Workshop Notes

10th International Workshop
“What can FCA do for Artificial Intelligence?”
FCA4AI 2022

31st International Joint Conference on Artificial Intelligence
IJCAI-ECAI 2022

July 23 2022

Wien Messe, Vienna, Austria

Editors
Sergei O. Kuznetsov (HSE University Moscow)
Amedeo Napoli (LORIA Nancy)
Sebastian Rudolph (TU Dresden)

http://fca4ai.hse.ru/2022/

IJCAI ECAI
VIENNA 22

Preface

The nine preceding editions of the FCA4AI Workshop showed that many researchers working in Artificial Intelligence are deeply interested by a well-founded method for classification and data mining such as Formal Concept Analysis (see `https://conceptanalysis.wordpress.com/fca/`).

The FCA4AI Workshop Series started with ECAI 2012 (Montpellier) and the last edition was co-located with IJCAI 2021 (Montréal, Canada). The FCA4AI workshop has now a quite long history and all the proceedings are available as CEUR proceedings (see `http://ceur-ws.org/`, volumes 939, 1058, 1257, 1430, 1703, 2149, 2529, 2729, and 2972). This year, the workshop has again attracted researchers from many different countries working on actual and important topics related to FCA, showing the diversity and the richness of the relations between FCA and AI.

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis and classification. FCA allows one to build a concept lattice and a system of dependencies (implications and association rules) which can be used for many AI needs, e.g. knowledge discovery, machine learning, knowledge representation, reasoning, ontology engineering, as well as information retrieval and text processing. Recent years have been witnessing increased scientific activity around FCA, in particular a strand of work emerged that is aimed at extending the possibilities of FCA w.r.t. knowledge processing. These extensions are aimed at allowing FCA to deal with more complex data, both from the data analysis and knowledge discovery points of view. Actually these investigations provide new possibilities for AI practitioners within the framework of FCA. Accordingly, we are interested in the following issues:

- How can FCA support AI activities such as knowledge processing, i.e. knowledge discovery, knowledge representation and reasoning, machine learning (clustering, pattern and data mining), natural language processing, information retrieval...
- How can FCA be extended in order to help AI researchers to solve new and complex problems in their domains, in particular how to combine FCA with neural classifiers for improving interpretability of the output and producing valuable explanations...

The workshop is dedicated to discussion of such issues. This year it can be noticed that researchers are mostly interested in XAI and using FCA for providing explanations in Knowledge Discovery, and also in NLP, which is nowadays a very important line of investigation.

First of all we would like to thank all the authors for their contributions and all the PC members for their reviews and precious collaboration. The papers submitted to the workshop were carefully peer-reviewed by three members of the program committee. Finally, the order of the papers in the proceedings (see page 5) follows the program order (see `http://fca4ai.hse.ru/2022/`).

The Workshop Chairs

Sergei O. Kuznetsov
National Research University Higher School of Economics, Moscow, Russia

Amedeo Napoli
Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France

Sebastian Rudolph
Technische Universität Dresden, Germany

Program Committee

Mehwish Alam (AIFB Institute, FIZ KIT Karlsruhe, Germany)

Gabriela Arevalo (Universidad Austral, Buenos Aires, Argentina)

Jaume Baixeries (UPC Barcelona, Catalunya)

Alexandre Bazin (LIRMM, Université de Montpellier, France)

Karell Bertet (L3I, Université de La Rochelle, France)

Aleksey Buzmakov (HSE University Perm, Russia)

Peggy Cellier (IRISA, Université de Rennes, France)

Miguel Couceiro (LORIA, Université de Lorraine, Nancy France)

Diana Cristea (Babes-Bolyai University, Cluj-Napoca, Romania)

Mathieu D'Aquin (LORIA, Université de Lorraine, Nancy France)

Florent Domenach (Akita International University, Japan)

Elizaveta Goncharova (NRU Higher School of Economics, Moscow, Russia)

Marianne Huchard (LIRMM, Université de Montpellier, France)

Dmitry I. Ignatov (HSE University Moscow, Russia)

Dmitry Ilvovsky (HSE University Moscow, Russia)

Mehdi Kaytoue (Infologic, Lyon, France)

Francesco Kriegel (Technische Universität Dresden, Germany)

Leonard Kwuida (Bern University of Applied Sciences, Switzerland)

Florence Le Ber (ENGEES/Université de Strasbourg, France)

Nizar Messai (Université François Rabelais Tours, France)

Rokia Missaoui (UQO University Ottawa, Canada)

Sergei A. Obiedkov (NRU Higher School of Economics, Moscow, Russia)

Jean-Marc Petit (Université de Lyon, INSA Lyon, France)

Uta Priss (Ostfalia University, Wolfenbüttel, Germany)

Christian Sacarea (Babes-Bolyai University, Cluj-Napoca, Romania)

Francisco José Valverde Albacete (Universidad Carlos III de Madrid, Spain)

Renato Vimieiro (Universidade Federal de Minas Gerais, Belo Horizonte, Brazil)

Contents

1 *Invited Talk: FCA, a Step From Lattice Theory to Efficient Pattern Mining Approaches*
Karell Bertet 7

1 *Building an Intrinsically Interpretable Document Classifier with Concept Lattices*
Eric George Parakal and Sergei O. Kuznetsov 9

2 *Towards Fast Finding Optimal Short Classifiers*
Egor Dudyrev and Sergei O. Kuznetsov 23

3 *Can FCA Provide a Framework for Artificial General Intelligence?*
Francisco J. Valverde-Albacete, Carmen Peláez-Moreno, Inma P. Cabrera, Pablo Cordero, and Manuel Ojeda-Aciego 35

4 *Small Overfitting Probability in Minimization of Empirical Risk for FCA-based Machine Learning*
Dmitry V. Vinogradov 41

5 *Framework for Pareto-Optimal Multimodal Clustering*
Mikhail Bogatyrev and Dmitry Orlov 51

6 *Lazy Classification of Underground Forums Messages Using Pattern Structures*
Abdulrahim Ghazal and Sergei O. Kuznetsov 63

7 *Organizing Contexts as a Lattice of Decision Trees for Machine Reading Comprehension*
Boris Galitsky, Dmitry Ilvovsky, and Elizaveta Goncharova 75

FCA, a Step From Lattice Theory to Efficient Pattern Mining Approaches

Karell Bertet[1]

La Rochelle Université
23 avenue Albert Einstein
BP 33060 – 17031 La Rochelle, France
karell.bertet@univ-lr.fr

Abstract. In this talk, I will retrace the main mathematical steps from lattice theory to current pattern mining approaches for complex data. I will first present a survey of lattice theory, from the algebraic definition of a lattice, to that of a concept lattice, through closure systems including the exploration of fundamental bijective links between lattices, reduced contexts and bases of implicational rules. The structure of lattice or "concept lattice" is highlighted in Formal Concept Analysis (FCA). This lattice, originally defined for binary or categorical data, has proved to be useful in many fields, e.g. artificial intelligence, knowledge management, data-mining, machine learning, etc. I will then present some recent extensions of FCA to deal with non binary and complex data in order to propose efficient pattern mining approaches.

Intrinsically Interpretable Document Classification via Concept Lattices

Eric George Parakal[0000−0002−2059−1608] and Sergei O. Kuznetsov[0000−0003−3284−9001]

National Research University Higher School of Economics, Pokrovsky Blvd, 11,
Moscow 109028, Russia
{eparakal,skuznetsov}@hse.ru

Abstract. Explanations for the predictions made by Machine Learning (ML) models are best framed in terms of abstract, high-level concepts that are easily comprehensible to human beings. The use of such concepts constitutes a subfield of interpretability methods known as concept-based explanations. This work uses concept-based explanations to build an intrinsically interpretable document classifier using a combination of Formal Concept Analysis (FCA) and approaches from applied graph theory. FCA is used to formalize the vague notion of concepts in terms of the formal concepts found in the concept lattices of various document classes. The graph of the lattice covering relation helps to utilize the topological information present in the document-class concept lattices for classifying documents. Finally, the formal concepts that made the strongest contributions to the predictions of the document classifier are revealed, along with their intents; thereby making their contribution more comprehensible to human beings.

Keywords: Formal Concept Analysis · Explainable Artificial Intelligence · Natural Language Processing.

1 Introduction

The extraordinary predictive performance of contemporary Deep Neural Networks (DNNs) for a wide variety of tasks can be largely attributed to their ability to generalize the solving of a task by utilizing a vast number of neuronal parameters. However, the ability to generalize a task also leads to DNNs being more complex with respect to their design, as compared to other ML models. This complexity causes DNNs to be perceived as opaque in terms of their predictive process and consequently, leads to a lack of verifiability with regard to their predictions. Thus, despite their extraordinary predictive performance, DNNs are not adopted for use in high-risk environments such as finance, medicine and the judiciary system due to a lack of trust in their predictions.

This work details the implementation and results obtained from building an intrinsically interpretable document classifier, by utilizing the conceptual hierarchies found in the concept lattices for each document class obtained via FCA.

The work as such can then be categorized as belonging to the field of Explainable Artificial Intelligence (XAI).

With respect to the field of XAI, this work can be further categorized as belonging to the subfield of concept-based explanations. The fundamental principle of concept-based explanations is that the explanations regarding the predictions of a DNN are best comprehended if they are framed in terms of abstract, high-level concepts. This principle is akin to the process of human reasoning, whereby concepts are informally related to groupings of examples according to the similarity of their descriptions.

Earlier works concerning concept-based explanations can be categorized as post hoc interpretability methods, meaning that they explain the predictions of a DNN after it has finished training. However, more recent concept-based explanation methods belong to the category of intrinsic interpretability methods, wherein a DNN is interpretable because of its design and not due to any post training steps.

The main reason of preferring intrinsic interpretability methods to post hoc interpretability methods is that interpretability is neither an inevitable result of the discriminative power of a DNN, nor a prerequisite for it as demonstrated in [2]. Thus, it is required to ensure that the DNN learns the concepts during its training phase, instead of verifying if they have been learned post training. This is usually done by either inducing inductive biases during its training phase or by constraining the latent space of the DNN during its training phase.

This work aims to prove that Formal Concept Analysis (FCA) can be an effective method to discover the concepts that the intrinsically interpretable document classifier should learn so as to be able to explain its predictions. This is achieved by mapping the ambiguous idea of a concept to the mathematically defined notion of a formal concept, thereby allowing the creation of a conceptual hierarchy that is expressed via a concept lattice.

The intrinsically interpretable document classifier cannot directly utilize the hierarchical order information present in the concept lattices of document classes for its training, which necessitates the mapping of the formal concepts in a concept lattice to a graph topological space. The formal concepts of the concept lattices of each class of documents are treated as vertices of a directed, acyclic graph whose components are the concept lattices themselves. The vertices of the graph correspond to different formal concepts that belong to various classes, with the edges between vertices denoting the subconcept/superconcept relation. A binary classifier that is trained to detect the presence of edges among the formal concept vertices of document-class concept lattices, can be used to classify a test document by predicting if/where potential edges connect the test document vertex to the formal concept vertices found within the document-class concept lattices.

Finally, the formal concepts that have contributed the most toward the classification of a test document as belonging to a particular document class are determined by the number of edges that are predicted to connect between them and the test documents. The intents of such formal concepts reveal attributes that are more comprehensible to human beings.

The final aim of this work is to create a document classifier that is both highly interpretable and possesses relatively high classification performance. Such a model seeks to overcome the trade-off between model interpretability and performance, which asserts that predictive performance of an ML model is usually sacrificed for interpretability and vice versa; as stated in [1].

2 Related work

2.1 Concept-based explanations

The idea of concept-based explanations was first introduced in the seminal work of [8]. The work formally defines the problem of finding concepts as a interpretation function $g : E_m \to E_h$; that maps from a vector space E_m that defines the state of the DNN, to E_h a vector space in which human beings operate. The work introduces Concept Activation Vectors (CAVs) as a means of translation between E_m and E_h. A CAV is used to formally represent the notion of a concept in any layer of the DNN. A CAV is defined for a layer l of the DNN as a vector that is normal to the hyperplane separating the activations of a set of examples where the concept is present from a set of random examples. CAVs were used as a component of a method named Testing with CAVs (TCAV) which uses directional derivatives to measure the sensitivity of the predictions made by the DNN toward a concept that was learned by a CAV.

The original work while revolutionary for introducing the notion of concept-based explanations, still has some flaws. It could not inherently point toward important concepts, but could only respond to queries from the user about the significance of concepts, that must be supplied by the user themselves. The awareness and availability of well-defined concepts also affects the performance of the TCAV method, as deficiencies in either area lead to the possibility of there being a possibly infinite space of concepts from which to query. These flaws were addressed in the work of [7].

Since it is difficult to exactly define a concept, the work of [7] states three desirable properties that any concept-based explanation method must have to be comprehensible to human beings. Additionally, the work also provides the Automated Concept-based Explanation (ACE) method that can automatically identify concepts that are significant to the predictions made by a DNN. The particular example demonstrated in the work used a combination of image segmentation and image clustering to find concepts that are significant to the predictions made by a DNN trained to classify images.

Both of the previously mentioned concept-based explanation methods can be classified as post hoc interpretability methods. For reasons mentioned in Section 1, more recent concept-based explanation methods tend to belong to the class of intrinsic interpretability methods. One noteworthy example of such a concept-based explanation method is the work of [10]. This work introduced the notion of concept bottleneck models, that train on data points (x, c, y); where an input x is annotated with a human-specified concept c and a target y. Given x, the concept bottleneck model is trained to first predict the intermediate concept $\hat{c}$, which is then used to predict the target $\hat{y}$. A unique characteristic of concept bottleneck models is that they allow intervention on $\hat{c}$ by a domain expert, allowing them to edit $\hat{c}$ and propagate the corresponding changes to $\hat{y}$.

The work of [5] is also a prominent example of a concept-based explanation method that introduces a mechanism known as concept whitening. Concept whitening is implemented via a module inserted into a given layer of a DNN in order to constrain its latent space so as to represent target concepts, as well as extract them in a straightforward manner. Given a DNN classifier $f : \mathcal{X} \to \mathcal{Y}$ which has a hidden layer $\mathcal{Z}$, the classifier can be divided into the following two parts: a feature extractor $\Phi : \mathcal{X} \to \mathcal{Z}$ with parameter θ and a classifier $g : \mathcal{Z} \to \mathcal{Y}$ parameterized by ω. The goal of concept whitening is to learn Φ and g simultaneously such that

i the classifier $g(\Phi(.;\theta);\omega)$ accurately predicts the class.
ii the j^{th} dimension z_j of the latent representation $\mathbf{z}$ aligns with concept c_j

2.2 Interpretability via FCA

An example of using FCA for interpreting a DNN is the work of [14]. The main idea proposed was the generation of the architecture of a DNN based on the covering relation (the graph of the diagram) of a lattice obtained from either an antitone Galois connection (concept lattice) or a monotone Galois connection (giving rise to another type of a lattice), where every neuron can be interpreted as a concept. In the derived architecture of the DNN, the vertices of the DNN correspond to sets of similar objects with the similarity given by the set of their common attributes. The edges connecting the vertices also add to the interpretability of the DNN by denoting either concept generality (bottom-up) or conditional probability (top-bottom). This work utilizes ideas from the work of [14] in grouping similar objects as formal concept vertices based their common attributes.

3 Basic definitions

FCA is a branch of applied lattice theory that deals with deriving a concept hierarchy from a collection of objects and their attributes. The scope of the applicability of FCA to the coinciding attributes of the complete concepts extracted from ML models trained on tabular, text or sequential data can be best understood by first formally defining what a formal context, formal concept and a concept lattice are.

Definition 3.1: A **formal context** $\mathbb{K} := (G, M, I)$ consists of two sets G and M and a relation I between G and M. The elements of G are called the **objects** and the elements of M are called the **attributes** of the context. An object g that is in a relation I with an attribute m is written as gIm.

Definition 3.2: For a set $A \subseteq G$ of objects, the **derivation operator** is defined as

$$A^{'} := \{m \in M \mid gIm \text{ for all } g \in A\}$$

(the set of attributes common to the objects in A). Correspondingly, for a set B of attributes, the **derivation operator** is defined as

$$B^{'} := \{g \in G \mid gIm \text{ for all } m \in B\}$$

Definition 3.3: A **formal concept** of the context (G, M, I) is a pair (A, B) with $A \subseteq G, B \subseteq M, A^{'} = B$ and $B^{'} = A$. A is called the **extent** and B is called the **intent** of the concept (A, B). $\mathfrak{B}(G, M, I)$ denotes the set of all concepts of the context (G, M, I).

