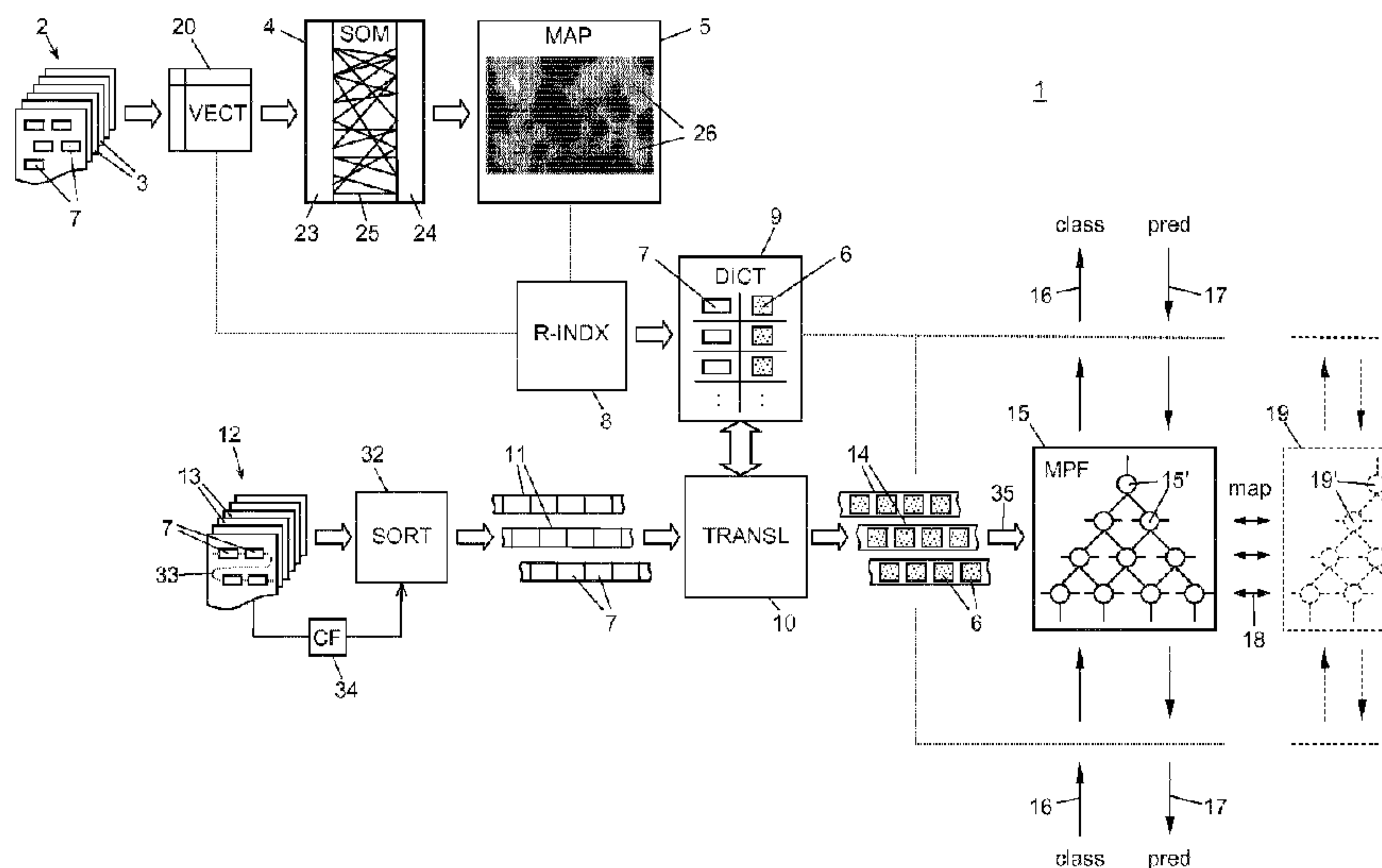




(86) Date de dépôt PCT/PCT Filing Date: 2013/02/22
(87) Date publication PCT/PCT Publication Date: 2013/09/19
(45) Date de délivrance/Issue Date: 2019/05/14
(85) Entrée phase nationale/National Entry: 2014/08/19
(86) N° demande PCT/PCT Application No.: EP 2013/053546
(87) N° publication PCT/PCT Publication No.: 2013/135474
(30) Priorité/Priority: 2012/03/15 (EP12159672.0)

(51) Cl.Int./Int.Cl. *G06N 3/08* (2006.01),
G06N 3/04 (2006.01), *G06F 17/28* (2006.01)
(72) Inventeur/Inventor:
DE SOUSA WEBBER, FRANCISCO EDUARDO, AT
(73) Propriétaire/Owner:
CORTICAL.IO AG, AT
(74) Agent: FETHERSTONHAUGH & CO.

(54) Titre : PROCÉDES, APPAREIL ET PRODUITS DE TRAITEMENT SEMANTIQUE D'UN TEXTE
(54) Title: METHODS, APPARATUS AND PRODUCTS FOR SEMANTIC PROCESSING OF TEXT



(57) Abrégé/Abstract:

The invention relates to a computer-implemented method of generating a computer-readable dictionary for translating text into a neural network-readable form, comprising: training a first neural network (4) of a self organizing map type with a first set (2) of first text documents (3) each containing one or more keywords (7) in a semantic context to map each text document (3) to a point (X_i/Y_j) in the self organizing map (5) by semantic clustering; determining, for each keyword (7) occurring in the first set (2), all points (X_i/Y_j) in the self organizing map (5) to which text documents (3) containing said keyword (7) are mapped, as a pattern (6) of points (X_i/Y_j) associated with said keyword (7); and storing all keywords (7) and associated patterns (6) as a computer-readable pattern dictionary (9). The invention further relates to computer-implemented methods of training neural networks, and classification, prediction and translation machines based on neural networks.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property

Organization

International Bureau

(43) International Publication Date

19 September 2013 (19.09.2013)



(10) International Publication Number

WO 2013/135474 A1

(51) International Patent Classification:

G06N 3/08 (2006.01) G06F 17/28 (2006.01)

G06N 3/04 (2006.01)

(21) International Application Number:

PCT/EP2013/053546

(22) International Filing Date:

22 February 2013 (22.02.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

12159672.0 15 March 2012 (15.03.2012) EP

(71) Applicant: CEPT SYSTEMS GMBH [AT/AT]; Fugbachgasse 12/7, A-1020 Wien (AT).

(72) Inventor: DE SOUSA WEBBER, Francisco Eduardo; Hainbachgasse 38, A-1140 Wien (AT).

(74) Agent: WEISER, Andreas; Kopfgasse 7, A-1130 Wien (AT).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: METHODS, APPARATUS AND PRODUCTS FOR SEMANTIC PROCESSING OF TEXT

