrilovich E, Markovitch S, et al. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In IJcAI. 2007;7:1606–11.
37. Sahlgren M, Cöster R. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In Proceedings of the 20th international conference on Computational Linguistics, page 487. Association for Computational Linguistics; 2004.
38. Miller GA. WordNet: An electronic lexical database. MIT press; 1998.
39. Rosch EH. Natural categories. Cogn Psychol. 1973;4(3):328–50.
40. Rosch E. Cognitive representations of semantic categories. J Exp Psychol Gen. 1975;104(3):192.
41. Wichert A. Sub-symbols and icons. Cogn Comput. 2009;1(4):342–7.
42. Wiggins GA. The mind's chorus: Creativity before consciousness. Cogn Comput. 2012;4(3):306–19.
43. Doran D, Schulz S, Besold TR. What does explainable ai really mean? a new conceptualization of perspectives. arXiv preprint arXiv:1710.00794; 2017.
44. Mueller ST, Hoffman RR, Clancey W, Emrey A, Klein G. 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.
45. DelVescovo G, Rizzi A. Online handwriting recognition by the symbolic histograms approach. In 2007 IEEE International Conference on Granular Computing (GRC 2007), pages 686. IEEE; 2007.
46. Bianchi FM, Livi L, Rizzi A, Sadeghian A. A granular computing approach to the design of optimized graph classification systems. Soft Comput. 2014;18(2):393–412.
47. Livi L, Del Vescovo G, Rizzi A. Graph recognition by seriation and frequent substructures mining. In ICPRAM. 2012;1:186–91.
48. Livi L, DelVescovo G, Rizzi A. Combining graph seriation and substructures mining for graph recognition. In Pattern Recognition-Applications and Methods, pages 79–91. Springer; 2013.
49. Rizzi A, DelVescovo G, Livi L, Mascioli FMF. A new granular computing approach for sequences representation and classification. In The 2012 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE; 2012.
50. Jing L, Lau RYK. Granular computing for text mining: New research challenges and opportunities. In International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing, pages 478–485. Springer; 2009.
51. Pedrycz W, Skowron A, Kreinovich V. Handbook of granular computing. John Wiley & Sons; 2008.
52. Zhang X, Yin Y, Haiyan Y. An application on text classification based on granular computing. Communications of the IIMA. 2007;7(2):1.
53. Possemato F, Rizzi A. Automatic text categorization by a granular computing approach: facing unbalanced data sets. In The 2013 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE; 2013.
54. Carrillo M, López-López A. Concept based representations as complement of bag of words in information retrieval. In IFIP International Conference on Artificial Intelligence Applications and Innovations, pages 154–161. Springer; 2010.
55. Mouriño-García MA, Pérez-Rodríguez R, Anido-Rifón LE. A bag of concepts approach for biomedical document classification using wikipedia knowledge. Methods Inf Med. 2017;56(05):370–6.
56. Shalaby W, Zadrozny W. Mined semantic analysis: A new concept space model for semantic representation of textual data. In 2017 IEEE International Conference on Big Data (Big Data), pages 2122–2131. IEEE; 2017.
57. Kim HK, Kim H, Cho S. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. Neurocomputing. 2017;266:336–52.
58. Matsumoto K, Yoshida M, Xiao Q, Luo X, Kita K. Emotion recognition for sentences with unknown expressions based on semantic similarity by using bag of concepts. In 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pages 1394–1399. IEEE; 2015.
59. Shalaby W, Zadrozny W. Learning concept embeddings for dataless classification via efficient bag-of-concepts densification. Knowl Inf Syst. 2019;1–24.
60. Zenker F, Gärdenfors P. Applications of conceptual spaces: the case for geometric knowledge representation, volume 359. Springer; 2015.
61. Ishwarya MS, Kumar CA. Quantum aspects of high dimensional conceptual space: a model for achieving consciousness. Cogn Comput., 2020;1–14.
62. Gärdenfors P. The geometry of meaning: Semantics based on conceptual spaces. MIT Press; 2014.
63. Palmer SE. Fundamental aspects of cognitive representation. Cognition and Categorization 1978.
64. Divjak D, Arppe A. Extracting prototypes from exemplars what can corpus data tell us about concept representation? 2013.
65. Langacker RW. Foundations of cognitive grammar: Theoretical prerequisites, volume1. Stanford university press; 1987.
66. Pčkalska E, Duin RPW. The dissimilarity representation for pattern recognition: Foundations and applications; 2005.
67. Okabe A, Boots B, Sugihara K, Chiu SN. Spatial tessellations: concepts and applications of Voronoi diagrams, volume 501. John Wiley & Sons; 2009.
68. Qiang D, Faber V, Gunzburger M. Centroidal voronoi tessellations: Applications and algorithms. SIAM Rev. 1999;41(4):637–76.
69. De S E, Martino A, Rizzi A. On component-wise dissimilarity measures and metric properties in pattern recognition. PeerJ Computer Science. 2022;8.
70. Lloyd SP. Least squares quantization in pcm. IEEE Trans Inf Theory. 1982;28:129–37.
71. Forgy EW. Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. Biometrics. 1965;21:768–9.
72. Del Vescovo G, Livi L, Mascioli FMF, Rizzi A. On the Problem of Modeling Structured Data with the MinSOD Representative. International Journal of Computer Theory and Engineering. 2014;6(1):9–14.

73. Vapnik VN, Vapnik V. Statistical learning theory, vol. 1. New York: Wiley; 1998.
74. Schölkopf B, Burges CJC. Advances in kernel methods: support vector learning. MIT press; 1999.
75. Ho TK. Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition, volume1, pages 278–282. IEEE; 1995.
76. Kleinberg EM. On the algorithmic implementation of stochastic discrimination. IEEE Trans Pattern Anal Mach Intell. 2000;22(5):473–90.
77. Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume2, pages 850–855 vol.2; Sep. 1999.
78. Huang K, Hussain A, Wang QF, Zhang R. Deep Learning: Fundamentals, Theory and Applications, volume2. Springer; 2019.
79. Kamath U, Liu J, Whitaker J. Deep learning for NLP and speech recognition; 2019.
80. Verikas A, Gelzinis A, Bacauskiene M. Mining data with random forests: A survey and results of new tests. Pattern Recogn. 2011;44(2):330–49.
81. Dobbins S, Topliss M, Steve Weinstein S. UCI Machine Learning Repository: Reuters-21578 Text Categorization Collection Data Set; 1987.
82. Dobbins S, Topliss M, Steve Weinstein S, -.
83. Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009;45(4):427–37.
84. Fawcett T. An introduction to roc analysis. Pattern Recognition Letters 2006;27(8):861–874. ROC Analysis in Pattern Recognition.
85. McHugh ML. Interrater reliability: The kappa statistic. Biochemia Medica: Biochemia Medica. 2012;22(3):276–82.
86. Seiffert C, Khoshgoftaar TM, VanHulse J, Napolitano A. Rusboost: Improving classification performance when training data is skewed. In 2008 19th International Conference on Pattern Recognition, pages 1–4. IEEE; 2008.
87. Freund Y. A more robust boosting algorithm. arXiv preprint arXiv:0905.2138; 2009.
88. Hastie T, Tibshirani R, Friedman J, Franklin J. The elements of statistical learning: data mining, inference and prediction. Math Intell. 2005;27(2):83–5.