Definition 3.4: If (A_1, B_1) and (A_2, B_2) are concepts of a context, (A_1, B_1) is called a **subconcept** of (A_2, B_2), provided that $A_1 \subseteq A_2$ (which is equivalent to $B_2 \subseteq B_1$). In this case (A_2, B_2) is the **superconcept** of (A_1, B_1) and it is possible to state that $(A_1, B_1) \leq (A_2, B_2)$. The relation $\leq$ is called the **hierarchial order** (or simply **order**) of the concepts. The set of all concepts of (G, M, I) ordered in this way is denoted by $\underline{\mathfrak{B}}(G, M, I)$ and is called the **concept lattice** of the context (G, M, I).

4 Methodology

The methodology of this work consists of the following parts:

- Extracting the keywords from the training documents, to be used as attributes for building the training formal contexts for each document class.
- Building the concept lattice for each document class to be used for training the intrinsically interpretable document classifier.
- Validating the concept lattice built for each document class using a lazy FCA document classifier, which is similar to the one in [11] in order to classify documents via the concept lattice built for each document class.
- Mapping the training formal concepts of the document-class concept lattices to a graph topological space.
- Training a binary classifier to predict the potential edges between a test document vertex and the training formal concepts vertices in the document-class concept lattices, then an aggregate scoring function classifies the test document vertex according to the number of potential edges it has and to which training formal concept vertices the edges connect to.
- Revealing the training formal concepts along with their respective intents that contributed the most toward the test document vertex being predicted as belonging to a particular document class.

The relevant components of the methodology are illustrated via the toy example in Fig. 1.

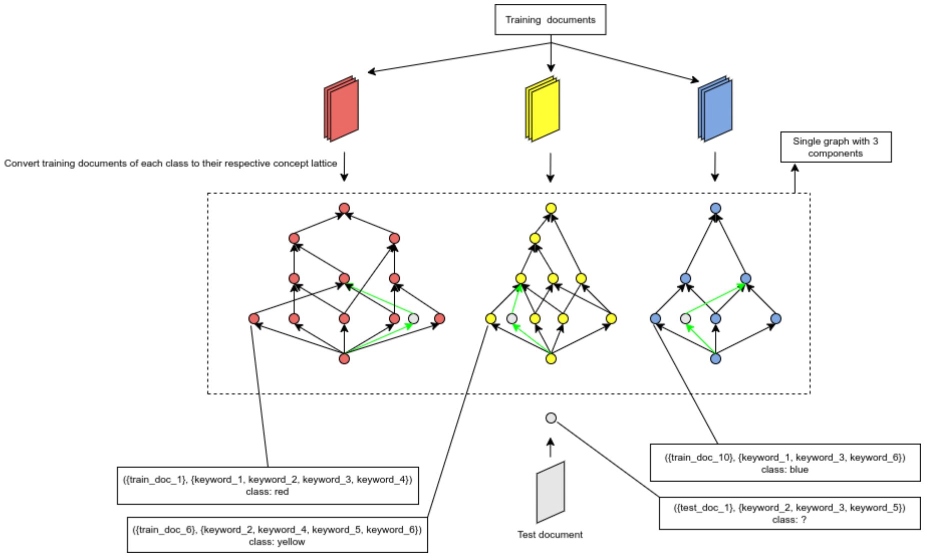

Training documents
Convert training documents of each class to their respective concept lattice
Single graph with 3 components
({train_doc_1}, {keyword_1, keyword_2, keyword_3, keyword_4})
class: red
({train_doc_6}, {keyword_2, keyword_4, keyword_5, keyword_6})
class: yellow
Test document
({train_doc_10}, {keyword_1, keyword_3, keyword_6})
class: blue
({test_doc_1}, {keyword_2, keyword_3, keyword_5})
class: ?

Fig. 1. First, documents of various classes (depicted in red, yellow and blue) are grouped together to create their respective concept lattices for training. The concept lattices are treated as components of a single, acyclic, directed, training graph whose vertices represent formal concepts. A binary classifier is first trained to predict the presence of edges (depicted in black) among the formal concept vertices of the training graph and then is used to predict if/where any potential edges (depicted in green) connect a test document vertex (depicted in grey) to the formal concept vertices of the training graph. An aggregate scoring function assigns the test document vertex to a class according to the formal concepts and the frequency of the potential edges between the test document vertex and the formal concept vertices of that particular document-class concept lattice.

4.1 Formal context creation

The classification performance of the intrinsically interpretable document classifier can only be reasonable if the formal contexts that are obtained for each class of the training data used to create the document-class concept lattices can adequately capture the underlying dependencies of the data, with minimum information loss. This can be validated by first testing the performance of a lazy FCA document classifier that uses the same document-class concept lattices that the intrinsically interpretable document classifier will use for training.

Each individual document is considered to be an object and a combined list of keywords extracted from all of the classes separately are considered to be its attributes. The keywords are extracted using the YAKE! algorithm [4], which was chosen because of its superior performance when compared to its contemporaries; as well as other conceptual scaling methods for text data. At test time, a test document is transformed into a formal context object with its attributes being the combined list of keywords extracted from all of the classes separately for all of the training documents.

4.2 Lazy FCA document classifier

The aim of the lazy FCA document classifier is to classify test documents by calculating a set of classification scores $\{S_{L_i}\}$ for each test document. Each document-class concept lattice L_i is built using the formal context obtained from the training documents of the i^{th} document class. A test document is classified as belonging to the class that has the highest classification score S_{L_i}. The classification scores of a lazy FCA document classifier can be defined in many different ways, this particular work uses the following classification score:

$$S_{L_i} = \frac{1}{|t(L_i)|} \sum_{j \in t(L_i)} |int_j \sqcap int_{test}| \times |\text{ext}(j)| \Big[|int_j \sqcap int_{test}| \geq 0.5 \times |int_{test}|\Big] \quad (1)$$

where *int* and *ext* stand for intent and extent, respectively. The S_{L_i} score first checks if the number of attributes present in the intersection of the attributes for an intent of a formal concept j and the attributes for an intent of the test document formal context object exceed a threshold that is greater than or equal to half of the total number of attributes present in the intent of j. If the aforementioned condition (written in the Iverson bracket of equation (1)) is satisfied, the number of attributes in the intersection that satisfy the condition is weighted by the extent of the formal concept j, which is used here as its interestingness measure [13].

This operation is done and the sum incremented for every formal concept j that belongs to the set $t(L_i)$. The threshold function t that is applied to a document-class concept lattice L_i is defined as $t(L_i) = \{j \in L_i \ \mid f(j) \leq \tau\}$ (τ being a hyperparameter). The function f is defined for a formal concept j as $f(j) = |\{ext_k \mid int_j \in int_k\}|$; with training documents ($\{ext_k\}$, $\{int_k\}$), called "counterexamples", belonging to any class other than the one that the formal concept j belongs to. The function f therefore, finds the number of counterexamples that a formal concept j has.

4.3 Intrinsically interpretable document classifier

The functioning of the intrinsically interpretable document classifier is explained with reference to the toy example illustrated in Fig. 1. Each of the documents belong to a class i, where $i \in$ {red, yellow, blue}. For the purpose of training the intrinsically interpretable document classifier, similar to the process in Section 4.2; three document-class concept lattices L_i are built for each class i using their respective training documents.

Referring to Fig. 1, there are three document-class concept lattices: L_{red}, L_{blue} and L_{yellow}. Each formal concept c_{ij} that belongs to the i^{th} document-class concept lattice L_i, has its own extent and intent pair ($\{ext_{ij}\}$, $\{int_{ij}\}$), where

ext_{ij} is a set of documents and int_{ij} is a set of keywords that are present in all of the documents that constitute its respective extent ext_{ij}, e.g., ({train_doc_1, train_doc_2, train_doc_3}, {keyword_1, keyword_2}).

With reference to Fig. 1, the three document-class concept lattices corresponding to the three classes can be considered as three components of a single, acyclic, directed, training graph $G_{train} = (V_{train}, E_{train})$. Each training vertex v_{ij} of G_{train} is a formal concept $c_{ij} = (\{ext_{ij}\}, \{int_{ij}\})$ that belongs to class i. A training edge $e \in E_{train}$ (depicted in black) that connects the vertices v_{ij} to each other is considered equivalent to an (implied) arc that connects two elements in a lattice diagram. Thus, the training graph G_{train} maintains the order relation, i.e., the superconcept/subconcept relation, among its training vertices v_{ij} (which represent the formal concepts c_{ij}).

A test document (depicted in grey) belonging to an unknown class i_{test}, is first converted to a test vertex v_{test} with its own extent and intent pair ($\{ext_{test}\}$, $\{int_{test}\}$) by means of the same formal context creation process described in Section 4.1. It is important to note that new keywords are not generated from the test documents so as to be used as the attributes of int_{test}, rather the keywords obtained from the training documents during the training phase are used as the attributes of int_{test}. Consequently, ext_{test} of a singular test document v_{test} is just a singleton, consisting of its own object id (file name). Thus, the extent and intent pair of a test document v_{test} is of the form ({test_doc_id}, {keyword_1, keyword_2, keyword_3, keyword_4,...}).

Using the topology of the training graph G_{train}; as well as the concepts related to its vertices v_{ij}, it is possible to predict all of the potential edges e_{test} (depicted in green) between the test vertex v_{test} and the training vertices v_{ij}. This is done in order to infer the class i_{test} of the test vertex v_{test}, based on the particular formal concepts c_{ij} (represented by the training vertices v_{ij}) that the potential edges e_{test} of v_{test} may connect to.

A DNN is trained as a binary classifier on the concatenated intents of pairs of training vertices v_{ij} in order to predict whether an edge e either exists between them or not. Then for predicting if/where the possible edges e_{test} of a test vertex v_{test} may connect to, the DNN takes as input concatenated intents of all pairs of training vertices v_{ij} and the test vertex v_{test}.

The specific architecture of the DNN used in this work is comprised of 6 fully connected layers with 192, 128, 64, 32, 16 and 1 neuron(s). Batch normalization is used after the second and fourth layers during the training phase. Dropout is used after the third and fifth layers during the training phase with respective probabilities of 0.3 and 0.2. The ReLU activation function is used after every layer. The DNN has a weight decay of 5e-4, uses a binary cross-entropy loss function and is optimized via the Adam optimization algorithm [9]; having a learning rate of 0.001 and a learning rate decay multiplicative factor of 0.1.

The aggregation function computes a score A_{L_i} for every document-class concept lattice L_i and infers the class i_{test} of the test vertex v_{test} as belonging to the document class whose concept lattice has the highest aggregate score A_{L_i}.

$$A_{L_i} = \frac{1}{|L_i|} \sum_{j \in L_i} E(v_{test}, v_{ij}) \times p(v_{ij}) \tag{2}$$

where $E(v_t, v_{ij}) = \begin{cases} 0, & \text{if no edge exists between } v_{test} \text{ and } v_{ij} \\ 1, & \text{if an edge exists between } v_{test} \text{ and } v_{ij} \end{cases}$

and $p(v_{ij}) = \frac{1}{f(v_{ij})+1}$

The function $f(v_{ij})$ counts the number of counterexamples of the formal concept v_{ij}, was previously defined in Section 4.2 and is a part of the penalty function p.

The rationale for creating such an aggregate score A_{L_i} for each document-class concept lattice, in which the function $E(v_{test}, v_{ij})$ is an essential component can be stated as follows:

1. It is not possible to accurately predict the exact training vertex v_{ij} that the predicted edge should connect to, in order to maintain the hierarchical order of formal concepts within the corresponding document-class concept lattice. This is particularly important for reasons of generating explanations as further elaborated in point 3.
2. Since a relatively simple method of representing attributes of the intents is used in this work, i.e., presence or absence of keywords, many pairs of the training vertices v_{ij} and the test vertex v_{test} can have the exact same set of intents. Thus, it is beneficial to maximize the number of edge prediction tasks for both the training (as a form of auxiliary training data) and the testing of the intrinsically interpretable document classifier (in order to bolster the aggregate scoring function A_{L_i}).
3. It is necessary to find those training vertices v_{ij} that may connect to that text vertex v_{test} in order to quantify the contribution of a formal concept v_{ij} toward classifying v_{test} as belonging to a particular class. This is illustrated by the use of the function $E(v_{test}, v_j)$ in computing the metric $Q_i(v_j)$.

4.4 Quantifying the contribution of a formal concept

The contribution of a formal concept v_j toward classifying test documents v_t as belonging to a particular class i can quantified by measuring the contribution of any predicted edges between them toward the aggregate score A_{L_i} and is defined as $Q_i(v_j) = \sum_{t \in \text{test}_i} E(v_t, v_j) \times p(v_j)$.

4.5 Dataset

The chosen dataset [3] for use in this work is a subset of a larger dataset consisting of newsgroup documents [15]. There are a total of 1000 documents, which are divided into 10 classes; with each class having 100 documents. The classes are disjoint, which means that they are no documents that belong to more than one class. After the formal context creation step specified in Section 4.1, each document has a set of attributes that denote the presence or absence of 96 unique keywords.

5 Experiment and results

The relatively small size of the dataset having only 1000 documents necessitates the need of using more data for training as compared to testing. Thus all experiments will be conducted with a 90:10 training to test split ratio. There will be 90 training documents for each class and 10 test documents for each class. The table below describes some characteristics of the training set after creating the document-class concept lattices.

Table 1. Table describing some characteristics of the document-class concept lattices obtained from the training documents of each class.

Class	No. of concepts	No. of attr.	Avg. no. of attr. per concept
Business	551	96	3.851
Entertainment	337	96	3.689
Food	185	96	3.537
Graphics	206	96	3.893
Historical	984	96	4.996
Medical	211	96	5.806
Politics	607	96	3.955
Space	552	96	4.842
Sport	297	96	4.513
Technology	986	96	4.062

5.1 Lazy FCA document classifier performance

Table 2. Table describing the classification performance of the lazy FCA document classifier described in Section 4.2

Class	Precision	Recall	F1-Score	Support
Business	1.00	0.30	0.46	10
Entertainment	1.00	0.50	0.67	10
Food	0.91	1.00	0.95	10
Graphics	0.86	0.60	0.71	10
Historical	0.35	0.60	0.44	10
Medical	0.26	0.50	0.34	10
Politics	0.89	0.80	0.84	10
Space	0.86	0.60	0.71	10
Sport	0.64	0.90	0.75	10
Technology	0.75	0.60	0.67	10
Accuracy			0.64	100
Macro avg	0.75	0.64	0.65	100
Weighted avg	0.75	0.64	0.65	100

.

As demonstrated by the table above, the lazy FCA document classifier demonstrates reasonable classification performance, across nearly all classes; with only certain classes showing poor performance. This means that the document-class concepts lattices were able to reasonably capture the underlying data dependencies in each class and map it to the conceptual hierarchy found in the concept lattices of each class. It is important to note that without this happening, there would be no reason to believe that the intrinsically interpretable document classifier would have good classification performance. The lazy FCA document classifier was run using hyperparameter value $\tau = 20$.

5.2 Intrinsically interpretable document classifier performance

Table 3. Table describing the classification performance of the intrinsically interpretable document classifier described in Section 4.3.

Class	Precision	Recall	F1-Score	Support
Business	0.86	0.60	0.71	10
Entertainment	0.78	0.70	0.74	10
Food	0.64	0.90	0.75	10
Graphics	1.00	0.60	0.75	10
Historical	0.71	0.50	0.59	10
Medical	0.67	0.60	0.63	10
Politics	0.83	1.00	0.91	10
Space	0.60	0.90	0.72	10
Sport	0.88	0.70	0.78	10
Technology	0.69	0.90	0.78	10
Accuracy			0.74	100
Macro avg	0.77	0.74	0.74	100
Weighted avg	0.77	0.74	0.74	100

As demonstrated by the table above, the intrinsically interpretable document classifier demonstrates a marked improvement in classification performance, across all classes; as compared to the lazy FCA document classifier.