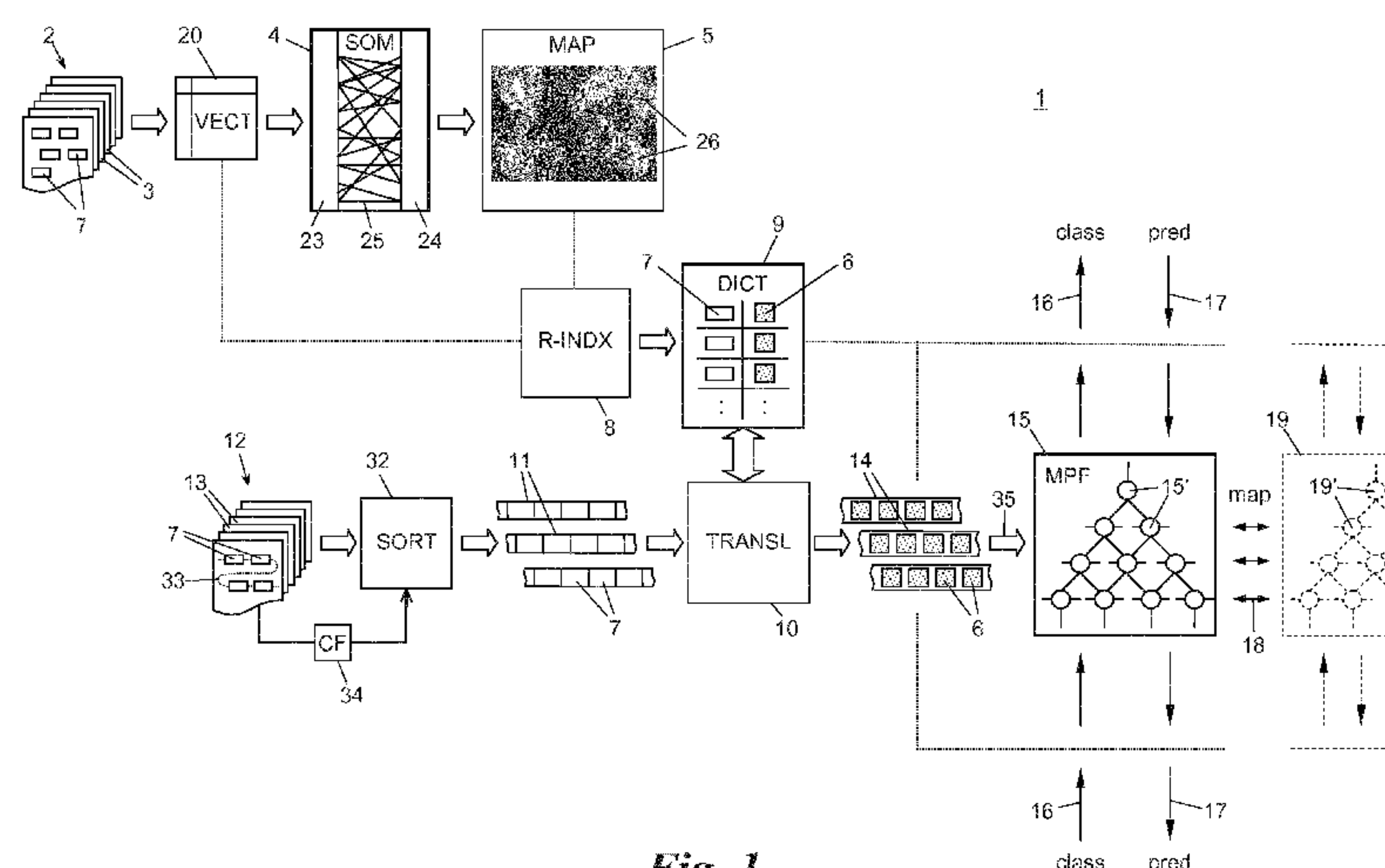


Fig. 1

(57) Abstract: The invention relates to a computer-implemented method of generating a computer-readable dictionary for translating text into a neural network-readable form, comprising: training a first neural network (4) of a self organizing map type with a first set (2) of first text documents (3) each containing one or more keywords (7) in a semantic context to map each text document (3) to a point (X_i/Y_j) in the self organizing map (5) by semantic clustering; determining, for each keyword (7) occurring in the first set (2), all points (X_i/Y_j) in the self organizing map (5) to which text documents (3) containing said keyword (7) are mapped, as a pattern (6) of points (X_i/Y_j) associated with said keyword (7); and storing all keywords (7) and associated patterns (6) as a computer-readable pattern dictionary (9). The invention further relates to computer-implemented methods of training neural networks, and classification, prediction and translation machines based on neural networks.



WO 2013/135474 A1

METHODS, APPARATUS AND PRODUCTS
FOR SEMANTIC PROCESSING OF TEXT

5 Field of the Invention

The present invention relates to a method of training a neural network, in particular for semantic processing, classification and prediction of text. The invention further relates
10 to computer-readable media and classification, prediction and translation machines based on neural networks.

Background of the Invention

15 In the context of the present disclosure, the term "neural network" designates a *computer-implemented, artificial* neural network. An overview of the theory, types and implementation details of neural networks is given e.g. in *Bishop C. M.*, "Neural Networks for Pattern Recognition", Oxford University Press,
20 New York, 1995/2010; or *Rey, G. D., Wender K. F.*, "Neurale Netze", 2nd edition, Hans Huber, Hofgrete AG, Bern, 2011.

The present invention particularly deals with the *semantic* processing of text by neural networks, i.e. analysing the meaning of a text by focusing on the relation between its words and
25 what they stand for in the real world and in their context. In the following, "words" (tokens) of a text comprise both words in the usual terminology of language as well as any units of a language which can be combined to form a text, such as symbols and signs. From these words, we disregard a set of all-too-
30 ubiquitous words such as "the", "he", "at" et cet. which have little semantic relevance to leave what we call "keywords" of a text.

Applications of semantic text processing are widespread and encompass e.g. classification of text under certain key-
35 words for relevance sorting, archiving, data mining and infor-

mation retrieval purposes. Understanding the meaning of keywords in a text and predicting "meaningful" further keywords to occur in the text is for example useful for semantic query expansion in search engines. Last but not least semantic text processing enhances the quality of machine translations by resolving ambiguities of a source text when considering its words in a larger semantic context.

Hitherto existing methods of semantic text processing, in particular for query expansion in search engines, work with large statistical indexes for keywords, their lemma (lexical roots) and statistical relations between the keywords to build large thesaurus files, statistics and dictionaries for relational analysis. Statistical methods are, however, limited in depth of semantic analysis when longer and more complex word sequences are considered.

On the other hand, neural networks are primarily used for recognising patterns in complex and diverse data, such as object recognition in images or signal recognition in speech, music or measurement data. Neural networks have to be correctly "trained" with massive amounts of training data in order to be able to fulfil their recognition task when fed with "live" samples to be analysed. Training a neural network is equivalent with configuring its internal connections and weights between its network nodes ("neurons"). The result of the training is a specific configuration of usually weighted connections within the neural network.

Training a neural network is a complex task on its own and involves setting a multitude of parameters with e.g. iterative or adaptive algorithms. Training algorithms for neural networks can therefore be considered as a technical means for building a neural network for a specific application.

While neural networks are currently in widespread use for pattern recognition in large amounts of numerical data, their application to text processing is at present limited by the

form in which a text can be presented to a neural network in a machine-readable form.

Summary of the Invention

5

It is an object of the invention to ameliorate the interface between text on the one hand and neural networks on the other hand in order to better exploit the analysing power of neural networks for semantic text processing.

10 In a first aspect of the invention, there is provided a computer-implemented method of training a neural network, comprising:

training a first neural network of a self organizing map type with a first set of first text documents each containing
15 one or more keywords in a semantic context to map each document to a point in the self organizing map by semantic clustering;

determining, for each keyword occurring in the first set, all points in the self organizing map to which first documents containing said keyword are mapped, as a pattern and storing
20 said pattern for said keyword in a pattern dictionary;

forming at least one sequence of keywords from a second set of second text documents each containing one or more keywords in a semantic context;

translating said at least one sequence of keywords into at
25 least one sequence of patterns by using said pattern dictionary; and

training a second neural network with said at least one sequence of patterns.

The second neural network trained with the innovative
30 method is configured for and ready to be used in a variety of applications, including the following applications:

i) processing of text which contains at least one keyword, comprising:

translating said at least one keyword into at least one
35 pattern by means of the pattern dictionary,

feeding said at least one pattern as an input pattern into said trained second neural network,

obtaining at least one output pattern from said trained second neural network, and

5 translating said at least output pattern into at least one keyword by means of the pattern dictionary;

ii) semantic classification of text, when a second neural network of a hierarchical type is used, wherein said at least one input pattern is fed into at least one lower layer of the hierarchy and said at least one output pattern is obtained from
10 at least one higher layer of the hierarchy; and

iii) semantic prediction of text, when a second neural network of a hierarchical type is used, wherein said at least one input pattern is fed into at least one higher layer of the hierarchy and said at least one output pattern is obtained from
15 at least one lower layer of the hierarchy.

In a further aspect the invention provides for a method of generating a computer-readable dictionary for translating text into a neural network-readable form, comprising:

20 training a neural network of a self organizing map type with text documents each containing one or more keywords in a semantic context to map each text document to a point in the self organizing map by semantic clustering;

determining, for each keyword occurring in the first set,
25 all points in the self organizing map to which text documents containing said keyword are mapped, as a pattern of points associated with said keyword; and

storing all keywords and associated patterns as a computer-readable dictionary.

30 The invention also provides for a computer readable dictionary of this kind which is embodied on a computer readable medium.

Further aspects of the invention are:

- a classification machine, comprising a neural network
35 of a hierarchical temporal memory type which has been trained

as said second neural network with a method according to the first aspect of the invention;

- a prediction machine, comprising a neural network of a hierarchical temporal memory type which has been trained as said second neural network with a method according to the first aspect of the invention;

- a translation machine, comprising such a classification machine, the neural network of which has been trained using first and second text documents in a first language, and a prediction machine, the neural network of which has been trained using first and second text documents in a second language, wherein nodes of the neural network of the classification machine are connected to nodes of the neural network of the prediction machine.

In all aspects the invention combines three different technologies in an entirely novel way, i.e. self-organizing maps (SOMs), the reverse-indexing of keywords in a SOM, and a target neural network exposed to text translated into a stream of patterns.

One of the principles of the invention is the generation of a novel type of a "keyword vs. pattern" dictionary (short: the "pattern dictionary") containing an association between a keyword and a two- (or more-) dimensional pattern. This pattern represents the semantics of the keyword within the context of the first document set. By choosing an appropriate collection of semantic contexts as first document set, e.g. articles of an encyclopaedia as will be described later on, each pattern reflects the semantic context and thus meaning of a keyword.

The patterns are generated by a SOM neural network, in particular a "Kohonen self organizing map" ("Kohonen feature map"). For details of SOMs see e.g. Kohonen, T., "The Self-Organizing Map", Proceedings of the IEEE, 78(9), 1464-1480, 1990; Kohonen, T., Somervuo, P., "Self-Organizing Maps of Symbol Strings", Neurocomputing, 21(1-3), 19-30, 1998; Kaski, S., Honkela, T., Lagus, K., Kohonen, T., "Websom-Self-Organizing

Maps of Document Collections", *Neurocomputing*, 21(1-3), 101-117, 1998; Merkl, D., "Text Classification with Self-Organizing Maps: Some Lessons Learned", *Neurocomputing*, 21(1-3), 61-77, 1998; Vesanto, J., Alhoniemi, E., "Clustering of the Self-Organizing Map", *IEEE Transactions on Neural Networks*, 11(3), 586-600, 2000; Pözlzbauer G., Dittenbach M., Rauber A., "Advanced Visualization of Self-Organizing Maps with Vector Fields", *IEEE Transactions on Neural Networks* 19, 911-922, 2006.

10 The SOM-generated patterns are subsequently used to translate keyword sequences from a second (training) set of text documents into pattern sequences to be fed into the second (target) neural network for pattern recognition. Pattern recognition is one of the core competences of neural networks. Since
15 each pattern represents an intrinsic meaning of a keyword, and a sequence of patterns represents a contextual meaning of keywords, the semantics of the keywords in the second document set is analysed by the target neural network under reference to, and before the background of, the intrinsic meaning of the keywords in the context of the first document set. As a result,
20 the target neural network can efficiently and meaningfully analyse the semantics of a text.