5.3 Maximally contributing formal concepts

Table 4. Table describing the formal concepts that contributed the most toward classifying test documents as belonging to a particular class as described in Section 4.4

Class	Intent	$Q_i(v_j)$
Business	('asia', 'firm', 'sale')	3
Entertainment	('film', 'fly')	5
Food	('cup', 'inch', 'vegetable')	9
Graphics	('psp',)	5
Historical	('citizenship', 'french', 'war', 'world')	6
Medical	('medical', 'wvnvms')	5
Politics	('bbc', 'government', 'local')	8
Space	('earth', 'put', 'space')	6
Sport	('big', 'london', 'race', 'thing', 'world', 'year', 'york')	5
Technology	('firm', 'junk', 'virus')	5

.

The table above lists the formal concepts v_j that have the maximum value for $Q_i(v_j)$ for a document class i. The attributes of the intents of such formal concepts are intuitively logical and appear in at least half of all but one of the test documents of each class.

6 Conclusion and future work

An intrinsically interpretable document classifier with moderately high classification performance that uses properties of the graph of the covering relation of a concept lattice (lattice diagram) was proposed. The intrinsically interpretable document classifier is able to quantify which formal concepts contributed the most toward the classification of a test document. Future work can focus on using more advanced methods for representing complex data such as pattern structures [6,12] as well as using more sophisticated methods to take into account the assortativity of the graph vertices.

Acknowledgements The work of Sergei O. Kuznetsov was supported by the Russian Science Foundation under grant 22-11-00323 and performed at HSE University, Moscow, Russia.

References

1. Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion **58**, 82–115 (2020). https://doi.org/https://doi.org/10.1016/j.inffus.2019.12.012, https://www.sciencedirect.com/science/article/pii/S1566253519308103
2. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 3319–3327. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.354, https://doi.org/10.1109/CVPR.2017.354
3. Baxter, J.: (10)dataset text document classification (2020), https://www.kaggle.com/datasets/jensenbaxter/10dataset-text-document-classification
4. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A.: Yake! keyword extraction from single documents using multiple local features. Inf. Sci. **509**, 257–289 (2020). https://doi.org/10.1016/j.ins.2019.09.013, https://doi.org/10.1016/j.ins.2019.09.013
5. Chen, Z., Bei, Y., Rudin, C.: Concept whitening for interpretable image recognition. Nat. Mach. Intell. **2**(12), 772–782 (2020). https://doi.org/10.1038/s42256-020-00265-z, https://doi.org/10.1038/s42256-020-00265-z
6. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) Conceptual Structures: Broadening the Base, 9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA, July 30-August 3, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2120, pp. 129–142. Springer (2001). https://doi.org/10.1007/3-540-44583-8_10, https://doi.org/10.1007/3-540-44583-8_10
7. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 9273–9282 (2019), https://proceedings.neurips.cc/paper/2019/hash/77d2afcb31f6493e350fca61764efb9a-Abstract.html
8. Kim, B., Wattenberg, M., Gilmer, J., Cai, C.J., Wexler, J., Viégas, F.B., Sayres, R.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 2673–2682. PMLR (2018), http://proceedings.mlr.press/v80/kim18d.html
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
10. Koh, P.W., Nguyen, T., Tang, Y.S., Mussmann, S., Pierson, E., Kim, B., Liang, P.: Concept bottleneck models. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 5338–5348. PMLR (2020), http://proceedings.mlr.press/v119/koh20a.html

11. Kuznetsov, S.O.: Fitting pattern structures to knowledge discovery in big data. In: Cellier, P., Distel, F., Ganter, B. (eds.) Formal Concept Analysis, 11th International Conference, ICFCA 2013, Dresden, Germany, May 21-24, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7880, pp. 254–266. Springer (2013). https://doi.org/10.1007/978-3-642-38317-5_17, https://doi.org/10.1007/978-3-642-38317-5_17
12. Kuznetsov, S.O.: Scalable knowledge discovery in complex data with pattern structures. In: Maji, P., Ghosh, A., Murty, M.N., Ghosh, K., Pal, S.K. (eds.) Pattern Recognition and Machine Intelligence - 5th International Conference, PReMI 2013, Kolkata, India, December 10-14, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8251, pp. 30–39. Springer (2013). https://doi.org/10.1007/978-3-642-45062-4_3, https://doi.org/10.1007/978-3-642-45062-4_3
13. Kuznetsov, S.O., Makhalova, T.P.: Concept interestingness measures: a comparative study. In: Yahia, S.B., Konecny, J. (eds.) Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16, 2015. CEUR Workshop Proceedings, vol. 1466, pp. 59–72. CEUR-WS.org (2015), http://ceur-ws.org/Vol-1466/paper05.pdf
14. Kuznetsov, S.O., Makhazhanov, N., Ushakov, M.: On neural network architecture based on concept lattices. In: Kryszkiewicz, M., Appice, A., Slezak, D., Rybinski, H., Skowron, A., Ras, Z.W. (eds.) Foundations of Intelligent Systems - 23rd International Symposium, ISMIS 2017, Warsaw, Poland, June 26-29, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10352, pp. 653–663. Springer (2017). https://doi.org/10.1007/978-3-319-60438-1_64, https://doi.org/10.1007/978-3-319-60438-1_64
15. Mitchell, T.: 20 newsgroups (1999), http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

Towards Fast Finding Optimal Short Classifiers

Egor Dudyrev[0000−0002−2144−3308]
Sergei O. Kuznetsov[0000−0003−3284−9001]

HSE University, Moscow, Russia

Abstract. Studies on Explainable Artificial Intelligence show that a model should be small in order to be human understandable. The restriction on the size of a model drastically reduces the space of possible solutions. Many rule learning models still rely on greedy algorithms for generating ensembles of decision trees This paper discusses FCA-inspired mathematical and engineering techniques to efficiently find most optimal short binary classifiers, i.e., classifiers that consist of no more than three binary attributes and are optimal w.r.t. F1 score.

Keywords: Supervised Machine Learning · Explainable Artificial Intelligence · Formal Concept Analysis

1 Introduction

Studies on Explainable Artificial Intelligence show that a model should be small in order to be human understandable.

Modern learning models such as Gradient Boosting over Decision Trees [5] or Random Forest [3] are universally recognized as mainstream state-of-the-art solutions for binary classification tasks. However, such models are too complex to be analyzed and to be used in trust requiring scenarios even with the help of XAI [12] [1]. This is a reason for the rising tendency of rejecting complex black box models in favor of small explainable ones [14] [13].

Besides making models highly explainable, restricting the size of model also drastically limits the space of all possible models. Thus, one can search for a globally optimal model instead of approaching locally optimal ones. This paper considers the models operating only up to three binary attributes. In order to find an optimal model meeting these conditions, one should iterate though all combinations of up to three given binary attributes. However, such brute-force algorithm suffers from combinatorial explosion with the increasing number of attributes. This paper dwells on both mathematical and engineering techniques based on Formal Concept Analysis (FCA) [7] that make this brute-force algorithm more efficient.

The challenge of constructing simple logical models has been addressed in many areas of previous research. In the early 1960s, the work on formal logic led to the inception of logical programming and rule-learning algorithms [2], [11]. The latter – including algorithms as Skope-Rules, RuleFit [6] and more – often rely on greedy approaches to extract short rules from more complex models (such as large decision trees). Contrary to these greedy approaches resulting in locally optimal models, this paper tackles the problem of finding the most optimal model of given size.

Another possible reason to develop and to use short models is that they make good benchmark for big black box models. Indeed, if a short model outputs the same prediction quality as a big black box then there is no interest in using the latter.

Extensive research shows that a man can operate with premises having no more three plus minus one ideas in his head simultaneously [4] [9]. In this paper, due to complexity constraints, we concentrate on finding the rules with premises having no more than three attributes for two reasons. Since it is our first approach to the problem, we should try to solve its easiest version. In addition, the choice of number three is justified by the cognitive studies. Thus, short rules consisting of more than three attributes represent a specific class of explainable machine learning models.

The structure of the paper is as follows. Section 2 presents the theoretical background used throughout the paper. Section 3 describes the mathematical techniques for optimizing the algorithm by minimizing the number of required operations. Complementing this, Section 4 discusses the engineering approaches to optimize the algorithm by maximizing the computation speed. Section 5 merges all the discussed techniques together in one algorithm. And Section 6 presents the experimental results of this algorithm. Finally, Section 7 concludes the paper.

2 Theoretical Background

This subsection introduces definitions we use throughout the paper. Firstly, we provide the basic terms of Formal Concept Analysis to describe the rule models. Secondly, we describe the space of premises that contains the machine learning models discussed in the paper. Thirdly, we describe the main topics of a binary classification in the language in the FCA notation.

2.1 Formal Concept Analysis

A formal context K describes the dataset to build the model on. It is presented as a triple $K = (G, M, I)$ where G is a set of objects (rows in a dataset), M is a set of attributes (columns in a dataset), and $I \subseteq G \times M$ represents relations between objects and attributes.

Prime ($'$) operators match a subset of objects $A \subseteq G$ and a subset of attributes B such that the objects from A are “described” by the attributes from B and vice versa:

$$A' = \{m \in M \mid \forall g \in A : gIm\} \quad B' = \{g \in G \mid \forall m \in B : gIm\} \tag{1}$$

Given a subset of attributes $B \subseteq M$, the subset of objects $A = B'$ is called **extent** of B. Dually, the subset of attributes $B = A'$ is called **intent** of A.

2.2 Premises

In this subsection we define a premise as a combination of attributes of a formal context, joined by conjunction, disjuction, and negation operations. The notion of a premise is necessary for the following subsections.

Definition 1. *The* ***premise space*** $\mathbb{P}$ *is a set of all combinations of attributes* M *constructed with conjunction* $\wedge$*, disjunction* $\vee$*, and negation* $^{-}$ *operations:*

$$\begin{aligned} &\mathbb{P} \textit{ is a set s.t.} \\ &\quad 1) M \subset \mathbb{P}, \\ &\quad 2) \forall p, q \in \mathbb{P} : p \wedge q, p \vee q, \overline{p} \in \mathbb{P} \end{aligned} \tag{2}$$

Each premise $p \in \mathbb{P}$ corresponds to a subset of objects $p' \subseteq G$ called an extent of a premise. Extents of conjunction, disjunction, and negation operations are defined as follows:

$$(p \wedge q)' = p' \cap q', \quad (p \vee q)' = p' \cup q', \quad \overline{p}' = G \setminus p' \tag{3}$$

This paper is specifically interested in premises consisting of no more than three attributes. Generally, a set of premises P_i constructed from i attributes can be described in the following way:

$$\begin{aligned} &P_1 = M \cup \{\overline{m} \mid m \in M\} \\ &P_{\substack{i \in \mathbb{N} \\ i > 1}} = \bigcup_{j=1}^{\lfloor i/2 \rfloor} \{p \wedge q, p \vee q, \overline{p \wedge q}, \overline{p \vee q} \mid p \in P_j, q \in P_{i-j}\} \end{aligned} \tag{4}$$

Uniting sets of premises P_i for each natural number i we obtain the premise space $\mathbb{P}$: $\mathbb{P} = \bigcup_{i \in \mathbb{N}} P_i$. In what follows, we say that premise $p \in \mathbb{P}$ has size i if it belongs to P_i.

2.3 Binary classification

Binary classification is a task in machine learning when a model is asked to predict whether an object g belongs to a "positive" or a "negative" class given the object's description (given by a subset of attributes). The model is obtained based on the provided training context (dataset) $K = (G, M, I)$ with predefined positive $G_+ \subset G$ and negative $G_- = G \setminus G_+$ objects. The first step to constructing a good binary classifier is to find a model operating the set of attributes M that efficiently separates positive objects from G_+ and negative objects from G_-. After that, the model can be applied to a test context $K_{test} = (G_{test}, M, I_{test})$ to predict its unknown positive and negative objects.

This paper studies binary classifiers of the form "if premise $p \in \mathbb{P}$ is true then object g is predicted positive, otherwise object g is predicted negative".

The prediction quality of a premise $p \in \mathbb{P}$ on a training dataset is measured by comparing the *given* sets of positive G_+ and negative objects G_- with the sets of positive G_{p+} and negative objects G_{p-} *predicted* by a premise p. Note that the set G_{p+} is exactly the extent of the premise $p : G_{p+} = p'$.

$$\begin{aligned} TP_p &= G_+ \cap G_{p+} = G_+ \cap p' \qquad & FP_p &= G_- \cap G_{p+} = G_- \cap p' \\ FN_p &= G_+ \cap G_{p-} = G_+ \setminus p' \qquad & TN_p &= G_- \cap G_{p-} = G_- \setminus p' \end{aligned} \tag{5}$$

For the sake of brevity, let us use lowercase letters to denote the cardinalities of so-called true positives TP_p, false positives FP_p, false negatives FN_p, and true negatives TN_p. Likewise, we use y_+, y_- to denote the cardinalities of the set of positive objects G_+ and the set of negative objects G_- respectively:

$$\begin{aligned} &tp_p = |TP_p|, \quad fp_p = |FP_p|, \qquad fn_p = |FN_p|, \quad tn_p = |TN_p|, \\ &y_+ = |G_+|, \quad y_- = |G_-| \end{aligned} \tag{6}$$

One of the most widely used quality scores for binary classifications are precision ($prec$), recall (rec), and their harmonic mean called F1 score ($F1$). Their definitions are as follows:

$$prec(p) = \frac{tp_p}{|p'|}, \quad rec(p) = \frac{tp_p}{y_+}, \quad F1(p) = 2\frac{prec(p) * rec(p)}{prec(p) + rec(p)} \tag{7}$$

This paper focuses on finding the premise p^* of size not bigger that 3 having the maximal F1 score:

$$p^* = \underset{p \in \bigcup_{i=1}^{3} P_i}{\arg\max} F1(p) \tag{8}$$

The presented techniques can be easily adjusted for other quality measures.

3 Minimizing the number of comparisons

3.1 F1 score optimization

The definition of F1 score as a harmonic mean of precision and recall makes the possible optimization strategies obscure. This subsection simplifies the task of maximizing the F1 score through maximizing the number of true positives and true negatives.

Proposition 1. *F1 score $F1(p)$ is comonotonic to Jaccard score $J(p)$ (denoted by $\sim$), where the latter represents the Jaccard similarity coefficient between the set of positive objects G_+ and the set of objects predicted positive $G_{p+} = p'$:*

$$F1(p) \sim J(p) = \frac{|G_+ \cap p'|}{|G_+ \cup p'|} \tag{9}$$

Proof. Let us describe the Jaccard score in terms of true positives tp_p and true negatives tn_p:

$$J(p) = \frac{|G_+ \cap p'|}{|G_+ \cup p'|} = \frac{tp_p}{|G| - tn_p} \tag{10}$$

Now we should also express F1 score in terms of true positives tp_p and true negatives tn_p:

$$\begin{aligned} F1(p) &= 2\frac{prec(p) * rec(p)}{prec(p) + rec(p)} = \frac{2tp_p}{y_+ + |p'|} = \frac{2tp_p}{(|G| - fp_p - tn_p) + (tp_p + fp_p)} \\ &= \frac{2tp_p}{|G| + tp_p - tn_p} = \frac{2}{1 + \frac{|G| - tn_p}{tp_p}} = \frac{2}{1 + \frac{1}{J(p)}} \end{aligned} \tag{11}$$

Therefore we obtain the relation:

$$F1(p) \sim J(p)$$

Corollary 1. *Since F1 score $F1(p)$ is monotonic with respect to the Jaccard score $J(p)$ then the F1 score optimization problem can be viewed as the problem of optimizing the fraction $tp_p/(|G| - tn_p)$, i.e. maximizing the number of true positives and true negatives:*

$$\arg\max_{p\in\mathbb{P}} F1(p) = \arg\max_{p\in\mathbb{P}} J(p) = \arg\max_{p\in\mathbb{P}} \frac{tp_p}{|G| - tn_p} \tag{12}$$

3.2 Logical operations effect on Jaccard score

This subsection discusses how conjunction and disjunction operations affect the Jaccard score. That is, given the Jaccard score of two premises $p, q \in \mathbb{P}$ can we expect that premises $p \wedge q, p \vee q$ would have higher or lower Jaccard score values?

Firstly, let us express the true positives and true negatives of premises constructed by conjunction and disjunction with the true positives and true negatives of the original premises:

$$\begin{aligned} TP_{p\wedge q} &= G_+ \cap (p' \cap q') = TP_p \cap TP_q \qquad & TP_{p\vee q} &= G_+ \cap (p' \cup q') = TP_p \cup TP_q \\ TN_{p\wedge q} &= G_- \setminus (p' \cap q') = TN_p \cup TN_q \qquad & TN_{p\vee q} &= G_- \setminus (p' \cup q') = TN_p \cap TN_q \end{aligned} \tag{13}$$

Therefore, conjunction operation shrinks the set of true positives while expanding the set of true negatives. Opposite to it, disjunction operation expands the set of true positives while shrinking the set of true negatives.

Now we derive the equations for the infimum and the supremum for the cardinalities of true positives and true negatives. To do so we incorporate two notions from Equation 13. Firstly, we use set cardinality restrictions for intersection and union operations. Secondly, since the number of true positives tp is limited by the number of positive objects y_+, then for two premises $p, q \in \mathbb{P}$ if the sum of tp_p, tp_q exceeds y_+, then the corresponding true positives should have at least $tp_p + tp_q - y_+$ objects in common (analogous conclusions can be provided for the number of true negatives tn and the number of negative objects y_-).

$$\begin{aligned} \max(tp_p + tp_q - y_+, 0) &\leq \boldsymbol{tp_{p\wedge q}} \leq \min(tp_p, tp_q) \\ \max(tn_p, tn_q) &\leq \boldsymbol{tn_{p\wedge q}} \leq \min(tn_p + tn_q, y_-) \\ \max(tp_p, tp_q) &\leq \boldsymbol{tp_{p\vee q}} \leq \min(tp_p + tp_q, y_+) \\ \max(tn_p + tp_q - y_-, 0) &\leq \boldsymbol{tn_{p\vee q}} \leq \min(tn_p, tn_q) \end{aligned} \tag{14}$$

Thus, the Jaccard score bounds are:

$$
\begin{aligned}
\frac{\max(tp_p + tp_q - y_+, 0)}{|G| - \max(tn_p, tn_q)} &\leq \boldsymbol{J(p \wedge q)} \leq \frac{\min(tp_p, tp_q)}{|G| - \min(tn_p + tn_q, y_-)} \\
\frac{\max(tp_p, tp_q)}{|G| - \max(tn_p + tp_q - y_-, 0)} &\leq \boldsymbol{J(p \vee q)} \leq \frac{\min(tp_p + tp_q, y_+)}{|G| - min(tn_p, tn_q)}
\end{aligned}
\tag{15}
$$

Figure 1 visualizes the bounds on TruePositive-TrueNegative plane for an abstract formal context with 40% of objects being positive. Maximizing the Jaccard score, shown by increasing grey colour gradient, requires us to reach the upper-right corner of the plane. However, conjunction and disjunction operations over premises $p, q \in \mathbb{P}$ can only "move" the premises to the upper-left $(p \vee q)$ and lower-right $(p \wedge q)$ corners.

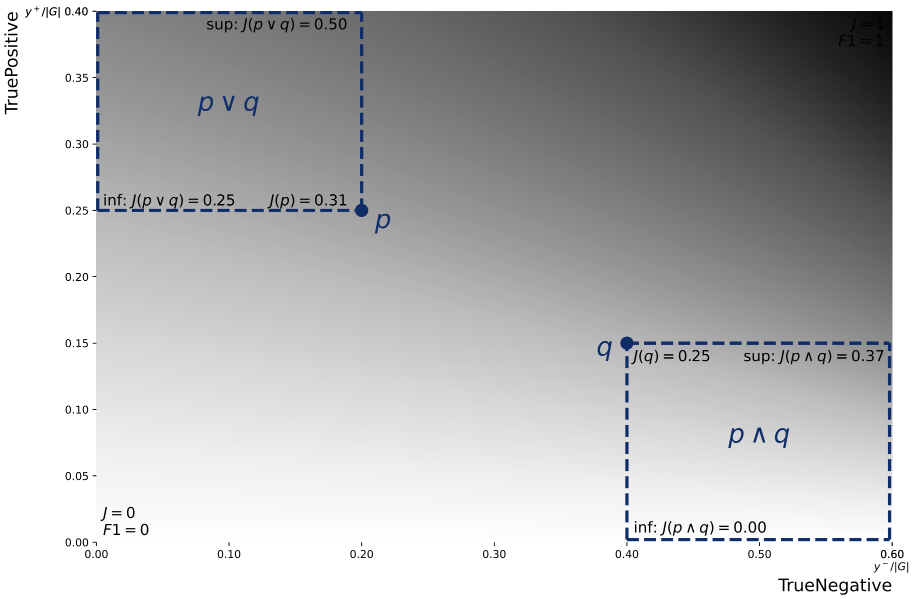


Fig. 1: The Jaccard score bounds for conjunction and disjunction operations on abstract premises p, q of abstract formal context. The gray colour intensity represents the value of the Jaccard score for a specific point on the plane.

The bounds derived for conjunction and disjunction operations are vague: they do not say whether the newly formed premise would have the higher or lower Jaccard score. The more precise estimation is only possible via intersecting the extents of premises p, q. In this case, however, one can immediately compute the resulting Jaccard score itself, rather than operating the estimations.

Proposition 2. *Given a threshold $\theta \in \mathbb{R}$ and a premise $p \in \mathbb{P}$, one can identify whether the Jaccard score of any premise $p \wedge q, p \vee q \in \mathbb{P}$ will not exceed the threshold θ using the following inequations:*

$$\begin{aligned} tp_p &\leq y_+\theta && \implies J(p \wedge q) \leq \theta, \quad \forall q \in \mathbb{P} \\ tn_p &\leq |G| - \frac{y_+}{\theta} && \implies J(p \vee q) \leq \theta, \quad \forall q \in \mathbb{P} \end{aligned} \tag{16}$$

Proof. Let us describe how the bounds in formulae 15 depend on true positives and true negatives of a premise $p \in \mathbb{P}$:

$$J(p \wedge q) \leq \frac{\min(tp_p, tp_q)}{|G| - \min(tn_p + tn_q, y_-)} \leq \frac{tp_p}{|G| - y_-} = \frac{tp_p}{y_+} \tag{17}$$

$$J(p \vee q) \leq \frac{\min(tp_p + tp_q, y_+)}{|G| - min(tn_p, tn_q)} \leq \frac{y_+}{|G| - tn_p} \tag{18}$$

Now we compare the obtained fractions with a threshold θ:

$$\frac{tp_p}{y_+} \leq \theta \Leftrightarrow \; tp_p \leq y_+\theta, \qquad \frac{y_+}{|G| - tn_p} \leq \theta \Leftrightarrow tn_p \leq |G| - \frac{y_+}{\theta} \tag{19}$$

Thus we result in the initial implications:

$$tp_p \leq y_+\theta \implies J(p \wedge q) \leq \theta, \quad \forall q \in \mathbb{P} \tag{20}$$

$$tn_p \leq |G| - \frac{y_+}{\theta} \implies J(p \vee q) \leq \theta, \quad \forall q \in \mathbb{P} \tag{21}$$

4 Engineering to maximize the speed of comparisons

Mathematical tricks help to minimize the number of comparing rules. However, the number of such rules is still high. This section concentrates on engineering tricks to make each comparison faster.

4.1 Extents and bitarrays

In this subsection we use two important facts about concept extents: (i) different premises may correspond to the same extent $\exists p, q \in \mathbb{P} : p \neq q, p' = q'$, (ii) any prediction quality measure of a premise p relies on the premise extent p' (see eq. 5).

Different premises $p, q \in \mathbb{P}, p \neq q$ may correspond to the same extent $p' = q' \subseteq G$ for many reasons. First, the premises can be logically equivalent: e.g. $(\overline{p \wedge q})' = (\overline{p} \vee \overline{q})'$ by De Morgan laws. Second, if one premise is less general than another $p' \subset q'$ then their conjunction would correspond to the extent p' and their disjunction would correspond to the extent q'. Lastly, different premises may correspond to the same extents due to the specific characteristics of the formal context K: e.g. it may occur that an extent of the conjunction of two premises $p, q \in \mathbb{P}$ would be equal to the extent of the third premise $r \in \mathbb{P} \setminus$

$\{p, q\} : (p \wedge q)' = r'$. Since the computation of prediction quality measures relies on extents of premises, then many various premises corresponding to the same extent would have the same prediction quality (on the train context K).

Thus, we propose to search for the most optimal extent instead of the most optimal premise. In order to formalize this idea, let us define the sets extents E_i:

$$\begin{gathered} E_1 = \{p' \mid \forall p \in P_1\} = \bigcup_{m \in M} \{m', (\overline{m})'\} \\ E_{\substack{i \in \mathbb{N} \\ i > 1}} = \bigcup_{j=1}^{i//2} \{a \wedge b, a \vee b \mid a \in E_j, b \in E_{i-j}\} \setminus \bigcup_{j=1}^{i-1} E_j \end{gathered} \tag{22}$$

The proposed definition of the sets of extents ensures that any extent $e \in E_i$ is generated by a premise of size at least i:

$$\forall i, j \in \mathbb{N}, j < i, e \in E_i : \exists p \in P_i : p' = e, \nexists q \in P_j : q' = e \tag{23}$$

Thus the optimization problem becomes as the following:

$$e^* = \underset{e \in \bigcup_{i=1}^{3} E_i}{\arg\max} F1(e) \tag{24}$$

where F1 score function is slightly modified to take an extent as its parameter and not the premise.

The last but not the least, an extent e, being a subset of objects G, can be represented and stored in a computer as a bit mask (a tuple of bits) of length $|G|$ where each bit represents whether the corresponding object is in the extent of not. Conjunction and disjunction operations become operations on bit masks, that are the most efficient operations performed of modern binary coded computers. So the use of extents instead of premises not only reduces the number of comparisons, it also highly accelerates each of the comparisons.

4.2 Array operations

This subsections describes the trick, that is well-known among data science practitioners, however we should cover it for the full disclosure.

Inequalities, presented in Proposition 2, allow us to skip the processing of many conjunctions $p \wedge q$ and disjunctions $p \vee q$ based on the characteristics of the initial premises $p, q \in \mathbb{P}$ and a prediction quality threshold θ. However, these characteristics are still to be computed. And to compute these numerical characteristics the most efficiently we use Numpy [10] package for Python. The package is specifically designed to work with large volume of numerical data through the use of C++ code.

5 The proposed algorithm

Here is a pseudo-code of the algorithm for finding the best k premises of size no bigger than 3:

- Step 1. Find all extents of size 1 that will not lose quality after conjunction, disjunction:
  1. Compute all extents E_1;
  2. Find the quality threshold θ as the minimal quality of the k best extents from E_1;
  3. Filter out extents from E_1 that satisfy both inequalities presented in Prop. 2;
- Step 2. Find all extents of size 2 that will not lose quality after conjunction, disjunction:
  1. Compute all extents E_2 based on filtered set E_1 while keeping the information about a pair of extents $a, b \in E_1$ and an operation $(\wedge, \vee)$ used for constructing each extent $e \in E_2$;
  2. Find the quality threshold θ as the minimal quality of the k best extents from $E_1 \cup E_2$
  3. Filter out extents from E_1, E_2 that satisfy both inequalities presented in Prop. 2
- Step 3. Find only the best extents of size 3:
  For each pair of extent $e_1, e2$ from filtered E_1, E_2
  1. If any of the extents e_1, e_2 satisfy both inequalities in Prop. 2 then proceed to the next pair; otherwise:
  2. Compute and measure the prediction quality of the conjunction $e_1 \cap e_2$;
  3. Compute and measure the prediction quality of the disjunction $e_1 \cup e_2$;
  4. Update θ if needed;
- Step 4. Reconstruct the premises corresponding to the best k extents using the kept information about extents and operations.

The time complexity of this algorithm is $O(|M|^3)$ where M is a set of attributes in a formal context K. From the asymptotic point of view, this is the same time complexity as that of the brute force algorithm to test all premises of size not bigger than three. However, the use of extents, as well as reducing the number of combinations, allows us to minimize the practical processing time of the algorithm.

6 Experiments

This section applies the proposed algorithm in practice. First, we study the statistics of the number of comparisons the algorithm has to make. Second, we roughly compare the prediction quality of short models with that of the black box model to show that there are cases where the former performs as efficient as the latter.

The algorithm is run on a real-world Myocard dataset [8] from UCI repository. The dataset contains 1700 objects and 124 attributes. The task behind Myocard dataset is to predict whether a hospital patient will have or have not a chronic heart failure based on its data. We were not successful to run the algorithm in fast time (i.e. hours and days) on bigger datasets due to the combinatorial explosion. However, we consider Myocard dataset being big enough to test the algorithm.

6.1 Number of comparisons

premise size	1	2	3
# premises	1752	3.07e+06	1.08e+10
# ext. combinations	1644	2.53e+06	1.49e+09
# ext. combs. to test	1644	2.53e+06	1.17e+09
# new extents	1644	9.44e+05	9.55e+08
# extents to keep	1644	9.44e+05	259
computation time	7.08 ms	6.55 s	1.06 h

Table 1: Statistics for algorithm iterations

Table 1 shows that the use of bounds from Prop. 2 does not filter out many extent combinations. For example, for the premise size of 3, the bounds filter only 21.5% of extents combinations (from 1.49e+09 to 1.17e+09). Although such percentage is better than nothing, it still requires to test 1.17 billions extent combinations. However, the use of bit arrays allows us to test 1.17 billions of extent combinations in only 1 hour on a laptop with 8 GB of RAM.

Legend for Table 1:

- # premises: number of premises of a given size; It is the combinatorically computed maximal amount of iterations of the algorithm;
- # ext. combinations: number of extent combinations resulting in a premise of a given size;
- # ext. combs. to test: number of extent combinations that can result in a good prediction quality (filtered by Prop. 2);
- # new extents: the number of newly generated extents resulted from testing extent combinations;
- # extents to keep: the number of extents to keep in memory. For size 1 and 2 we keep all the extents, for size 3 we keep only the extents with high prediction quality;
- computation time: the time it took to process all combinations for a given premise size.

6.2 Prediction quality of short rules

The simplicity of short rule models allows us to fully describe some of the obtained models in the paper. The following list provides the best short models (w.r.t. F1 score defined in a standard way for a binary classification method) obtained on Myocard dataset:

- Premise size 1: F1 score = 0.401426
  Premise: There is data about the use of painkillers in intensive care unit in the third day of the hospital period

- Premise size 2: F1 score = 0.448819
  Premise: (a person had a chronic heart failure) OR (has diabetes mellitus in the anamnesis)
- Premise size 3: F1 score = 0.473786
  Premise: (has data on use of opioid drugs in the intensive care unit in the third day of the hospital period) AND ((Had Chronic heart failure) OR (Age $\geq$ 66))
- XGBoost model: F1 score = 0.464000
  The model contains 100 decision trees of max depth 6
- CatBoost model: F1 score = 0.434783
  The model contains 1000 decision trees of depth 6

We can also show all the obtained short rule models on TruePositive-TrueNegative space.

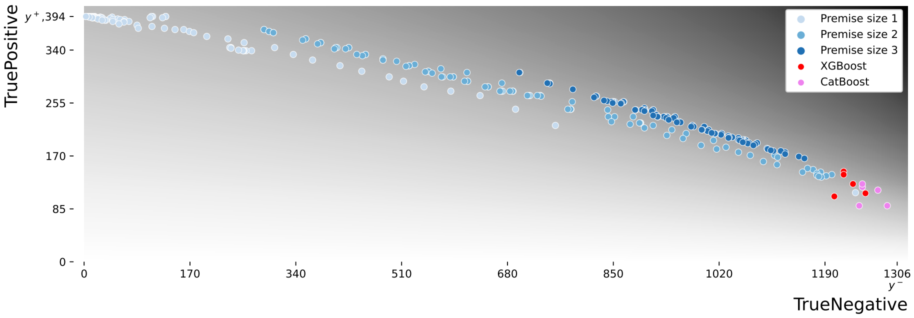


Fig. 2: Short models prediction quality on TruePositive-TrueNegative space. The prediction quality of the default XGBoost and CatBoost models are presented as a reference.