 The methods and apparatus of the invention are suited for training all sorts of target neural networks. A preferred application is the training of neural networks which are hierarchical and - at least partly - recurrent, in particular neural networks of the memory prediction framework (MPF) or hierarchical temporal memory (HTM) type. For theory and implementation details of MPFs and HTMs see e.g. Hawkins, J., George, D., Niemasik, J., "Sequence Memory for Prediction, Inference and Behaviour", *Philosophical Transactions of the Royal Society of London, Series B, Biological Sciences*, 364(1521), 1203-9, 2009;
30 Starzyk, J. A., He, H., "Spatio-Temporal Memories for Machine Learning: A Long-Term Memory Organization", *IEEE Transactions*

on Neural Networks, 20(5), 768-80, 2009; Numenta, Inc., "Hierarchical Temporal Memory Including HTM Cortical Learning Algorithms", Whitepaper of Numenta, Inc., Version 0.2.1, September 12, 2011; Rodriguez A., Whitson J., Granger R., "Derivation and
5 Analysis of Basic Computational Operations of Thalamocortical Circuits", Journal of Cognitive Neuroscience, 16:5, 856-877, 2004; Rodriguez, R. J., Cannady, J. A., "Towards a Hierarchical Temporal Memory Based Self-Managed Dynamic Trust Replication Mechanism in Cognitive Mobile Ad-hoc Networks", Proceedings of
10 the 10th WSEAS international conference on artificial intelligence, knowledge engineering and data bases, 2011; as well as patents (applications) Nos. US 2007/0276774 A1, US 2008/0059389 A1, US 7 739 208 B2, US 7 937 342 B2, US 2011/0225108 A1, US 8 037 010 B2 and US 8 103 603 B2.

15 MPF and HTM neural networks store hierarchical and time-sequenced representations of input pattern streams and are particularly suited to grasp time-spanning and hierarchical semantics of text. Their nodes (neurons) on different hierarchical layers represent *per se* hierarchical abstractions (classes) of
20 keywords; *classification* (abstraction) is an intrinsic working principle of such networks when input is fed from bottom to top of the hierarchy, and *prediction* (detailing) is an intrinsic working principle when input is fed from top to bottom of the hierarchy.

25 In a further aspect of the invention the concept of nodes representing entire classes (abstractions, categories) of keywords is utilised to build a translation machine as a prediction machine mapped to node outputs of a classification machine.

30 According to a further aspect of the invention several second documents can be used and translated into training pattern streams to train the second neural network on a specific set of second documents.

In some embodiments of the invention the second documents are sorted by ascending complexity and, when training the second neural network, the separate sequences of patterns are fed into the second neural network in the sorting order of the second documents from which they have each been formed and translated. This leads to a faster training of the second neural network.

In some other aspects of the invention the complexity of a second document is ascertained on the basis of one or more of: the number of different keywords in that second document, the average length of a sentence in that second document, and the frequency of one or more keywords of the first set in that second document.

According to one aspect of the present invention, there is provided A computer-implemented method of generating a computer-readable dictionary for translating text into a neural network-readable form, comprising: training a first neural network of a self organizing map type with a first set of first text documents each containing one or more keywords in a semantic context, the first neural network being trained with input vectors each representing a document of the first set and its keyword contents, to map each text document to a point in the self organizing map by semantic clustering, as a result of which training, in the map the documents have been mapped to individual points of the map; determining, for each keyword occurring in the first set, all points in the self organizing map to which text documents containing said keyword are mapped, as a two- or more-dimensional, as a pattern of points associated with said keyword; and storing all keywords and

associated patterns as a computer-readable pattern dictionary, each pattern being associated to one keyword in the pattern dictionary.

Brief Description of the Drawings

5 The invention is further described in detail under reference to the accompanying drawings, in which:

Fig. 1 is an overview flowchart of the method of the invention, including block diagrams of first and second neural networks, a pattern dictionary, as well as classification,
10 prediction and translation machines according to the invention;

Fig. 2 is a flowchart of the vector processing stage for the first document set as input vector to the first neural network in Fig. 1;

Fig. 3 is an exemplary self organizing map (SOM)
15 created as output of the first neural network in Fig. 1;

Fig. 4 is a flowchart of the reverse-indexing stage, receiving inputs from the vector processing stage and the SOM, to create the pattern dictionary in Fig. 1;

Fig. 5 shows reverse-indexed SOM representations with
20 exemplary patterns for two different keywords within the SOM;

Fig. 6 shows examples of some predetermined patterns for stop words (non-keywords);

Fig. 7 is a flowchart of the keyword sequence extraction stage for the second set of second documents in Fig. 1;

Fig. 8 shows the result of an optional document sorting step for the second documents of the second set;

5 Fig. 9 is a flowchart of the steps of translating a keyword sequence into a pattern sequence in Fig. 1; and

Fig. 10 shows an exemplary hierarchical node structure of a MPF used as the second neural network in Fig. 1.

10 Detailed Description of the Invention

In a general overview, Fig. 1 shows a semantic text processing method and system 1 which uses a first set 2 of first text documents 3 to train a first neural network 4. The first
15 neural network 4 is of the self organizing map (SOM) type and creates a self organizing map (SOM) 5. From SOM 5 patterns 6 representative of keywords 7 occurring in the first document set 2 are created by reverse-indexing stage 8 and put into a pattern dictionary 9.

20 The pattern dictionary 9 is used in a translation stage 10 to translate keyword sequences 11 extracted from a second set 12 of second documents 13 into pattern sequences 14. With the pattern sequences 14 a second neural network 15 is trained. The second neural network 15 is preferably (although not necessarily) of the memory prediction framework (MPF) or hierarchical
25 temporal memory (HTM) type. The trained second neural network 15 can then be used either to semantically classify text translated with pattern dictionary 9, see path 16, or to semantically predict text translated with pattern dictionary 9, see
30 path 17. A further optional application of the trained second neural network 15 is a hierarchical mapping, see paths 18, to an optional third neural network 19 which is similar in construction to the second neural network 15 but has been trained in a different language than the second neural network 15; node
35 mappings 18 then represent semantic coincidences between seman-

tic nodes 15' of first language network 15 and semantic nodes 19' of second language network 19.

The processes and functions of the components shown in Fig. 1 are now described in detail with reference to Figs. 2 to 5 10.

Fig. 2 shows a preprocessing and vectorisation step 20 to index and vectorise the first set 2 of first documents 3. In step 20 from first set 2 a sequence of input vectors 21 is produced, one vector 21 for each first document 3, as an input 10 training vector set or matrix (table) 22 applied to the input layer 23 of the first neural network (SOM) 4. As known to the man skilled in the art, SOM neural network 4 usually comprises only two layers, an input layer 23 and an output layer 24 of neurons (nodes), interconnected by connections 25 the weights 15 of which can be represented by a weighting matrix. SOM neural networks can be trained with unsupervised learning algorithms wherein the weights of the weighting matrix are self-adapting to the input vectors, to specifically map nodes of the input layer 23 to nodes of the output layer 24 while taking into account the spatial relation of the nodes of the output layer 24 20 in a two- (or more-) dimensional map 5. This leads to maps 5 which cluster input vectors 21 with regard to their similarity, yielding regions 26 in the map 5 with highly similar input vectors 21. For details of SOM neural networks, see the above-cited bibliographic references. 25

The first set 2 and the first documents 3 therein are chosen in such a number and granularity, e.g. length of the individual documents 3, that each of the documents 3 contains a number of e.g. 1 to 10, 1 to 20, 1 to 100, 1 to 1000 or more, 30 preferably about 250 to 500, keywords 7 in a semantic context. A first document 3 may contain - in addition to the keywords 7 - words of little semantic relevance (such as articles "a", "the" et cet.) which are usually called stop words, here non-keywords.

The number of documents 3 in the set 2 is chosen to obtain a representative corpus of semantic contexts for the keywords 7, e.g. thousands or millions of documents 3. In an exemplary embodiment, about 1.000.000 documents 3, each comprising about 250 to 500 keywords 7, are used as first document set 2.

The length (keyword count) of the documents 3 should be fairly consistent over the entire set 2, keywords 7 should be evenly and sparsely distributed over the documents 3 in the set 2, and each document 3 should contain a good diversity of keywords 7.

Keywords 7 can also be roots (lemma) of words, so that e.g. for singular and plural forms (cat/cats) or different verb forms (go/going) only one keyword 7 is taken into account. Keywords 7 can thus be both, specific word forms and/or roots of words. After stripping-off words incapable of building significant keywords, such as stop words, each document 3 can be considered a "bag of words" of keywords 7.

In a practical embodiment, a suitable first set 2 can e.g. be generated from articles from an encyclopaedia, such as Wikipedia® articles obtained under the "Creative Commons Attribution Licence" or the "GNU Free Documentation Licence" of the Wikipedia® project. Such encyclopaedic articles, or entries, respectively, can be parsed according to chapters, paragraphs et cet. into documents 3 of fairly uniform length, so that each document 3 contains keywords 7 in a *semantic, i.e. meaningful* context.

To generate the vectors 21, an index of all keywords 7 occurring in the entire set 2 is generated and spread horizontally as column heading 27 of the matrix (table) 22. Vice versa, document identifications ("id") of all documents 3 in the entire set 2 are spread vertically as row heading 28 in matrix 22. Then for each occurrence of a specific keyword 7 in a specific document 3, a flag or binary "1" is put into the respective cell of the matrix 22. Thus, in matrix 22 one horizontal row represents a normalized "keyword-occurrence" vector 21

for one document 3, wherein a binary "1" at a specific keyword position (column position) indicates that this keyword 7 is contained in the "bag of words" of this document 3; and a binary "0" indicates the absence of this keyword 7 in this document 3. Or, the other way around, each column in matrix 22 shows for a specific keyword 7 all those documents 3 marked with a binary "1" which contain that keyword 7.