Figure 2 shows the prediction quality of the obtained short models on TruePositive-TrueNegative scale. It should be noted that the points corresponding to complex black box gradient boosting models are not far away from the ones of short models. Thus, it is not reasonable to use complex black boxes on Myocard dataset, since simple short models offer the same prediction quality.

7 Conclusion

In this paper we have presented some preliminary results on finding the most optimal rule with antecedent consisting of no more than three binary attributes. We described the F1 score optimization task in terms of true positive and true negative predictions. We computed upper and lower bounds on Jaccard coefficients for premises obtained with conjunction and disjunction operations. We also covered FCA-inspired technique of iterating over extents of premises in order to minimize the computation runtime.

In the following studies we plan to develop sharper lower and upper bounds on Jaccard score for premises constructed with conjunction and disjunction operations. We also plan to discuss other logical operations that will increase the prediction quality of rules keeping the number of used attributes the same.

Acknowledgements

The article was prepared within the framework of the Basic Research Program at HSE University, RF

References

1. Alejandro Barredo Arrieta, Natalia Daz-Rodrguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.
2. Mikhail Moiseevich Bongard. The recognition problem. Technical report, FOREIGN TECHNOLOGY DIV WRIGHT-PATTERSON AFB OHIO, 1968.
3. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.
4. Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114, 2001.
5. Jerome Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 10 2001.
6. Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The annals of applied statistics*, 2(3):916–954, 2008.
7. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis.* Springer, Berlin, 1999.
8. Sergey E Golovenkin, Jonathan Bac, Alexander Chervov, Evgeny M Mirkes, Yuliya V Orlova, Emmanuel Barillot, et al. Trajectories, bifurcations, and pseudotime in large clinical datasets: applications to myocardial infarction and diabetes data. *GigaScience*, 9(11), 11 2020. giaa128.
9. Graeme S Halford, Rosemary Baker, Julie E McCredden, and John D Bain. How many variables can humans process? *Psychological science*, 16(1):70–76, 2005.
10. Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
11. Ryszard S Michalski. Discovering classification rules using variable-valued logic system vl1. 1973.
12. Christoph Molnar. *Interpretable Machine Learning.* 2019. `https://christophm.github.io/interpretable-ml-book/`.
13. Oleg Pianykh, Egor Dudyrev, Andrew Sharp, Gleb Gusev, Sergei O. Kuznetsov, and Ilia Semenkov. Human knowledge models: Learning applied knowledge from the data. Unpublished, 2022.
14. Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019.

Can FCA Provide a Framework for AGI?

Francisco J. Valverde-Albacete[1] (✉)*, Carmen Peláez-Moreno[2], Inma P. Cabrera[3], Pablo Cordero[3], and Manuel Ojeda-Aciego[3]

[1] Depto. Teoría de Señal y Comunicaciones, Sistemas Telemáticos y Computación, Univ. Rey Juan Carlos, Madrid, Spain francisco.valverde@urjc.es

[2] Depto. Teoría de Señal y Comunicaciones, Univ. Carlos III de Madrid, Madrid, Spain, carmen@tsc.uc3m.es

[3] Depto. Matemática Aplicada, Univ. de Málaga, Málaga, Spain ipcabrera@uma.es pcordero@uma.es aciego@uma.es

Abstract. This paper is an attempt at bridging two strains of research being developed by the authors: a theory of information flows to subserve intelligence and a theory of affordances for the modelling of Embodied, Embedded, Extended and Enacted Computational Intelligence as provided by FCA. We list previous successes, present challenges, and future avenues of research that suggest themselves.

1 Towards a theory of 4E Intelligence

One of the inceptors of present-day's state-of-the-art in AI, blames the failure of old-time, symbol-based incarnation of AI on its refusal to take Biological Neural Networks (BNN) as a source of inspiration [12]. They cite three "generations" of NN in Machine Learning (ML) and suggest how each succeeded, once the hurdles of modelling specific behaviour of neural tissue and devising computational techniques to implement them were overcome.

In the Ecological Theory of Perception [8] intelligence amounts to performant cognition. At least in one stream of Cognitive Theory, cognition is *embodied, embedded, extended and enacted (4E)* [11]. These adjectives refer to qualities that the system organism-within-an-environment should have to demonstrate cognition and ultimately intelligence:

- *embodiment*, refers to having a body to behave with,
- *embedding*, to being situated within an enveloping environment,
- *extension*, to the fact that organisms can supplement their bodies through tools to extend their effect onto the environment, and
- *enaction*, to the fact that the interaction with the environment, e.g. *carrying out behaviours*, mediates in developing meaning and goals for the organism.

In this paper we speculate about machines attaining *(4E) intelligence*—as a synonym of (Natural) General Intelligence—using the techniques and tools stemming from Formal Concept Analysis (FCA), as the general framework to deal with Galois connections induced by formal contexts [7].

* Corresponding author.

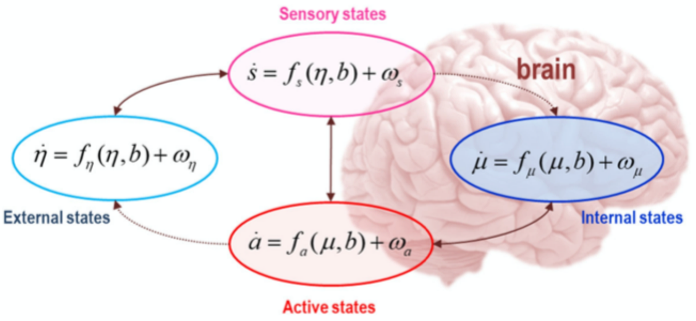

Sensory states
$\dot{s} = f_s(\eta,b)+\omega_s$
brain
$\dot{\eta} = f_\eta(\eta,b)+\omega_\eta$
External states
$\dot{\mu} = f_\mu(\mu,b)+\omega_\mu$
Internal states
$\dot{a} = f_a(\mu,b)+\omega_a$
Active states

Fig. 1: Situated Complex Dynamical System in the FE framework, with implicit dependence $b = (s, a)$ (from [6]).

From the *description* of the phenomenology of cognition and intelligence above, we move naturally onto the *design* of 4E intelligence: one way forward is to invoke the Predictive Coding or Processing hypothesis [9]. We have elsewhere summarized how this leads to a series of hypotheses about the forward and backward flows of information in BNN [15], to wit:

- BNN implement *communications at multiple spatial and temporal scales* with the goal of maintaining enaction through the *sensory-motor loop* closed by the environment. In this view, rhythms at different temporal scales subserve both the information flow along dual forwards/backwards streams as well as the prediction of the timing of events at different time-scales adapted to environment dynamics [1].
- BNNs are instances of biological *Complex Dynamical Systems (CDS)*, whose model [6] is tuple (X, S, M, A, Ω) describing an agent (that defines the "internal" space) embedded within and environment (defining the "external" space) with explicit dependences on the tuple of states $b = (s, a)$, and implicit dependence on a set of statistical parameters θ, where we have
  a) *hidden external states* $\eta \in X \subset \mathbb{R}^n$, governed by a nonlinear funtion $f_\eta : X \times S \times A \to R^n$, modeling the outer environment,
  b) *sensory states* $s \in S \subset \mathbb{R}^p$, governed by another nonlinear function $f_s : S \times S \times A \to S$, capturing the evolution of perception,
  c) *hidden internal states* $\mu \in M \subset \mathbb{R}^q$, governed by another nonlinear function $f_\mu : M \times S \times A \to M$ capturing its evolution, and
  d) *(control) actions* $a \in A \subset \mathbb{R}^m$, governed by another nonlinear function $f_a : A \times S \times A \to A$ designed to drive the dynamics,

$$\begin{aligned} \dot{\eta} &= f(\eta, b; \theta) + \omega^\eta \qquad & \dot{s} &= g(s, b; \theta) + \omega^s \\ \dot{\mu} &= f(\mu, b; \theta) + \omega^\mu \qquad & \dot{a} &= g(a, b; \theta) + \omega^a \end{aligned} \tag{1}$$

  e) Ω is a sample space from which *random fluctuations* $\omega^\eta \in \mathbb{R}^n$, $\omega^s \in \mathbb{R}^p$, $\omega^\mu \in \mathbb{R}^q$, and $\omega^a \in \mathbb{R}^m$ are drawn.

- Free-Energy (FE) models are specializations of said CDS, Bayesian models developed to reconcile the predictive coding hypothesis with neural architectures in a sensory- motor loop [9]. A basic FE model is a scale-agnostic model of the behaviour of situated agents for perception and learning [2].
- Neural processing is the communication of *entropy flows* though BNN. However, entropy being a *quantitative* property of probability measure values, it is "blind" to any particular events or "bins" of distributions, so
- BNN require *topical maps* to keep track of the *qualities* quantified by entropy flows from their information sources to their destinations.

For lack of a better name, let's call the model we envision a *4E Artificial Neural Network (4E-ANN).* Then these should be machines (embodied) that route entropy—the quantity of information—flows between sensors, actuators and the environment (embodied and embedded) and provide interpretation of these—the quality of information—in the form of connectivity or topic maps, to subserve their own goals (so as to attain extension and enaction).

2 The Affordances of FCA for a theory of 4E Intelligence: Quanta and Qualia

2.1 The Formal Qualia in Binary Formal Contexts

Ever since Wille himself cautioned against *only* reading hierarchical knowledge from FCA, there have been attempts at "other readings" from the information collected in a formal context, e.g. [5,4].

Recently, we have been calling collectively these three kinds of analyses emanating from a formal context Formal *Context* Analysis [18,17,19] but we now think about these in terms of *Formal Context Transform (FCT)*, using an analogy based on Fourier Analysis:

- In the *analysis phase* of any type of FCT the information contained in a formal context $\mathbb{K}$ is described in terms of the *formal qualia (sing. quale)*, basic abstractions Q that capture some essence of mathematical modelling, e.g. chains, antichains, partitions, etc. endowed with a partial order $\langle Q, \leq_Q \rangle$. The sets of formal qualia result in a pair of (possibly dual) complete lattices join- or meet-embedded within their order.
- In the *synthesis phase* using the join- and meet-irreducibles of some complete lattice $\langle L, \leq_L \rangle$ we synthesise a context that would return an isomorphic lattice $\mathbb{L}' \equiv L$.

The composition of both steps only allows us to maintain structure up to an isomorphism that aligns with the restrictions that the formal qualia allow, that is, they provide a focus or lens on *some* type of information included in the context, missing others:

- FCA, the analysis in terms of upper and lower bounds of the order imposed by the polars of the context on the object extents and attribute intents, producing quale *dependence or hierarchy.*
- FIA, the analysis in terms of maximal antichains of that order, evincing quale *independence* and

- FEA, the analysis in terms of the equivalence relations which are refinements of the standard congruences on objects and attributes imposed by the polars of the context, evincing quale *undistinguishability.*

But whether there might be other types of FCT is still an open issue.

2.2 The Need for Quantification in Modelling AI with FCA

A simplified (no control $u(t)$, only considers η and s states), linearized, discrete-time form of (1) can be Z-transformed as:

$$\eta_{k+1} = A_k \cdot \eta_k + \omega_k^{\eta} \qquad\qquad s_k = C_k \cdot \eta_k + \omega_k^{s} \tag{2}$$

where the instantaneous *transition and observation matrices A_k and C_k*, respectively, are to be learned[4].

Considering a set of state and observation dimensions I_η and I_s, respectively, and by virtue of the cryptomorphism between matrices with entries in an algebra, bipartite digraphs with weights in an algebra and relationships with strength in an algebra, we may consider the formal contexts $\mathbb{K}_k^{\eta} = \langle I_\eta, I_\eta, A_k \rangle$, describing the instantaneous dynamics of the CDS, and $\mathbb{K}_k^{s} = \langle I_s, I_\eta, C_k \rangle$, describing its instantaneous observation process.

We know that when the underlying algebra is an idempotent semifield $\mathcal{K}$ these formal contexts allow the definition of $\mathcal{K}$-FCA, that provides the FCA-flavour of analysis in a quantitative setting [13]. Indeed, the linear spaces associated with the input and output spaces of A_k and C_k are dually-isomorphic lattices that describe hierarchies of (quantitative) vectors that prove that standard FCA is the analogue of the Singular Value Decomposition for linear forms over vector spaces over idempotent semifields [21].

In fact, there are also different "conceptualizations" of the spaces associated with a matrix that resembles the structures needed for FIA and FEA [16,3], and also the concept of "disparity" or "discord" [10] can be modelled in the fuzzy setting—a sister theory to $\mathcal{K}$-FCA. This is a different quale to those seen above, and a certain sign that more formal qualia can be discovered.

At least all of these (and more) strains of research would need to be fused to be able to provide instantaneous lattice-like pictures of what (2) represents. Note that, after our argument, important concepts in the study of CDS, e.g. phase space, evolve in a (constrained) lattice and therefore trajectories and cycles may take strange forms, unseen so far.

3 Conclusions: Towards a Quanta-and-Qualia Theory of 4E Intelligence

In summary, we believe the generic FCA framework shows promise to help with modelling 4E intelligence, e.g. as encapsulated in a model like Friston's [6]. But in order to do so, it has to be extended:

[4] Note that this is a qualitatively different model to that of (1).

- Qualitatively, by enabling the inference of new formal qualia that cater for a better foundation of enaction and modelling the information-distinctions effectively carried out by BNNs. This is the purpose of our efforts towards better understanding FCT [20].
- Quantitatively, by enabling such qualitative inferences in the presence of quantitatively rich data, e.g. as provided by transition and observation matrices with entries in an information semifield. This is the purpose of our efforts towards understanding information semifields [23], their linear-algebraic constructions [22,16], and their relationship with Galois connections [14,3].

This Quantitative-and-Qualitative, or *Quanta-and-Qualia (QaQ)* theory of 4E-intelligences seems a promising point for 4E-NN intelligent machines, whose learning theory we have not even broached in this paper. This is left for future work.

References

1. Arnal, L.H., Giraud, A.L.: Cortical oscillations and sensory predictions. Trends in Cognitive Sciences **16**(7), 390–398 (Jul 2012)
2. Bogacz, R.: A tutorial on the free-energy framework for modelling perception and learning. Journal of Mathematical Psychology **76**, 198–211 (2017)
3. Cabrera, I., Cordero, P., Ojeda-Aciego, M.: On fuzzy relations, functional relations, and adjunctions. In: Proc. of Foundations of Computational Intelligence, FOCI (2016). `https://doi.org/10.1109/SSCI.2016.7850149`
4. Dubois, D., Prade, H.: Possibility theory and formal concept analysis: Characterizing independent sub-contexts. Fuzzy Sets and Systems **196**, 4–16 (2012)
5. Düntsch, I., Gediga, G.: Modal-style operators in qualitative data analysis. In: Proc. IEEE International Conference on Data Mining, ICDM 2002. pp. 155–162 (2002)
6. Friston, K.J., Wiese, W., Hobson, J.A.: Sentience and the Origins of Consciousness: From Cartesian Duality to Markovian Monism **22**(5), 516. `https://doi.org/10.3390/e22050516`
7. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, Heidelberg (1999)
8. Gibson, E.J., Pick, A.D.: An Ecological Approach to Perceptual Learning and Development. Oxford University Press (2000)
9. Hohwy, J.: The Predictive Processing Hypothesis. In: The Oxford Handbook of 4E Cognition. OUP (2018)
10. Madrid, N., Ojeda-Aciego, M.: Residuated structures via the f-index of inclusion. In: Computational and Mathematical Methods in Science and Engineering (2021)
11. Newen, A., de Bruin, L., Gallagher, S. (eds.): The Oxford Handbook of 4E Cognition. Oxford University Press, London (2018)
12. Sejnowski, T.J.: The unreasonable effectiveness of deep learning in artificial intelligence. Proc Natl Acad Sci USA **117**(48), 30033–30038 (Dec 2020). `https://doi.org/10.1073/pnas.1907373117`
13. Valverde-Albacete, F.J., Peláez-Moreno, C.: Extending conceptualisation modes for generalised Formal Concept Analysis. Information Sciences **181**, 1888–1909 (2011)

14. Valverde-Albacete, F.J., Peláez-Moreno, C.: K-formal concept analysis as linear algebra over idempotent semifields. Information Sciences **467**, 579–603 (2018)
15. Valverde-Albacete, F.J., Peláez-Moreno, C.: The case for quantifying artificial general intelligence with entropy semifields. In: Pap, E. (ed.) Artificial Intelligence: Theory and Applications, pp. 1–18. Springer Berlin Heidelberg (2021). `https://doi.org/10.1007/978-3-030-72711-6`, `https://link.springer.com/chapter/10.1007/978-3-030-72711-6_5`
16. Valverde-Albacete, F.J., Peláez-Moreno, C.: Four-fold formal concept analysis based on complete idempotent semifields. Mathematics MDPI **9**(2), 173 (2021). `https://doi.org/10.3390/math9020173`, `https://www.mdpi.com/2227-7390/9/2/173`
17. Valverde-Albacete, F.J., Peláez-Moreno, C., Cabrera, I.P., Cordero, P., Ojeda-Aciego, M.: A Data Analysis Application of Formal Independence Analysis. In: Concept Lattices and their Applications (CLA'18). pp. 1–12 (2018)
18. Valverde-Albacete, F.J., Peláez-Moreno, C., Cabrera, I.P., Cordero, P., Ojeda-Aciego, M.: Formal independence analysis. Communications in Computer and Information Science **853**, 596–608 (2018)
19. Valverde-Albacete, F.J., Peláez-Moreno, C., Cordero, P., Ojeda-Aciego, M.: Formal equivalence analysis. In: Proc. Conf. Internat. Fuzzy Syst. Assoc. and European Soc. Fuzzy Logic and Technol. (EUSFLAT 2019). Atlantis Press (2019)
20. Valverde-Albacete, F.J., Peláez-Moreno, C., Cordero, P., Ojeda-Aciego, M.: Formal context transforms. In preparation (2022)
21. Valverde-Albacete, F.J., Peláez-Moreno, C.: The Linear Algebra in Extended Formal Concept Analysis Over Idempotent Semifields. In: Bertet, K., Borchmann, D., Cellier, P., Ferré, S. (eds.) Formal Concept Analysis, pp. 211–227. Springer Berlin Heidelberg, Rennes (2017)
22. Valverde Albacete, F.J., Peláez-Moreno, C.: The Singular Valued Decomposition over completed idempotent semifields. Mathematics MDPI pp. 1–39 (2020)
23. Valverde-Albacete, J.F., Peláez-Moreno, C.: The Rényi Entropies Operate in Positive Semifields. Entropy **21**(8) (2019)

Small Overfitting Probability in Minimization of Empirical Risk for FCA-based Machine Learning[⋆]

Dmitry V. Vinogradov[1][0000−0001−5761−4706]

Dorodnicyn Computing Center, Federal Research Center for Computer Science and
Control, Russian Academy of Science, Moscow 119333, Russia
vinogradov.d.w@gmail.com
http://ccas.ru

Abstract. Main result of the paper provides a small upper bound on a probability of overfitting in FCA-based Machine Learning in the simplest case of Boolean algebra without counter-examples. This Machine Learning approach uses a set of randomly generated formal concepts to predict test examples. The well-known Vapnik–Chervonenkis' technique of empirical risk minimization determines a number of generated concepts. Estimations of Mixture and Stopping times of several probabilistic algorithms based on Markov chains leads to a plausible assumption on the uniform independent distribution of elements of Boolean algebra. In this case the main theorem proves that the probability of overfitting in fixed fraction of test examples tends to zero faster than exponent when the number of attributes goes to infinity.

Keywords: FCA · Markov chain · empirical risk · overfitting · Boolean algebra.

1 Introduction

Formal Concept Analysis (FCA) [1] is a popular means of data analysis in case of small samples.

However an applicability of FCA to Big Data has several obstacles:

- There will be an exponentially large number of hypotheses with respect to a size of an formal context in the worst case (for instance, the case of Boolean algebra, see next section).
- Many problems of FCA [4] belong to famous classes of NP- and $\#P$-complete problems of computational complexity.
- There is a positive probability of “accidental" concepts appearance that corresponds to the overfitting phenomenon [9].

The paper [8] introduces Markov chain approach to probabilistic generation of formal concepts to construct a Machine Learning system based on FCA

[⋆] partially supported by RFBR grant 18-29-03063mk

(FCAML). Recently FCAML-system applies the coupling Markov chain to generate a random sample of concepts. Each run of this chain terminates with probability 1. Early the system uses a monotonic Markov chain that corresponds to the famous Lazy Random Walk in the case of Boolean algebra. The paper [10] discuses Induction procedure for generalization of training examples into hypotheses about causes of the property under research with counter-examples forbidding. Finally the system predicts a target class of each test example by Analogy reasoning.

The main result of paper [10] gives a sufficient number of hypotheses to predict a target class with given level of confidence. The framework is dual to the famous one of V.N. Vapnik and A.Ya. Chervonenkis with respect to their dimension of a classifiers class.

However V.N. Vapnik and A.Ya. Chervonenkis also developed another approach to choose a good hypothesis. Best one must minimize an empirical risk (a fraction of wrongly predicted training examples). The main problem in this approach is to estimate a fraction of test examples with prediction error. We provide a partial answer on this task in the simplest case of Boolean algebra without counter-examples.

2 Background

2.1 Basic definitions and facts

Here we recall some basic definitions and facts of Machine Learning based on Formal Concept Analysis (FCAML) in the particular case of Boolean algebra. Most general situation is considered in [10]. Book [1] is the best source of information about Formal Concept Analysis itself.

The smallest **(formal) context** for n-dimensional Boolean algebra is a triple $(G, M, \neq)$, where $G = \{g_1, \ldots, g_n\}$ is a set of coatoms (**objects**), $M = \{m_1, \ldots, m_n\}$ is a set of binary **attributes**, and $\neq \subseteq G \times M$ is relation defined as $g_i \neq m_j \Leftrightarrow i \neq j$.

A **concept** of the Boolean algebra context $(G, M, \neq)$ is defined to be a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A = \{g_i \in G : \forall m_j \in B\,[i \neq j]\}$. The first component A of the concept (A, B) is called the **extent** of the concept, and the second component B is called its **intent**. It clear that the lattice of concepts coincides with Boolean algebra of all the subsets $B \subseteq M$. We consider the partial order on it dual to set inclusions.

This observation proves truth of the first obstacle of applicability of FCA to Big Data from Introduction. When formal context occupies n^2 bites only, the full description of n-dimensional Boolean algebra requires $n \cdot 2^n$ bites of memory.

Proposition 1. *For context $(G, M, \neq)$ corresponding Boolean algebra of concepts has $(\emptyset, M)$ as the bottom element and $(G, \emptyset)$ as the top element. In other words, for every the concept (A, B) the following inequalities hold:*

$$(\emptyset, M) \leq (A, B) \leq (G, \emptyset). \qquad (1)$$

Definition 1. *For a concept (A, B), $g_i \in G$, and $m_j \in M$ define*

$$CbO((A,B),g_i) = \begin{cases} (A,B) & g_i \in A \\ (A \cup \{g_i\}, B \setminus \{m_i\}) & g_i \notin A \end{cases}, \tag{2}$$

$$CbO((A,B),m_j) = \begin{cases} (A,B) & m_j \in B \\ (A \setminus \{g_j\}, B \cup \{m_j\}) & m_j \notin B \end{cases}. \tag{3}$$

We call these operations CbO because the first one is used in Close-by-One (CbO) Algorithm to generate all the formal concepts of an arbitrary context, see [3] for details.

Monotonicity property of introduced operations are summarized in the following Lemma.

Lemma 1. *Let $(G, M, \neq)$ be a context, (A, B) and (C, D) be concepts for it, $g \in G$, and $m \in M$. Then*

$$(A,B) \le (C,D) \Rightarrow CbO((A,B),g) \le CbO((C,D),g), \tag{4}$$

$$(A,B) \le (C,D) \Rightarrow CbO((A,B),m) \le CbO((C,D),m). \tag{5}$$

Initially the system used the monotonic Markov chain algorithm as a core of probabilistic approach to Machine Learning based on FCA.

```
Data: context $(G, M, \neq)$
Result: random concept $(A, B)$
$V := G \sqcup M$; $(A, B) := (\emptyset, M)$;
for $(i := 0; i < T; i++)$ do
    select random element $v \in V$;
    $(A, B) := CbO((A, B), v)$;
end
```

Algorithm 1: Monotonic Markov chain

The main difficulty with the monotonic Markov chain in general case is an absence of a good estimation on length T of its trajectory to achieve approximately stationary distribution. However the case of Boolean algebra was investigated successfully by the author. Next subsection contains the key results about it.

Then the coupling Markov chain algorithm described below appears, where there exists the natural stopping moment. Now it is a working horse for our approach.

```
Data: context $(G, M, \neq)$
Result: random concept $(A, B)$
$V := G \sqcup M$; $(A, B) := (\emptyset, M)$; $(C, D) = (G, \emptyset)$;
while $(A \neq C)$ do
    select random element $v \in V$;
    $(A, B) := CbO((A, B), v)$;
    $(C, D) := CbO((C, D), v)$;
end
```

Algorithm 2: Coupling Markov chain

The algorithm terminates when upper and lower concepts coincide. Condition on remaining of ordering between two concepts $(A, B) \leq (C, D)$ at any intermediate step of the while loop of Algorithm 2 follows from Lemma 1.

Now we represent Machine Learning based on FCA (FCAML-method) for our setting (Boolean algebra without counter-examples). See seminal paper[10] for description of the general scheme of FCAML-method.

The reader can learn the classical deterministic FCA-based approach to Machine Learning from Kuznetsov [5]. Our technique uses probabilistic Algorithm 2 for computing a random subset of concepts.

As usual, there are two sets of objects called the training $G = \{g_1, \ldots, g_n\}$ and test $X = \{o_1, \ldots, o_n\}$ samples, respectively. Set X contains examples to predict the target class (so-called test objects).

From the training samples the program generates a formal context $(G, M, \neq)$, where $M = \{m_1, \ldots, m_n\}$. After that the program applies the coupling Markov chain Algorithm 2 to generate a given number N of random concepts (A, B).

```
Data: number $N$ of concepts to generate
Result: random sample $S$ of formal concepts
while $(i < N)$ do
    Generate concept $(A, B)$ by Algorithm 2;
    $S := S \cup \{(A, B)\}$;
    $i := i + 1$;
end
```

Algorithm 3: Inductive generalization

FCAML-method replaces a time-consuming deterministic algorithm (for instance, "Close-by-One") for generation of all concepts by the probabilistic one to randomly generate the prescribed number of concepts. The goal of Markov chain approach is to select a random sample of formal concepts without computation of the (possibly exponential size) set of all the concepts.

How to select number N of concepts for the coupling Markov chain? There are 2 different approaches, both based on ideas of V.N. Vapnik and A.Ya. Chervonenkis. The approach promoted by K.V. Vorontsov is the empirical risk minimization.

```
Data: random sample $S$ of concepts
Result: empirical risk value
$G$ :=training examples; $k := 0$;
for $(g \in G)$ do
    for $(\langle A, B \rangle \in S)$ do
        if $(B \subseteq \{g\}')$ then
            $k := k + \frac{1}{n}$;
        end
        break;
    end
end
```

Algorithm 4: Calculation of empirical risk

In the Boolean algebra case it is possible select sufficiently large N to make the empirical rick equals to 0.

With permutation, we can assume without reducing generality that the first n objects were included in the training sample, and the last n objects form the test sample.

Let N of FCA Machine Learning hypotheses be generated using a coupling Markov chain from a training sample for Boolean algebra, where N chosen sufficiently large to obtain $\eta = 0$.

Stationary distribution uniformity on hypotheses allows to construct hypothesis $h_j = (A_j, B_j)$ (where $1 \leq j \leq N$) from independent Bernoulli sequence $E_j = (\epsilon_{j,1}, \ldots, \epsilon_{j,n})$ as $A_j = \{g_i : \epsilon_{j,i} = 0\}$ and $B_j = \{f_i : \epsilon_{j,i} = 1\}$.

Finally, the FCAML program predicts the target class of test examples and computes tests risk.

Consider set F of binary features $F = \{f_1, \ldots, f_n\}$. For each $1 \leq i \leq n$ introduce test example g_{n+i} with the intent $\{g_{n+i}\}' = \{f_j \in F : j \neq i\}$.

Independent Bernoulli sequence $E_j = (\epsilon_{j,1}, \ldots, \epsilon_{j,n})$ determines the corresponding hypothesis $h_j = (A_j, B_j)$, where $A_j = \{o_i : \epsilon_{j,i} = 0\}$ and $B_j = \{f_i : \epsilon_{j,i} = 1\}$.

```
Data: random sample $S = \{(A_1, B_1), \ldots, (A_N, B_N)\}$ of concepts
Result: number of erroneous predicted test examples
$X$ :=test examples; $r = 0$;
for ($g \in X$) do
    for ($j := 0; j < N; j++$) do
        if ($B_j \subseteq \{g\}'$) then
            $r := r + 1$;
        end
        break;
    end
end
```

Algorithm 5: Calculation of fraction of erroneous predictions

2.2 Approximate Uniformity of Random Subsets

Algorithm 1 in the case of Boolean algebra coincides with famous Lazy Random Walk on Boolean hypercube.

Lemma 2. *Stationary distribution π of Lazy Random Walk is uniform.*

The simplest proof of Lemma 2 uses reversibility of the corresponding Markov chain with Balance equations with respect to uniform distribution π.

Definition 2. ***Total variation*** *distance between distributions $\mu = \langle \mu_i : g_i \in G \rangle$ and $\pi = \langle \pi_i : g_i \in G \rangle$ on finite space G is defined as the half of L_1-metric, i.e. $\|\mu - \pi\|_{TV} = \frac{1}{2} \cdot \sum_{g_i \in G} |\mu_i - \pi_i|$.*

Lemma 3. $\|\mu - \pi\|_{TV} = \boldsymbol{max}_{A \subseteq G} |\mu(A) - \pi(A)|$.

This Lemma is a direct consequence of Definition 2.

Proposition 2. *For Lazy Random Walk let*

$$\mu(0) = \mathbb{P}\left[X_{t+1} = g_i \mid X_t = g_i\right] = \frac{1}{2},$$
$$\mu(e_j) = \mathbb{P}\left[X_{t+1} = (g_i \oplus e_j) \mid X_t = g_i\right] = \frac{1}{2n},$$

and $\mu = 0$ otherwise, and let π be the uniform distribution. Then

$$\left(\|\mu^{*t} - \pi\|_{TV}\right)^2 \leq \frac{1}{4} \cdot \left(e^{e^{-c}} - 1\right).$$

holds for $t \geq \frac{1}{2} \cdot n \cdot (\log n + c)$.

This proposition is analogue of result of Diakonis [2] and it was proved by the author during his research on monotonic Markov chain (Algorithm 1).

Comparison of Algorithms 1 and 2 gives assertion that the lower component of the coupling Markov chain coincides with a state of the monotonic Markov chain.

The next two propositions estimate mean length $\mathbb{E}T$ of trajectory of coupling Markov chain (Algorithm 2) and proves the strong concentration of trajectory length T around the mean $\mathbb{E}T$. The author proved them during research on coupling Markov chain (Algorithm 2).

Proposition 3. *For n-dimensional Boolean algebra*

$$\mathbb{E}T = \sum_{j=1}^{n} \frac{n}{j} \approx n \cdot \ln(n) + n \cdot \gamma + \frac{1}{2}. \tag{6}$$

Proposition 4. *For n-dimensional Boolean algebra*

$$\mathbb{P}\left[T \geq (1 + \varepsilon) \cdot n \cdot \ln(n)\right] \to 0, \tag{7}$$

when $n \to \infty$ for any $\varepsilon > 0$.

Statements of Propositions 2 and 3 and Lemma 4 imply the assertion that outputs of Algorithm 2 are approximately uniformly distributed.

Since each trajectory depends only on (pseudo-)random number generator, these outputs are independent.

But random subsets of binary attributes can be generated by Bernoulli sequences. It provides possibility of direct probabilistic computations. In the next section these considerations lead to main result of the paper.

3 Main result

Any set of hypotheses about causes of the target property can be considered as a classifier: if a test example includes at least one hypothesis, then the classifier will predict the target class positively; if none of the hypothetical reasons is embedded in a test example, then this example is classified negatively.