The input vectors 21, i.e. rows of the matrix 22 representing the documents 3 and their keyword contents, are then supplied successively to the input layer 23 of SOM neural network 4 to train it. This means that if a first set 2 of e.g. 1.000.000 first documents 3 is used, a training run of 1.000.000 vector inputs is supplied to the first neural network 4.

As a result of this training run, the output layer 24 of SOM neural network 4 has produced map 5 in which documents 3 (vectors 21) have been mapped to individual points ("pixels") X_i/Y_j of the map 5, clustered by similarity. Fig. 3 shows an example of a map 5. To each map point X_1/Y_1 , X_2/Y_2 , ..., X_i/Y_j , ..., zero, one or more document(s) 3 with their bag of keywords 7 has/have been mapped. Documents 3 (vectors 21) are identified in map 5 e.g. by their document id from row heading 28. By that SOM clustering process, different documents 3 which contain very similar keywords 7, e.g. which coincide in 80% or 90% of their keywords, are mapped in close spatial relationship to one another, thus forming semantic "regions" 26_a, 26_b, 26_c, 26_d, et cet. in map 5.

Next, in the reverse-indexing stage 8 of Fig. 4, on the basis of matrix 22 for a given keyword 7 from keyword index 27 all those documents 3 are identified which contain that keyword 7. This can e.g. be easily done by retrieving all binary "1" in the specific column of the given keyword 7 in matrix 22 and looking-up the id of the document 3 listed in row heading 28.

For those documents 3 which have been ascertained as containing that given keyword 7, all map points X_i/Y_j referencing

that specific document id are determined from map 5. This set $\{X_i/Y_j\}$ of map points represents the pattern 6. The pattern 6 is representative of the semantic contexts in which that given keyword 7 occurred in the first set 2: The spatial (i.e. two-
5 or more-dimensional) distribution of the points X_i/Y_j in the pattern 6 reflects those specific semantic regions $26_a, 26_b, \dots$ in the context of which the keyword 7 occurred in the first set 2.

Pattern 6 can be coded as a binary map 31, see Fig. 4, and
10 also regarded as a binary "fingerprint" or "footprint" of the semantic meaning of a keyword 7 in a document collection such as the first set 2. If the first set 2 covers a vast variety of meaningful texts in a specific language, the pattern 6 is of high semantic significance of the keyword 7.

15 The spatial resolution of the pattern 6 can be equal to or lower than the spatial resolution of the SOM neural network 4 and/or the map 5. The spatial resolution of the latter can be chosen according to the required analysis performance: For example, map 5 can be composed of millions of map points X_i/Y_j ,
20 e.g. 1000 x 1000 points, and pattern 6 can have the same resolution for high precision, or a coarser resolution for lower memory requirements.

Fig. 5 shows an example of two different patterns 6 (depicted as black dots) overlying map 5 for ease of comprehension. In this example, regions $26_a, 26_b, 26_c, 26_d$ have been
25 manually labeled with semantic classes such as "predator", "felines", "my pet" and "canis". This is only for exemplary purposes; it should be noted that such a labeling is not necessary for the correct functioning of the present methods, processes
30 and algorithms which only require the spatial SOM distribution of the map points X_i/Y_j .

In the left representation of Fig. 5, all documents 3 in which the keyword "cat" occurred have been marked with a dot. In the right representation of Fig. 5, all documents 3 contain-
35 ing the keyword "dog" have been marked with a dot. It can eas-

ily be seen that "cat" documents primarily fall, or are clustered, into regions 26_b ("my pet") and 26_d ("felines"), whereas "dog" documents 3 are primarily clustered into regions 26_b ("my pet") and 26_c ("canis").

5 Returning to Fig. 1, for each keyword 7 occurring in the first set 2 the respective pattern 6 is stored in pattern dictionary 9 in the form of a two-way mapping, i.e. association between a keyword 7 and its pattern 6. Pattern dictionary 9 constitutes a first, intermediate product of the method and
10 system 1 of Fig. 1. Pattern dictionary 9 can be stored ("embodied") on a computer-readable medium, e.g. a data carrier such as a hard disk, CD-Rom, DVD, memory chip, internet server, a cloud storage in the Internet et cet.

It should be noted that the generation of pattern dictionary 9 may involve the use of massive processing power for
15 training the first neural network 4 and reverse-indexing the map 5. Therefore, pattern dictionary 9 is preferably pre-computed once and can then be used repeatedly in the further stages and modules of the processes and machines of Fig 1.

20 Based on different first sets 2 of first documents 3, which can e.g. be chosen application-specific and/or and language-specific, different pattern dictionaries 9 can be pre-computed and distributed on computer-readable media to those entities which perform the subsequent stages and implement the
25 subsequent modules of the processes and machines which will now be described in detail.

In these subsequent stages and modules the second (target) neural network 15 is trained for semantic text processing on the basis of the second set 12 of second documents 13. While
30 the second set 12 could be identical with the first set 2, in practice the second set 12 may comprise a subset of the first set 2 or indeed quite different application-specific second documents 13. For example, while the first set 2 comprises a vast number of general ("encyclopaedic") documents 3, the sec-
35 ond set 12 can be an application-specific user data set of user

documents 13 which e.g. need to be searched by semantic query (keyword) expansion, classified or sorted by semantic classification, or translated by semantic translation. Pattern dictionary 9 then reflects background semantic knowledge about general semantic meanings of keywords 7, while second neural network 15 performs an in-depth analysis of a user data set 12 of user documents 13.

User documents 13 can e.g. be records from product databases, web-pages, patent documents, medical records or all sorts of data collections which shall be analysed by the second neural network 15. One prerequisite for the second set 12 is that it has been written in the same language as the first set 2 since otherwise the pattern dictionary 9 could not be applied meaningfully to the second set 12. Furthermore, it is preferably - although not obligatory - that keywords 7 occurring in the second documents 13 of the second set 12 are comprised within the entire set, i.e. index 27, of keywords 7 in the first set 2 so that keywords 7 of the second set 12 are listed and can be looked-up in the pattern dictionary 9.

In the pattern dictionary 9, stop words or non-keywords can either be disregarded or incorporated as predetermined or preconfigured symbolic patterns such as those shown in Fig. 6.

For training the second neural network 15, in a first stage 32 sequences 11 of keywords 7 are extracted from the second set 12. Figs. 1, 7 and 8 show this extraction stage in detail. Basically it would be sufficient if only one or a few second document(s) 13 is/are sequentially read, word by word, line by line, paragraph by paragraph, chapter by chapter, document by document, in a normal reading sequence 33. Stop words or non-keywords could be skipped (or dealt with separately as described in Fig. 6), and the result is one sequence 11 of keywords 7. Preferably, however, the second set 12 is split into a multitude of second documents 13, and one sequence 11 of keywords 7 is generated for one document 13. The sequences 11 are

then used - e.g. in the order of the documents 13, they originate from - as training input for the second neural network 15.

Training of the second neural network 15 can be accelerated if an optional sorting of the documents 13 and/or sequences 11 is performed in extraction stage 32. For this optional sorting, a "complexity factor" CompF is calculated in a process 34 for each document 13 of the second set 12. The complexity factor CompF can be calculated on the basis of one or more of the following parameters of a document 13:

- 10 - the number of different keywords 7 in a document 13;
- the average word count of a sentence or paragraph in a document 13;
- the frequency, or diversity, of one or more of the keywords 7, e.g. of all keywords 7 of the first set 2, in a document 13;
- 15 - the frequency of one or more of the keywords 7, e.g. all keywords 7, of a document 13 in the entire first set 2 or another text corpus representative of colloquial language, e.g. a collection of newspapers.

20 In extraction stage 32 the documents 13 can then be sorted (ranked) according to ascending complexity factor CompF, see Fig. 8. In this way the second neural network 15 is fed with sequences 11 of increasing complexity, e.g. primitive or simple sequences 11 or sequences 11 with a modest diversity of keywords 7 are used first, and sequences 11 with complicated semantic and linguistic structures are used last for training the
25 second neural network 15.

Before being fed to the second neural network 15 the sequences 11 of keywords 7 are translated in translation stage 10
30 on the basis of the pattern dictionary 9. Each keyword 7 in a sequence 11 is looked-up in pattern dictionary 9, the associated pattern 6 is retrieved, and the results are sequences 14 of patterns 6, one pattern sequence 14 for each document 13. Each pattern sequence 14 can be considered as a time-series or
35 "movie clip" of patterns 6 representing the semantic context of

keywords 7 in a document 13 within the global semantic context of the first document set 2.

It should be noted that in simple embodiments it would be sufficient to use only one long sequence 14 of patterns 6 to
5 train the second neural network 15. Preferably a large number of pattern sequences 14 (a "sequence of sequences") is used, each pattern sequence 14 representing a time-lined training vector (matrix) for the second neural network 15. Fig. 9 shows an example of the translation stage 10 translating a keyword
10 sequence 11 into a pattern sequence 14.

In the training stage (arrow 35 in Fig. 1) the second neural network 15 is fed successively with pattern sequences 14 to learn the patterns 6 and their sequences over time. As discussed at the outset, all types of neural networks adapted for
15 time-series processing of patterns can be used, e.g. feed-forward pattern processing neural networks with sliding windows. Alternatively and preferably, recurrent or at least partly recurrent neural networks, with or without delay loops, can be used to learn and remember temporal sequences, e.g.
20 self- or auto-associative neural networks.