The method of minimizing empirical risk proposed by V.N. Vapnik and A.Ya. Chervonenkis [7] consists in choosing algorithms for which the classification of training examples contains the minimum number of errors (empirical, or observed, risk). In our case, there will always be classifiers (sets of hypotheses) whose empirical risk is zero. We restrict ourselves to these situations only. On the other hand, a risk of making a mistake in predicting test examples remains.

Following K.V. Vorontsov[11], we will randomly divide objects into two groups: training and test examples. For simplicity, let's assume that the number of objects is even, and the splitting is done in half. This assumption does not reduce generality, since the mean binomial coefficient is the largest one.

Let's denote the empirical risk by η and introduce the prediction risk as a fraction of $\theta = r/n$ incorrectly predicted test examples. We are interested in the probability of $\mathbb{P}\left[\eta = 0, \theta = \delta\right]$ when objects are evenly divided into training and test samples in half.

Since the probabilities are equal for each partition, we can assume without reducing generality that the first n objects were included in the training sample, and the last n objects form the test sample.

Let N of FCAML hypotheses be generated using a coupling Markov chain from a training sample for Boolean algebra. If the trajectories are chosen long enough, then the distribution of hypotheses will be (almost) uniform and independent. Uniformity follows from the property of fast mixing to a uniform stationary distribution, and independence follows from the independence of the Markov chain trajectories generating the FCAML hypotheses.

Denote generated hypotheses as $H = \{h_1, h_2, \ldots, h_N\}$ and form a table like

$G \mid H$	h_1	h_2	$\ldots$	h_N
g_1	0	1	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
g_n	0	0	$\ldots$	1
g_{n+1}	0	0	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$	0	$\vdots$
$g_{(1+\delta)n}$	0	0	$\ldots$	0
$g_{(1+\delta)n+1}$	1	0	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
g_{2n}	0	1	$\ldots$	1

Here 1 corresponds to inclusion of given hypothesis into a given example, i.e. the hypothesis predicts the example correctly (positively).

To reach the empirical risk $\eta = 0$ each of the first n rows must contain at least one 1.

Due to the uniform distribution and independence of hypotheses, the corresponding cells form an independent Bernoulli series with a probability of success $\frac{1}{2}$.

Lemma 4. *If hypotheses number* $N \geq (1+\sigma)\log_2 n$ *for some* $\sigma > 0$ *then* $\lim_{n\to\infty} \mathbb{P}\left[\eta = 0\right] = 1$.

Proof.

$$1 \geq \lim_{n\to\infty} \left(1-2^{-N}\right)^n = \lim_{n\to\infty} \left[\left(1-2^{-N}\right)^{2^N}\right]^{n\cdot 2^{-N}} =$$

$$= \lim_{n\to\infty} \left[e^{-1}\right]^{n\cdot 2^{-N}} \geq \lim_{n\to\infty} e^{-1/n^{\sigma}} = 1.$$

To achieve $\theta = \delta$ fraction of erroneous predictions of test examples it needs to select $\delta \cdot n$ rows of the lower half (there are $\binom{n}{\delta\cdot n}$ ways to do this) containing zeroes only, the rest rows can contains ones somewhere. Table above corresponds to the situation with choice of rows $g_{n+1}, \ldots, g_{(1+\delta)\cdot n}$.

The task is to estimate $\mathbb{P} = \binom{n}{\delta\cdot n} \cdot \left(1-2^{-N}\right)^{(2-\delta)n} \cdot \left(2^{-N}\right)^{\delta\cdot n}$ when $n \to \infty$.

Lemma 5 (Stirling's formula). $n! \approx \sqrt{2\pi n}\frac{n^n}{e^n}$ *for* $n \to \infty$.

Lemma 6 (Entropy inequality). $-\delta \cdot \ln\delta - (1-\delta)\cdot\ln(1-\delta) \leq \ln 2$.

Theorem 1. $\lim_{n\to\infty} \mathbb{P} \leq \frac{1}{\sqrt{2\pi\delta(1-\delta)}} \exp\left\{-\left(1+\sigma\right)\cdot\delta\cdot n\cdot\ln n + \ln 2\cdot n - \frac{\ln n}{2}\right\}$ *for* $n \to \infty$ *and* $N \geq (1+\sigma)\log_2 n$.

Proof. The second factor of $\mathbb{P} = \binom{n}{\delta\cdot n}\cdot\left(1-2^{-N}\right)^{(2-\delta)n}\cdot\left(2^{-N}\right)^{\delta\cdot n}$ does not exceed 1. Stirling's formula and Entropy inequality imply

$$\mathbb{P} \leq \frac{\sqrt{2\pi\cdot n}\cdot n^{\delta\cdot n}\cdot n^{(1-\delta)\cdot n}\cdot e^{\delta\cdot n}\cdot e^{(1-\delta)\cdot n}\cdot\left(2^{-N}\right)^{\delta\cdot n}}{e^n\cdot\sqrt{2\pi\delta\cdot n}\cdot\sqrt{2\pi(1-\delta)\cdot n}\cdot\delta^{\delta\cdot n}\cdot n^{\delta\cdot n}\cdot(1-\delta)^{(1-\delta)\cdot n}\cdot n^{(1-\delta)\cdot n}} \leq$$

$$\leq \frac{2^{-\delta\cdot n\cdot(1+\sigma)\cdot\log_2 n}}{\sqrt{2\pi\delta(1-\delta)\cdot n}\cdot e^{\delta\cdot n\cdot\ln\delta}\cdot e^{(1-\delta)\cdot n\cdot\ln(1-\delta)}} =$$

$$= \frac{e^{-\delta\cdot(1+\sigma)\cdot n\cdot\ln n}\cdot e^{n\cdot(-\delta\cdot\ln\delta-(1-\delta)\cdot\ln(1-\delta))}}{\sqrt{2\pi\delta(1-\delta)}\cdot\sqrt{n}} \leq \frac{e^{-\delta\cdot(1+\sigma)\cdot n\cdot\ln n}\cdot e^{\ln 2\cdot n}}{\sqrt{2\pi\delta(1-\delta)}\cdot\sqrt{n}}$$

Conclusions

Main theorem of the paper provides a small upper bound on a probability of overfitting in FCA-based Machine Learning in the simplest case of Boolean algebra without counter-examples. It states that the probability of overfitting in fixed fraction of test examples tends to zero faster than exponent when the number of attributes goes to infinity. Interesting alternative for our research is the work of T.P. Makhalova and S.O. Kuznetsov [6], where classifiers form a lattice.

Acknowledgements.

The author would like to thank his colleagues at Dorodnicyn Computing Center of Russian Academy of Science and especially his PhD student L.A. Yakimova for collaboration and helpful discussions.

The author is grateful to anonymous reviewers for improving the style of presentation.

References

1. Ganter Bernard and Wille Rudolf. *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, 1999
2. Diaconis Persi. *Group representations in probability and statistics.* IMS Lecture Notes – Monograph Series Vol. 11.– Hayward (CA): Institute of Mathematical Statistics, 1988
3. Kuznetsov S.O. A Fast Algorithm for Computing all Intersections of Objects in a Finite Semi-Lattice. *Automatic Documentation and Mathematical Linguistics.* Vol. 27. – no. 5. – 1993. – pp. 11–21
4. Kuznetsov S.O. Complexity of Learning in Concept Lattices from Positive and Negative Examples. *Discrete Applied Mathematics*, no. 142(1-3). – 2004. – pp. 111–125
5. Kuznetsov S.O. Machine Learning and Formal Concept Analysis. *Proc. 2nd International Conference on Formal Concept Analysis: Springer LNAI*, Vol. 2961. – 2004. – pp. 287–312
6. Makhalova T.P. and Kuznetsov S.O. On Overfitting of Classifiers Making a Lattice. *Proc. 14th International Conference on Formal Concept Analysis: Springer LNAI*, Vol. 10308. – 2017. – pp. 184–197
7. Vapnik V.N. *Statistical Learning Theory*, Wiley-Interscience, 1998
8. Vinogradov D.V. A Markov Chain Approach to Random Generation of Formal Concepts. *Proc. of Workshop Formal Concept Analysis Meets Information Retrieval (FCAIR 2013): CEUR Workshop Proceedings*, Vol. 977. – 2013. – p. 127–133
9. Vinogradov D.V. Accidental Formal Concepts in the Presence of Counterexamples. *Proc. of Workshop on Formal Concept Analysis for Knowledge Discovery (FCA4KD 2017): CEUR Workshop Proceedings*, Vol. 1921. – 2017. – p. 104–112
10. Vinogradov D.V. FCA-based Approach to Machine Learning. *Proc. of Workshop on Formal Concept Analysis for Artificial Intelligence (FCA4AI 2019): CEUR Workshop Proceedings*, Vol. 2529. – 2019. – p. 57–64
11. Vorontsov K.V. and Ivahnenko A. Tight Combinatorial Generalization Bounds for Threshold Conjunction Rules. *Proc. of 4th International Conference on Pattern Recognition and Machine Intelligence.* – 2011. – p. 66–73

Framework for Pareto-Optimal Multimodal Clustering

Mikhail Bogatyrev[1][0000-0001-8477-6006], Dmitry Orlov[1]

[1] Tula State University, 92 Lenin ave., Tula, Russia
okkambo@mail.ru

Abstract. The data used in Artificial Intelligence systems is often multimodal. Their representation in the form of formal contexts leads to contexts of high dimension. When constructing formal concepts and clustering on such contexts, the algorithms that are robust to the increasing dimension of contexts and capable of displaying a variety of clustering options are in demand. The modeling framework that meets these requirements is proposed. The framework uses multi-objective optimization and Evolutionary computation. The clustering results performed in the framework are compared with the known ones.

Keywords: multi-objective optimization, multimodal clustering, Pareto optimization, fact extraction.

1 Introduction

This paper is related to ongoing research in the area of applying Evolutionary computation for multi-objective optimization. It is a continuation of the research presented in [10]. In the Formal Concept Analysis (FCA), several similar tools are known [8, 9].

In FCA, the problem of multimodal clustering is solving on formal contexts and its solution depends on the dimension of the context. There are two types of solutions that are recognized as dense and non-dense clusters. Formal concepts acquired from conceptual lattice or by corresponding algorithm are dense clusters. Non-dense clusters of certain modality differ from dense ones in that their tensors may contain empty elements.

This paper describes an experimental framework for solving multi-objective optimization problems using evolutionary algorithms. Pareto-optimal solutions on formal contexts are considered here for two criteria: cluster density and volume. It is known that these criteria are contradictory. An increase in the value of one criterion is impossible without a decrease in the value of the other. The compromise solutions are needed here, which can be Pareto-optimal.

The paper is organized as follows. In the Section 2, background and related work are described. In the Section 3, proposed experimental framework is presented. The Section 4 contains some results of clustering made in the framework and compared with corresponding results acquired by the Data-Peeler algorithm. The paper ends with Conclusion section and References.

2 Background and Related Work

The background of this work consists of two areas. The first area is multi-objective optimization using evolutionary algorithms [11]. The second area is multimodal clustering in FCA.

2.1 Multi-objective Optimization with Evolutionary Algorithms

Multi-objective optimization is simultaneous optimization a number of objectives. Its specificity is manifested when the objectives conflict each other, i.e., improving the values of one objective, we worsen the values of another. This problem initiates the emergence of a set of compromise optimal solutions, commonly known as the Pareto-optimal solutions.

Pareto-optimal Multi-objective Optimization. The concept of Pareto optimality belongs to the mainstream in the domain of multi-objective optimization. Pareto optimality from the viewpoint of maximization optimization problem may be defined as follows. Let $\mathbb{S} = \{S_i\}$ is the set of solutions of multi-objective optimization problem, i =1, 2, …, n, F={f_j}, j =1, 2, …, m is the set of objective functions. Every solution is characterized by vector $\mathbf{f}_i$= $\{f_j\,(S_i)\}$. One feasible solution $\mathbf{f}_i$ is said to *dominate* another feasible solution $\mathbf{f}_k$ if and only if $f_j\,(S_i) \geq f_j\,(S_k)$ for all j =1, 2, …, m and $f_j\,(S_i) > f_j\,(S_k)$ for least one objective function f_d, $d \in \{1,2,\ldots,m\}$. A solution is said to be *Pareto optimal* if it is not dominated by any other solution. A Pareto optimal solution cannot be improved with respect to any objective function without worsening value at least one other objective function. The set of all feasible non-dominated solutions is referred to as the *Pareto optimal set*, and for a given Pareto optimal set, the corresponding objective function values in the objective space are called the *Pareto front*.

Evolutionary algorithms belong to the Evolutionary computation, the set of global optimization methods that use the *evolution of solutions*. The first known evolutionary algorithm is genetic algorithm, which realizes a probabilistic optimization technique based on the biological principles of evolution:

- encoding every solution as the string of symbols from certain alphabet (*chromosome*);
- using of a set (*population*) of solutions that evolves to one solution or to a subset of solutions corresponding to the extreme value of the certain quality criterion;
- applying various types of selecting the better solutions and (genetic) operators for manipulating solutions in the form of *mutation* and *crossover* of chromosomes.

The first two features of genetic and evolutionary algorithms determine their effectiveness in solving multi-objective optimization problems.

Encoding solutions as chromosomes allows one to simulate solutions of various problems. For example in clustering, chromosomes can directly represent clusters.

Using population of chromosomes is suitable for creating Pareto optimal solutions. There are several well-known multi-objective evolutionary algorithms (MOEAs) focused on obtaining Pareto optimal solutions: Niched Pareto Genetic Algorithm

(NPGA), Strength Pareto Evolutionary Algorithm (SPEA), Non-dominated Sorting Genetic Algorithm (NSGA), and others reviewed in [11].
The mentioned algorithms are used in solving various problems of multi-objective optimization in engineering, business and science. They are also used in data clustering [12]. In this work, the use MOEAs in multimodal clustering of formal contexts represents their new application.

2.2 Multimodal Clustering Problem in FCA

In FCA, multimodal clustering is formulated as follows.

If $R \subseteq D_1 \times D_2 \times ... \times D_n$ is a relation on data domains $D_1, D_2, ..., D_n$ then *formal context* is an n +1 set:

$$\mathbb{K} = <K_1, K_2, ..., K_n, R> \tag{1}$$

where $K_i \subseteq D_i$. *Multimodal clusters* on the context (1) are n – sets

$$\mathbb{C} = < X_1, X_2, ..., X_n > \tag{2}$$

which have the closure property [4]:

$$\forall u = (x_1, x_2, ..., x_n) \in X_1, X_2, ..., X_n, u \in R, \tag{3}$$

$\forall j = 1,2,...,n, \forall x_j \in D_j \setminus X_j < X_1, ..., X_j \cup \{x_j\}, ..., X_n >$ does not satisfy (3).

A multimodal cluster is a subset in the form of combinations of elements from different sets K_i. It is also defined as a closed n-set [3] since the closure property (3) provides its "self-sufficiency": it cannot be enlarged without violating (2).

Formal concepts on multimodal formal context are those multimodal clusters where $for\ all\ u = (x_1, x_2, ..., x_k) \in X_1, X_2, ..., X_k, u \in R$ and k is maximally possible. In other words, they are the largest possible k-dimensional hypercubes completely filled with units. The concept of the density of a multimodal cluster is introduced in FCA and formal concepts are interpreted as absolutely dense clusters [3].

There are some practical arguments in favour of studying multimodal clusters as none dense concepts additionally to studying formal concepts. Non-dense multimodal clusters can contain important information. For example, the very fact that there are certain data instances in a subset of a cluster may be an indicator of the importance of this fact. If the cluster is not dense, then to find the rest of the data that is combined with the found instance, one need to refer to the formal context. However, the search in this case will be limited by the size of the found cluster.

The Need of Multi-objective Optimization. In addition to the density of clusters, their other characteristics of *volume, modality, diversity* and *coverage* have been introduced [6]. These characteristics illustrate the quality of multimodal clustering and in some cases help to interpret the contents of clusters.

Having a set of clustering quality parameters, the multimodal clustering problem is formulated as an optimization problem in which the extremum of the criterion based on mentioned characteristics is searched for [5, 6]. In fact, some of these characteristics, for example, the volume of clusters and their density, form conflicting criteria.

Therefore, multimodal clustering on formal context may be formulated as a multi-objective optimization problem.