In advantageous embodiments the second neural network 15 is also *hierarchical* in that upper layers of the hierarchy comprise fewer nodes (neurons) than lower layers of the hierarchy. Fig. 10 shows an example of such a hierarchical network, in
25 particular a memory prediction framework (MPF) which also contains lateral (intra-layer, see Fig. 1) and vertical (cross-layer) feedback connections for learning temporal sequences. A preferred form of such a MPF architecture are neural networks of the hierarchical temporal memory (HTM) type. Theory and im-
30 plementation details of MPF and HTM neural networks are described in the above cited papers, the disclosures of which are herein incorporated by reference.

MPF and HTM networks develop - in trained configuration - neurons (nodes) within the hierarchy which stand for abstrac-
35 tions (classifications) of firing patterns of neurons (nodes)

in lower layers of the hierarchy. By using trained recurrent (feedback) intra-layer and cross-layer connections, in particular between nodes of "columnar" sub-layer structures, they can model the temporal behaviour of entire temporal streams of firing patterns. In this way, MPF and HTM networks can learn, remember and classify streams of patterns and both recognise pattern sequences as well as predict possible future pattern sequences from past pattern sequences.

Once the neural network 15 has been trained with the pattern sequences 14, new patterns 6 or new pattern sequences 14 can be applied as new inputs to a "classification" input at lower hierarchy levels of the network 15, to obtain semantic classifications/abstractions as patterns from the outputs of nodes at higher hierarchy levels, see route 16; or, new patterns 6 or new pattern sequences 14 can be fed into "prediction" inputs at higher hierarchy levels and predicted patterns (semantical predictions) can be obtained from lower levels in the hierarchy, see route 17.

As can be seen in Fig. 1, pattern dictionary 9 is used on both routes 16, 17 to translate any new "query" sequence of keywords 7 into a "query" sequence 14, and to retranslate the output patterns of the neural network 15 into "resulting" classification or prediction keywords 7.

Classification route 16 can thus be used to classify a query text by the trained neural network 15 using the pattern dictionary 9 on the input and output interfaces of the network 15; and prediction route 17 can be used to predict keywords from a query text, e.g. to "expand" a query keyword phrase to further (predicted) keywords 7 which semantically match the query phrase, using pattern dictionary 9 at both input and output interfaces of the neural network 15.

A further application of the trained neural network 15 is shown in dotted lines in Fig. 1. A third neural network 19 trained with sets 2, 12 of documents 3, 13 in a different language than that in which the neural network 15 had been trained

is nodewise mapped - if corresponding classification nodes 15',
19' within the networks 15 and 19 can be identified - to the
second network 15. On the inputs and outputs 38, 39 of the
third neural network 19 a further pattern dictionary 9, gener-
5 ated from a document set 2 in the language of the third network
19, is used. In this way, semantic translations between two
languages can be obtained by semantic mapping of two trained
MPF or HTM networks 15, 19.

While the invention has been described with reference to
10 two-dimensional maps 5 and patterns 6, it should be noted that
the first neural network 4 could also generate three- or more-
dimensional maps 5, thus leading to three- or more-dimensional
patterns 6 in pattern dictionary 9, subsequently to three- or
more-dimensional pattern sequences 14 and second and third neu-
15 ral networks 15, 19 working in three or more dimensions.

The invention is in no way limited to the specific embodi-
ments described as examples in detail but comprises all vari-
ants, modifications and combinations thereof which are encom-
passed by the scope of the appended claims.

CLAIMS:

1. A computer-implemented method of generating a computer-readable dictionary for translating text into a neural network-readable form, comprising:

5 training a first neural network of a self organizing map type with a first set of first text documents each containing one or more keywords in a semantic context, the first neural network being trained with input vectors each representing a document of the first set and its keyword contents, to map each
10 text document to a point in the self organizing map by semantic clustering, as a result of which training, in the map the documents have been mapped to individual points of the map;

determining, for each keyword occurring in the first set, all points in the self organizing map to which text documents
15 containing said keyword are mapped, as a two- or more-dimensional, as a pattern of points associated with said keyword; and

storing all keywords and associated patterns as a computer-readable pattern dictionary, each pattern being
20 associated to one keyword in the pattern dictionary.

2. The method of claim 1 for training a neural network, further comprising:

forming at least one sequence of keywords from a second set of second text documents each containing one or more
25 keywords in a semantic context;

translating said at least one sequence of keywords into at least one sequence of patterns by using said pattern dictionary; and

training a second neural network with said at least one
5 sequence of patterns.

3. The method of claim 2, wherein the second neural network is hierarchical and at least partly recurrent.

4. The method of claim 2, wherein the second neural network is a memory prediction framework.

10 5. The method of claim 2, wherein the second neural network is a hierarchical temporal memory.

6. The method of any one of the claims 1 to 5, wherein the first neural network is a Kohonen self organizing map.

7. The method of any one of the claims 2 to 6, wherein
15 for each of the second documents of the second set a separate sequence of keywords is formed and translated into a separate sequence of patterns and the second neural network is trained successively with each of said separate sequences of patterns.

8. The method of claim 7, wherein the second documents
20 are sorted and, when training the second neural network, the separate sequences of patterns are fed into the second neural network in the sorting order of the second documents from which they have each been formed and translated.

9. The method of claim 8, wherein the second documents
25 are sorted by ascending complexity, wherein the complexity of a second document is ascertained on the basis of one or more of:

the number of different keywords in that second document, the average length of a sentence in that second document, the frequency of one or more keywords of the first set in that second document, the frequency of one or more keywords of that
5 second document in the first set or another text corpus.

10. The method of any one of the claims 2 to 9 for processing text containing at least one keyword, comprising:

translating said at least one keyword into at least one pattern by means of the pattern dictionary;

10 feeding said at least one pattern as an input pattern into said trained second neural network;

obtaining at least one output pattern from said trained second neural network; and

15 translating said at least one output pattern into at least one keyword by means of the pattern dictionary.

11. The method of claim 10 for semantic classification of text, wherein the second neural network is hierarchical, said at least one input pattern is fed into at least one lower layer of the hierarchy and said at least one output pattern is
20 obtained from at least one higher layer of the hierarchy.

12. The method of claim 10 for semantic prediction of text, wherein the second neural network is hierarchical, said at least one input pattern is fed into at least one higher layer of the hierarchy and said at least one output pattern is
25 obtained from at least one lower layer of the hierarchy.

13. A classification or prediction machine, comprising a neural network of a hierarchical type which has been trained as said second neural network with a method according to any one of the claims 2 to 9.

5 14. A translation machine, comprising

a classification machine according to claim 13, the neural network of which has been trained with a method according to one of the claims 2 to 9 using first and second text documents in a first language; and

10 a prediction machine according to claim 14, the neural network of which has been trained with a method according to one of the claims 2 to 9 using first and second text documents in a second language;

15 wherein nodes of the neural network of the classification machine are connected to nodes of the neural network of the prediction machine.

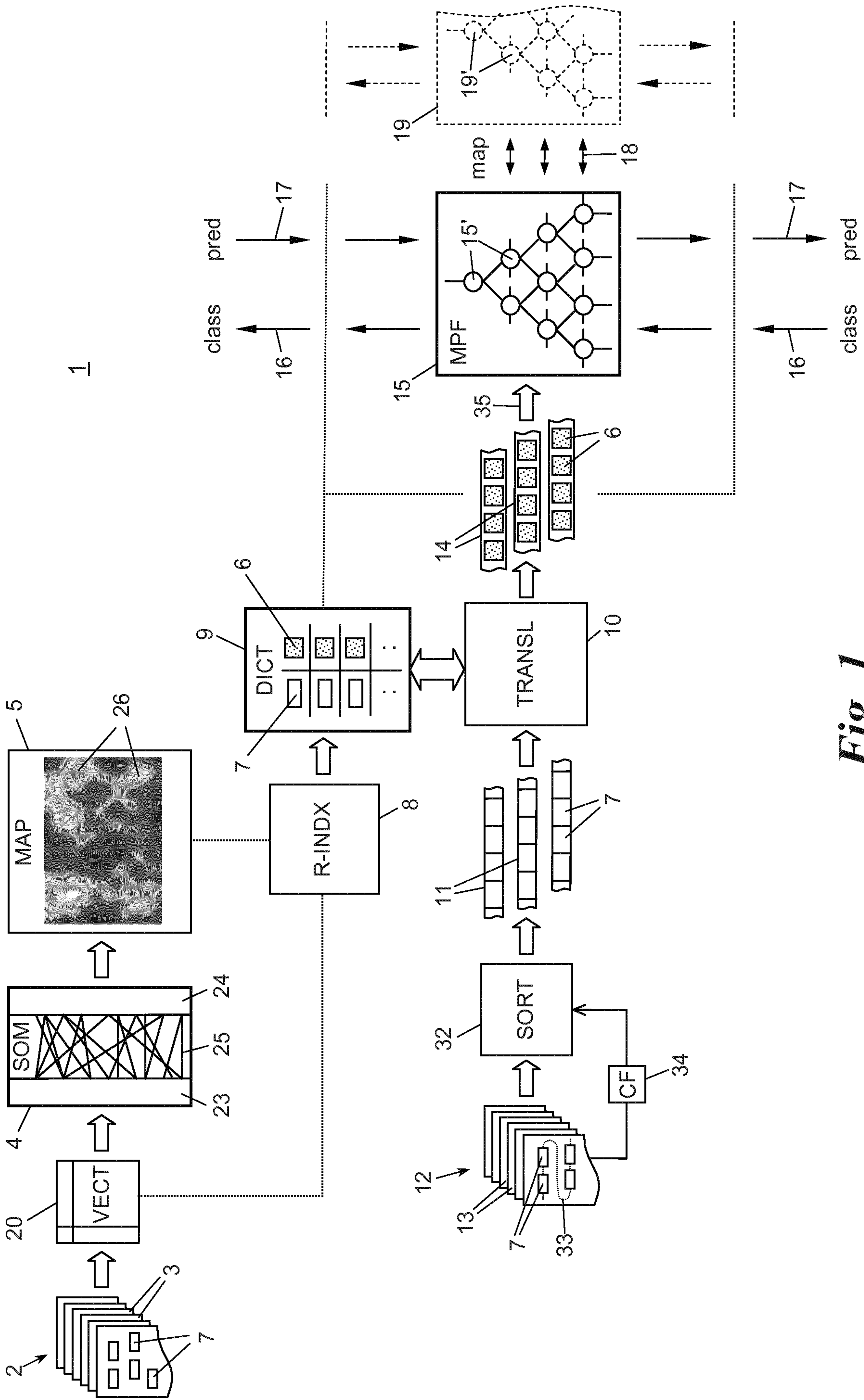


Fig. 1

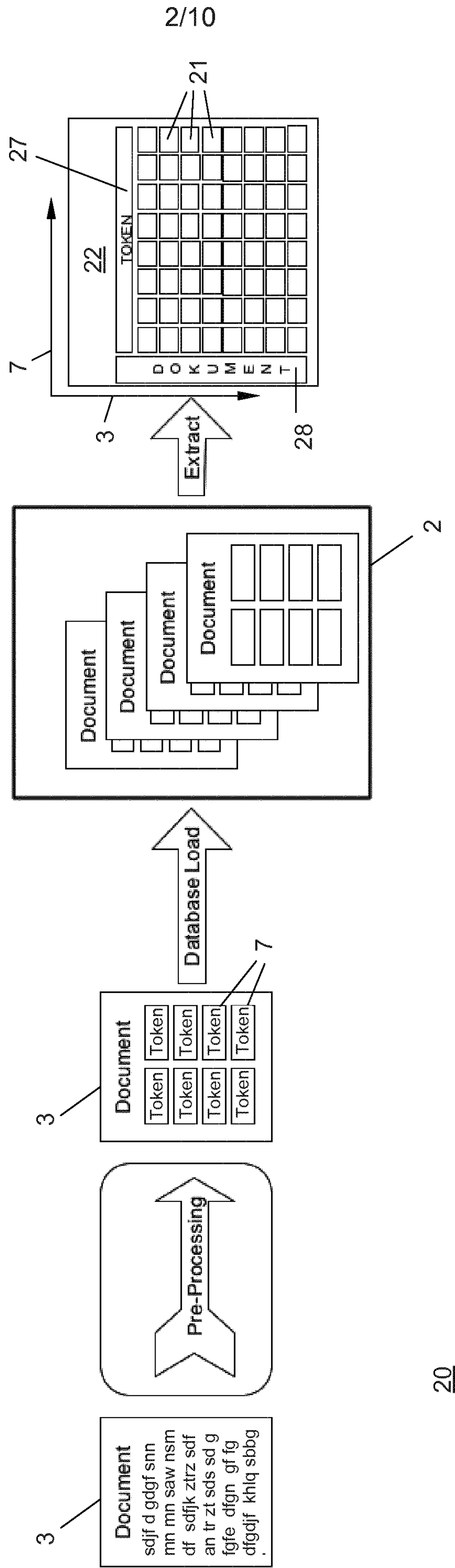


Fig. 2

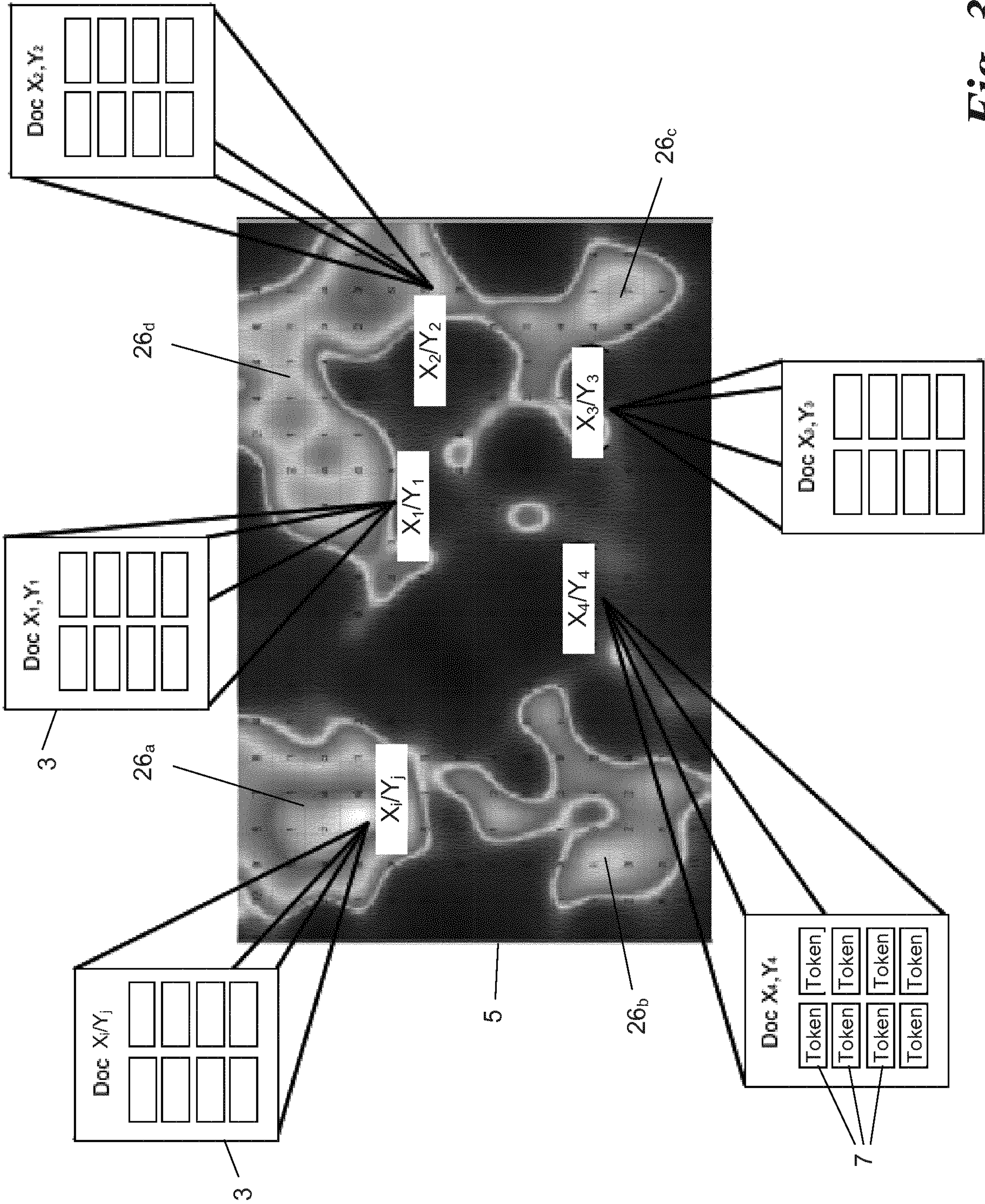


Fig. 3

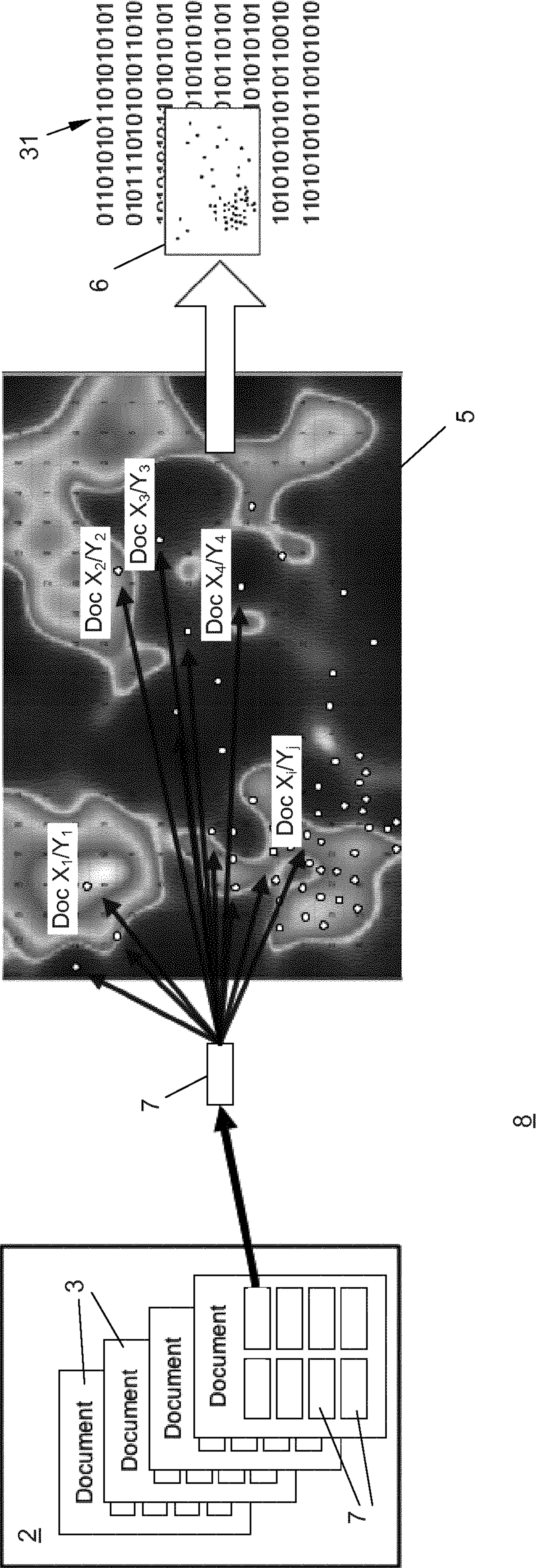


Fig. 4

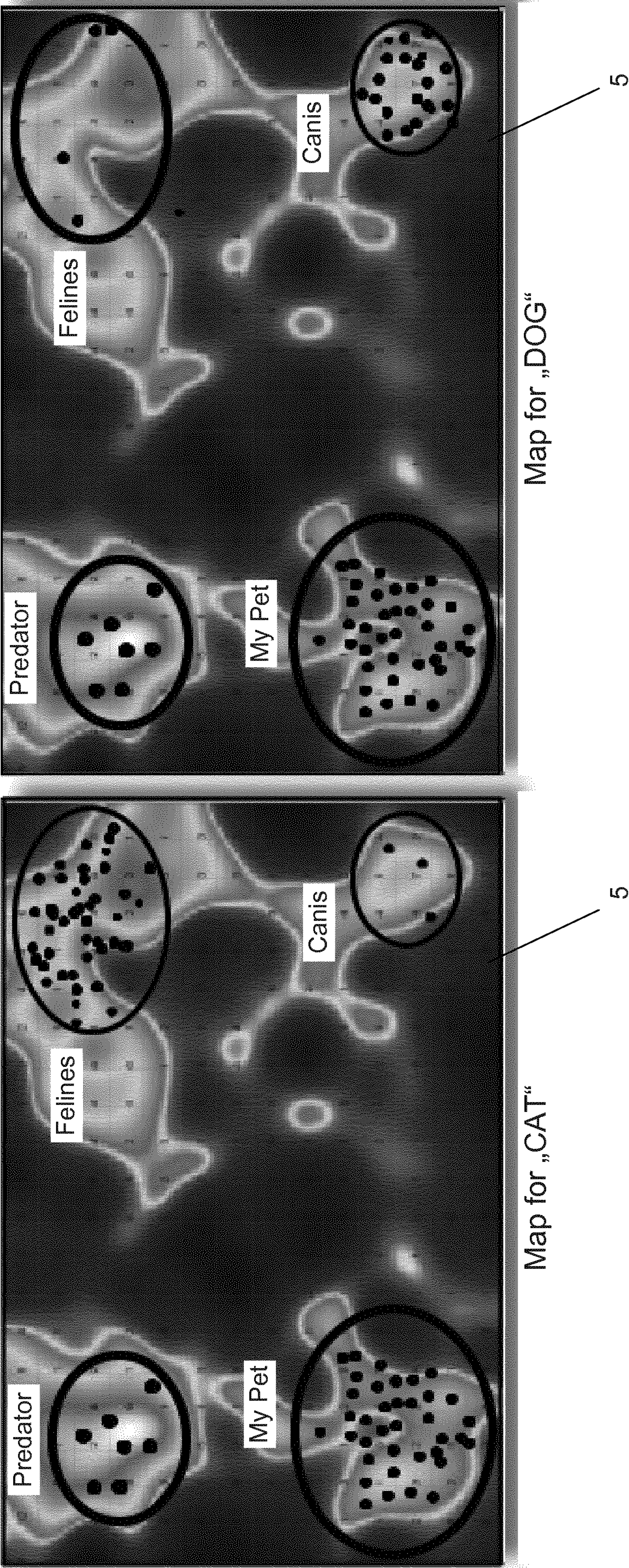


Fig. 5

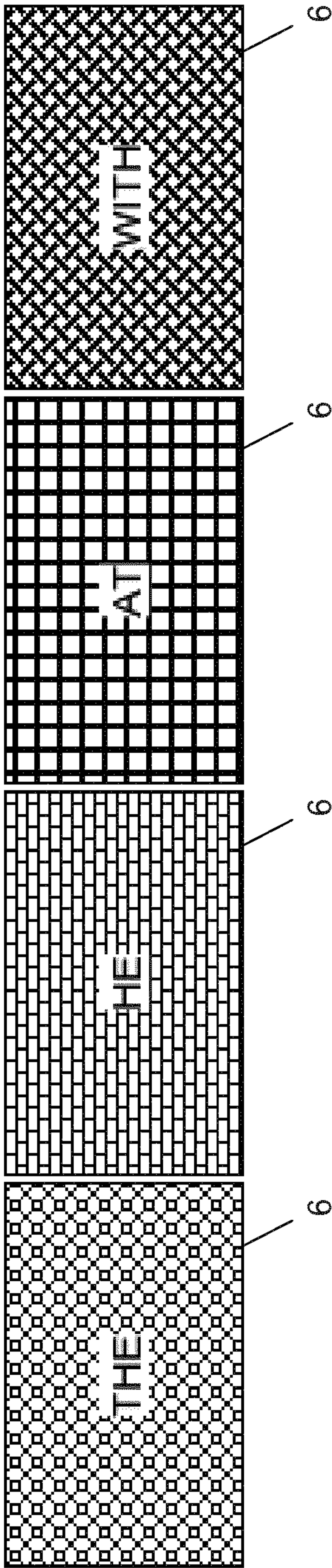


Fig. 6

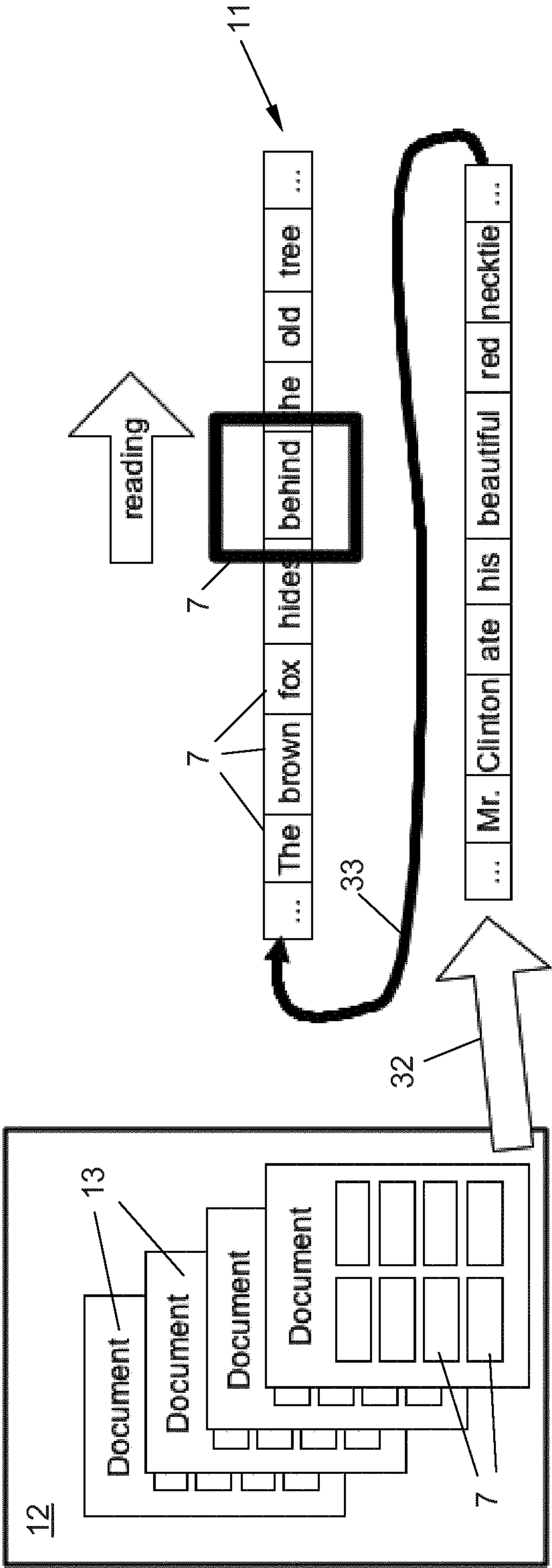


Fig. 7

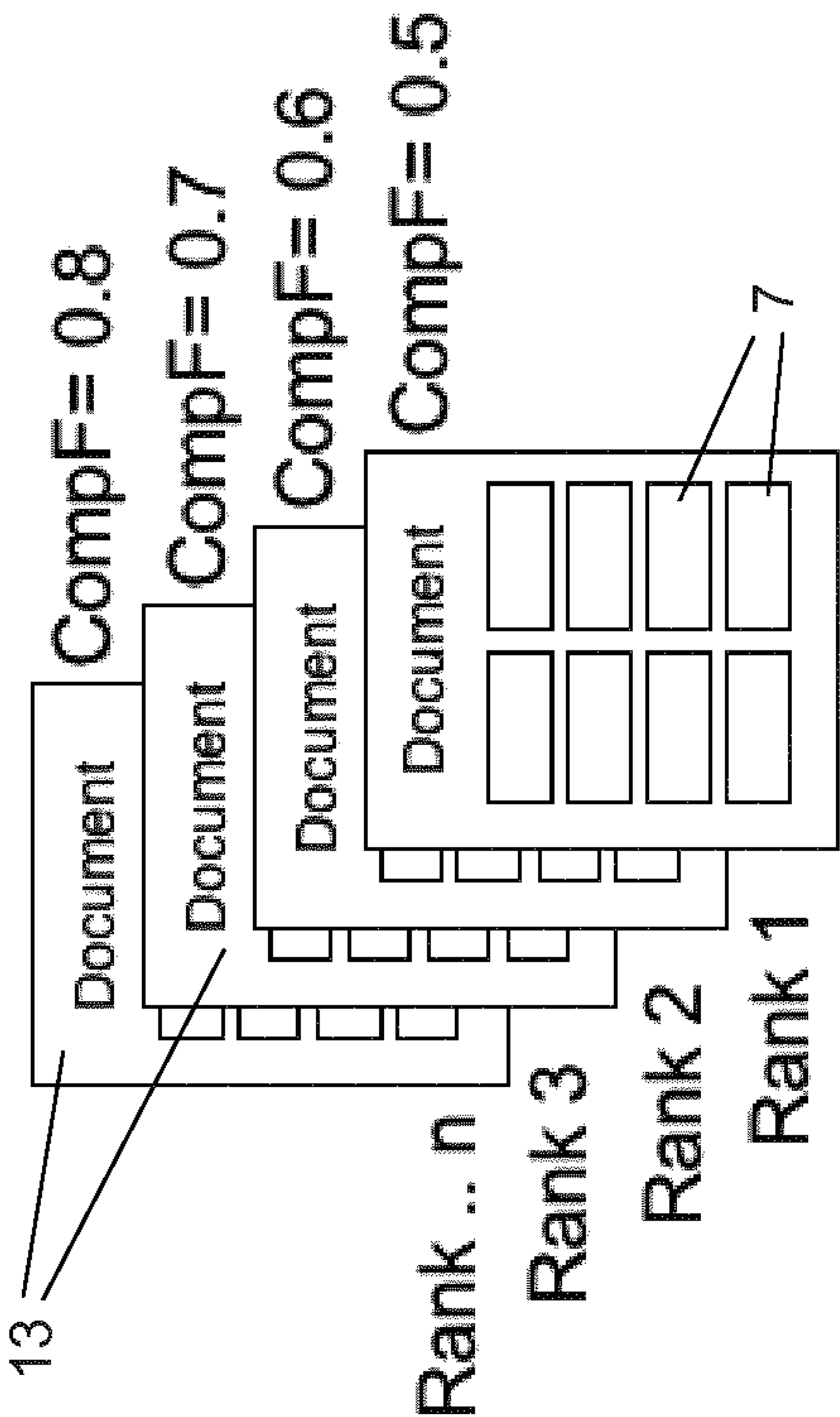


Fig. 8

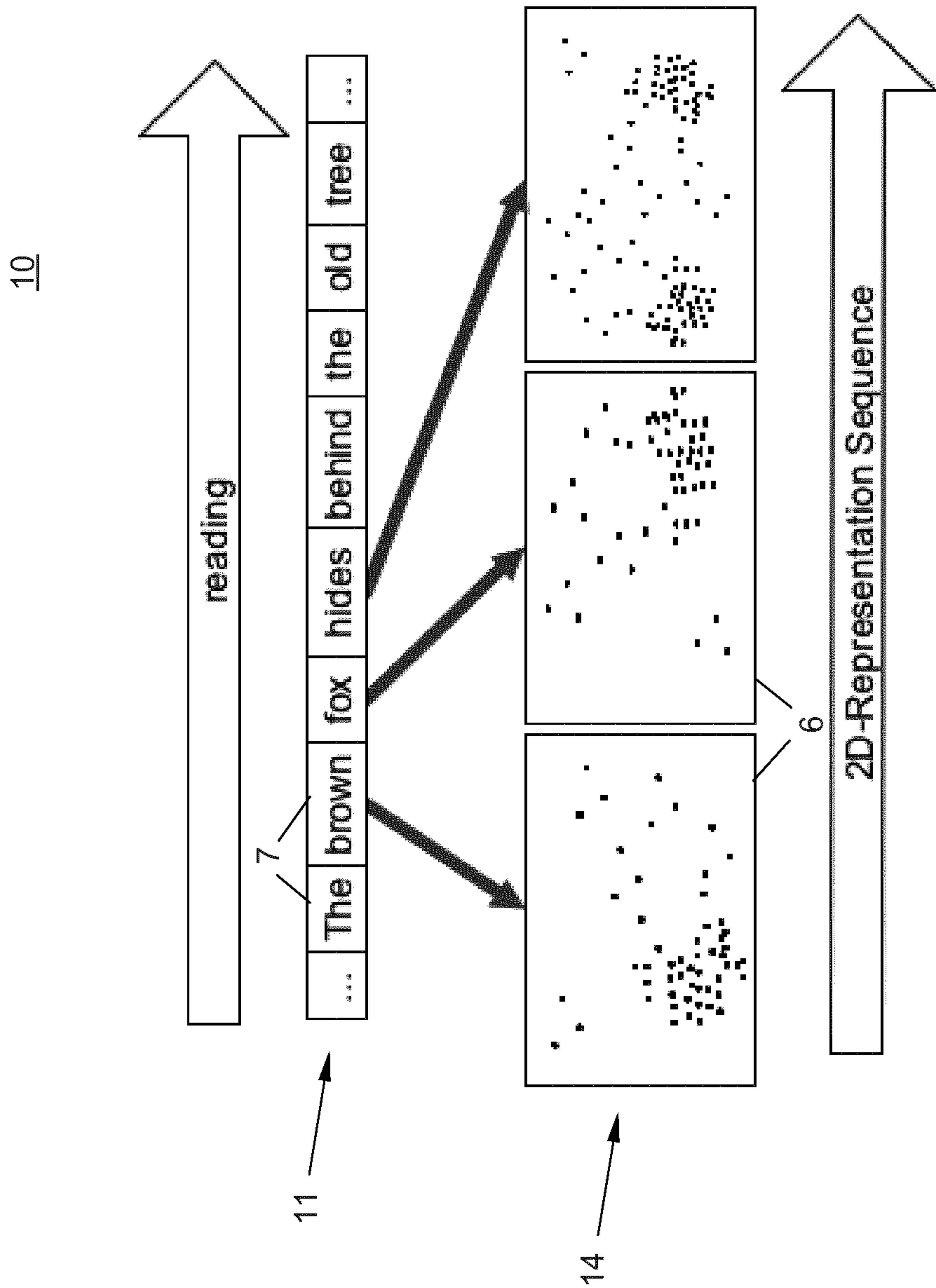


Fig. 9

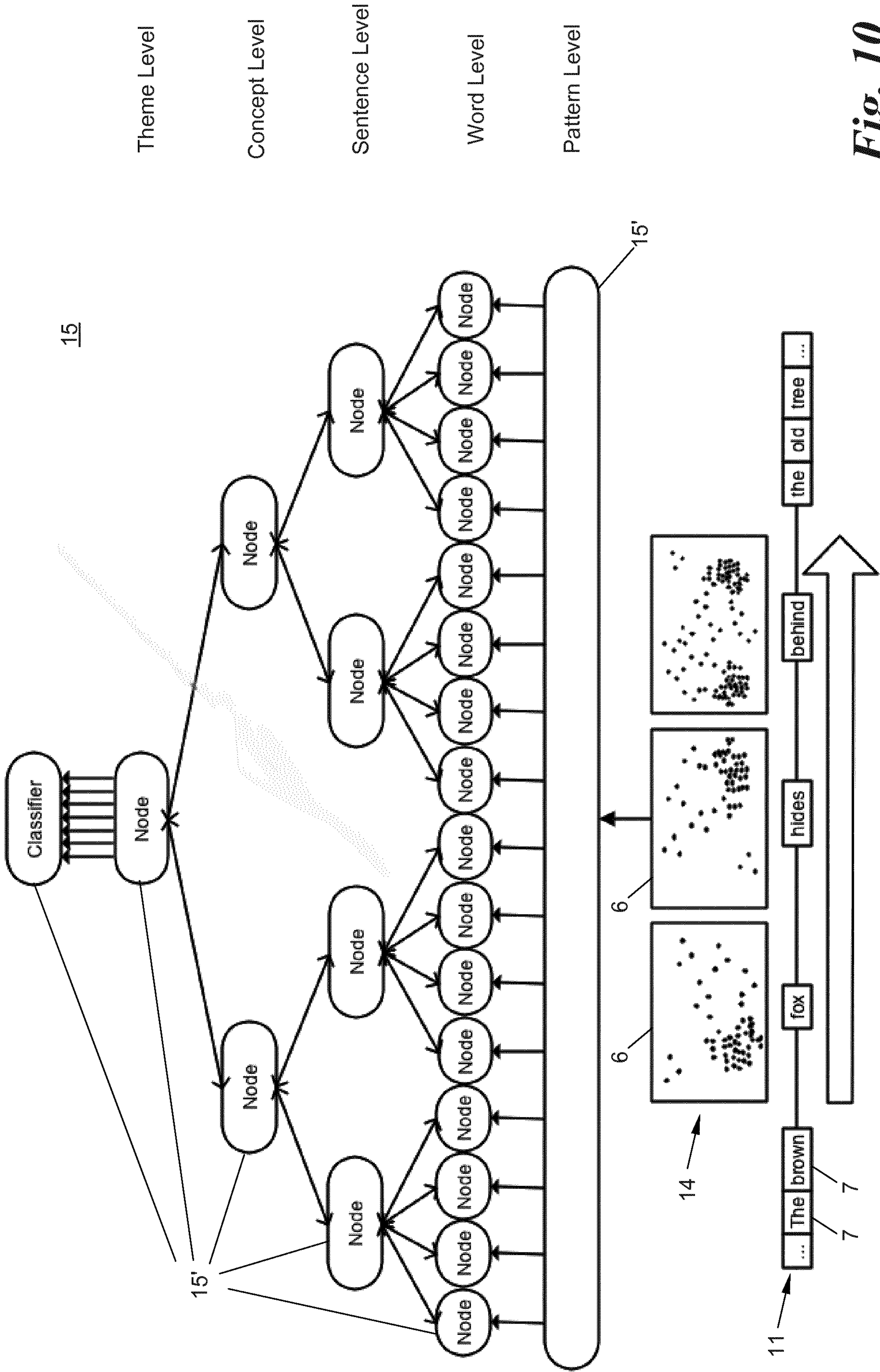


Fig. 10