There are directions in FCA in which the construction of multimodal clusters is associated with the solution of optimization problems [6, 7].

3 Experimental Framework

Consider the main functional elements of the proposed framework.

3.1 Evolutionary Multi-Objective Algorithm for Multimodal Clustering

The basis of our system is evolutionary multi-objective algorithm for multimodal clustering. The algorithm uses Evolutionary computation. Evolutionary approach is applied in Pareto-optimal optimization.

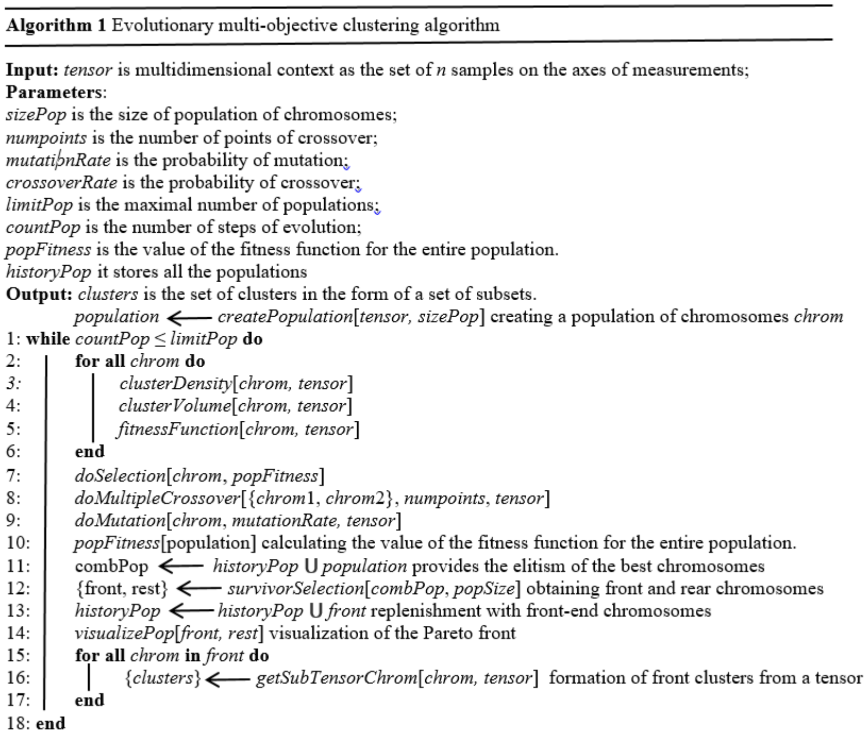
Our algorithm is based on the NSGA-II algorithm [13], which was adapted for clustering. We also expanded it with functions for visualization Pareto fronts.

The algorithm is shown on the Fig. 1. As any evolutionary algorithm, it contains functions being characteristic for genetic algorithms.

doSelection function realizes selection chromosomes according to the selection method. There are *proportional*, *random universal*, *tournament* and *truncation* selection methods realized in the algorithm. The specific selection method is picked through the user interface.

The *doMultipleCrossover* function, in addition to performing a crossover, accesses the original tensor in order to filter out the wrong chromosomes. We have also provided the crossover mode which is performed only in certain sections of chromosomes.

Encoding scheme. Encoding chromosomes is core element in evolutionary algorithms. After analyzing the several variants of chromosome encoding [12], we settled on the binary scheme organized according to the principle "one chromosome – one cluster". If formal context has modality *n* then a chromosome has *n* modal sections. In the sections, a number of gene is the number of an element of corresponding set in multimodal context. The units in the chromosome representing the cluster denote the elements included in this cluster. This binary encoding scheme is not compact because for large contexts with high modality the chromosomes will be very long. Nevertheless, in the task of clustering, it is much more convenient to work with such chromosomes than with chromosomes with more compact length. Explicit representation of clusters in the form of separate chromosomes does not require additional computations, which are necessary for other encodings. In addition, handling large binary strings is not a problem.



```
Algorithm 1 Evolutionary multi-objective clustering algorithm

Input: tensor is multidimensional context as the set of n samples on the axes of measurements;
Parameters:
sizePop is the size of population of chromosomes;
numpoints is the number of points of crossover;
mutationRate is the probability of mutation;
crossoverRate is the probability of crossover;
limitPop is the maximal number of populations;
countPop is the number of steps of evolution;
popFitness is the value of the fitness function for the entire population.
historyPop it stores all the populations
Output: clusters is the set of clusters in the form of a set of subsets.
    population $\longleftarrow$ createPopulation[tensor, sizePop] creating a population of chromosomes chrom
while countPop $\leq$ limitPop do
    for all chrom do
        clusterDensity[chrom, tensor]
        clusterVolume[chrom, tensor]
        fitnessFunction[chrom, tensor]
    end
    doSelection[chrom, popFitness]
    doMultipleCrossover[{chrom1, chrom2}, numpoints, tensor]
    doMutation[chrom, mutationRate, tensor]
    popFitness[population] calculating the value of the fitness function for the entire population.
    combPop $\longleftarrow$ historyPop $\cup$ population provides the elitism of the best chromosomes
    {front, rest} $\longleftarrow$ survivorSelection[combPop, popSize] obtaining front and rear chromosomes
    historyPop $\longleftarrow$ historyPop $\cup$ front replenishment with front-end chromosomes
    visualizePop[front, rest] visualization of the Pareto front
    for all chrom in front do
        {clusters} $\longleftarrow$ getSubTensorChrom[chrom, tensor]  formation of front clusters from a tensor
    end
end
```

Fig. 1. Evolutionary multi-objective algorithm for multimodal clustering.

The following characteristics of multimodal clusters are used in the clustering algorithm.

Cluster density and volume. For a cluster (2) its density is defined as

$$d(\mathbb{C}) = \frac{|R \cap (X_1 \times X_2 \times \ldots \times X_n)|}{|X_1| \times |X_2| \times \ldots \times |X_n|} \tag{4}$$

and volume of a cluster has the following form

$$\mathrm{v}(\mathbb{C}) = |X_1| \times |X_2| \times \ldots \times |X_n| \tag{5}$$

Cluster density and volume are contradictory criteria for cluster quality. A large and dense cluster is interesting because combinations of elements of its subsets set a property that manifests itself on a large number of elements and, possibly, means a regularity. However, often the clustered data is sparse and the existence of large and dense clusters on them is unlikely. Therefore, when selecting clusters, a trade-off between density and volume is provided by the algorithm.

Coverage and diversity. These two cluster characteristics were discussed and defined for triclustering problem in [6]. They also have generalized for multimodal clustering.

Coverage is defined as a fraction of the tuples of the context included in at least one of the multimodal clusters. This can be defined by analogy with the definition in [6]:

$$\sigma(\Omega) = \sum_{(x_1,x_2,\dots,x_n)\in R} [(x_1,x_2,\dots,x_n) \in \bigcup_{(x_1,x_2,\dots,x_n)\in\Omega} (x_1 \times x_2 \times \dots \times x_n)]/|R|, \tag{6}$$

where Ω is a set of multimodal clusters.

The data of the sets that make up the cluster modalities have different meanings. Sometimes it is important to control the coverage of the context by some subset of the cluster. In this case, in the expression (6), instead of a whole tuple $(x_1, x_2, \dots, x_n)$ one of its elements is used.

The definition of cluster diversity given in [6] is valid for multimodal clusters:

$$\tau(\Omega) = 1 - \frac{\sum_j \sum_{i<j} \gamma(\Omega_i,\Omega_j)}{\frac{|\Omega|(|\Omega|-1)}{2}}, \tag{7}$$

where $\gamma(\Omega_i,\Omega_j)$ is an intersection function which is equal to 1 if clusters Ω_i, Ω_j intersect at least one of their subsets and 0 otherwise.

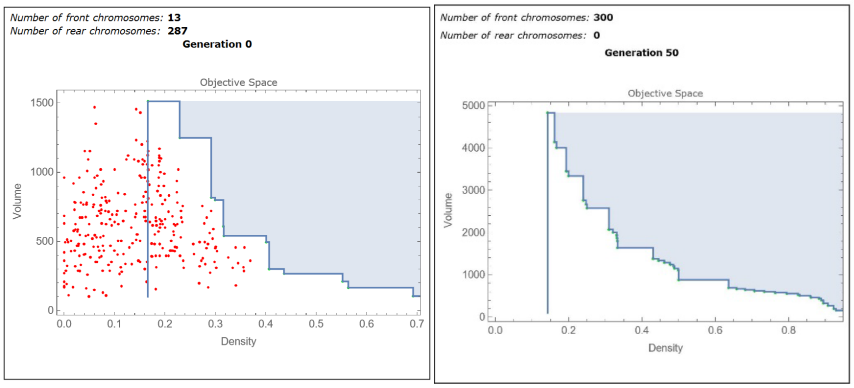
Elitist Nondominated Sorting. Evolution of solutions in the evolutionary algorithm is performed by applying genetic operators of selection, mutation and crossover to chromosome population. If the probability of mutation and crossover is high enough and the crossover is not tied to the peculiarities of chromosome encoding, then the algorithm performs random uncontrolled walks in the search space. By this way, the algorithm may explore most of the search space to find the global extremum of the fitness function. However, such walks reduce the convergence of the algorithm and, in principle, do not exclude its cycling in the regions of local extrema. Moreover, when calculating the Pareto front, random walks can lead to a "loss of the front", when the constructed Pareto front is destroyed at the next step of evolution. To exclude such phenomena we apply *elitism* [12, 13]. Elitism may be considered as an operator which preserves the better of parent and child solutions (or populations or Pareto fronts) so that a previously found better solution is never deleted. In the case of Pareto optimization, elitism is associated with dominance, and it is necessary to preserve not individual solutions, but, if possible, the entire front. In the MOEAs, elitism is realized as *nondominated sorting* [13].

3.2 Framework Realization

Considered framework is currently realized as desktop PC application with the use of some elements of Wolfram Mathematica™ environment. We use several Mathematica kernels for parallel computation. Since parallelization is natural for evolutionary algorithms, it can be realized on Mathematica kernels and helps to increase computing performance.

Java technology is also applied in the framework. Json data format is used for representing multidimensional formal contexts. We also plan to apply Java in future Web realization of the framework.

The framework uses program interface (API) Mathematica – Java and user interface with visualization Pareto fronts during evolution of computation.



a) b)

Fig. 2. The initial (a) and final (b) Pareto fronts visualizations.

Fig. 2 illustrates the evolution of solutions in the evolutionary algorithm. The area of the search space in the initial generation on the Fig. 2 a) was expanded in the final generation on the Fig. 2 b).

4 Experiments

To demonstrate functionalities of the framework we present some results of multimodal clustering on the several data sets.

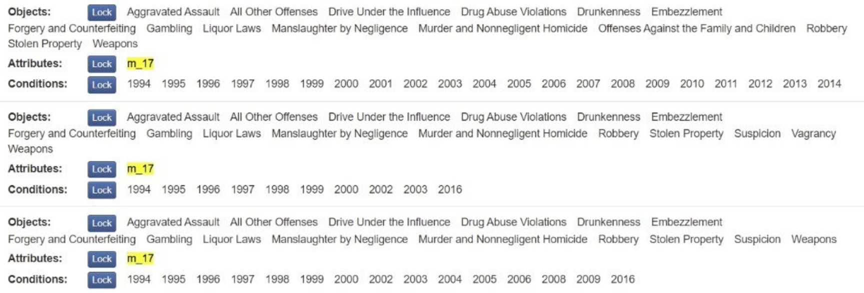
Data sets. The first data set contains data about offenses committed by juveniles [14]. We selected this data set to be able to compare our results with the results of triclustering performed by FCA algorithm of Data-Peeler [4]. This data set contains 30 objects which are the offense names, 7 attributes being the age group (m/f) which had the most amount of the certain sort offense, and 23 conditions being the years when offenses took place. Tricontext is presented as 690 incidents in the form {*offense name*, *age group*, *year*}. There are 79 formal concepts acquired from the context by Data-Peeler algorithm.

Other data sets are five tensors of dimensions from 2 to 6 generated randomly on the set {1, 2, …, 10}. They are used in experiments to study the scaling of the algorithm.

Comparison with Data-Peeler. Using the juvenile offenses data set from [14] we have the possibility to compare the results of evolutionary clustering with the results of acquiring formal concepts from this data set performed by Data-Peeler algorithm [4]. The results of the comparison are as follows.

Juvenile offenses data have a feature that manifests itself in the multimodal clustering. All formal concepts found by Data-Peeler algorithm, with the exception of concepts having empty subsets of elements, contain unique attribute values denoting the gender and age of juveniles. Most of these concepts, namely 46 ones, contain attribute m_17 denoting boys of 17 years old. On the Fig. 3 a) there are examples of three such formal concepts. They are intersecting over subsets of objects and conditions and may be united into clusters having a larger volume and containing the same information as the set of small clusters.

Our algorithm, following the principle of increasing the volume of the cluster, finds namely these, coarse-grained clusters. Example of such cluster is shown on the Fig. 3 b). It demonstrates the fact that 17 years old boys had all types of offenses at all the years of observation.

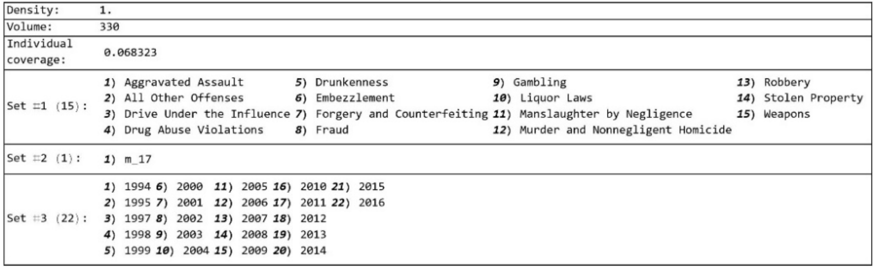


Objects: Aggravated Assault All Other Offenses Drive Under the Influence Drug Abuse Violations Drunkenness Embezzlement Forgery and Counterfeiting Gambling Liquor Laws Manslaughter by Negligence Murder and Nonnegligent Homicide Offenses Against the Family and Children Robbery Stolen Property Weapons
Attributes: m_17
Conditions: 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014

Objects: Aggravated Assault All Other Offenses Drive Under the Influence Drug Abuse Violations Drunkenness Embezzlement Forgery and Counterfeiting Gambling Liquor Laws Manslaughter by Negligence Murder and Nonnegligent Homicide Robbery Stolen Property Suspicion Vagrancy Weapons
Attributes: m_17
Conditions: 1994 1995 1996 1997 1998 1999 2000 2002 2003 2016

Objects: Aggravated Assault All Other Offenses Drive Under the Influence Drug Abuse Violations Drunkenness Embezzlement Forgery and Counterfeiting Gambling Liquor Laws Manslaughter by Negligence Murder and Nonnegligent Homicide Robbery Stolen Property Suspicion Weapons
Attributes: m_17
Conditions: 1994 1995 1996 1997 1998 1999 2000 2002 2003 2004 2005 2006 2008 2009 2016

a)

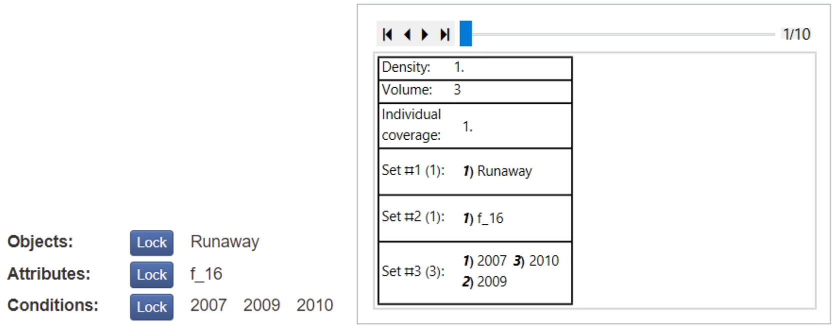


```
Density:      1.
Volume:       330
Individual
coverage:      0.068323
               1) Aggravated Assault       5) Drunkenness                9) Gambling                          13) Robbery
               2) All Other Offenses       6) Embezzlement              10) Liquor Laws                       14) Stolen Property
Set #1 (15):   3) Drive Under the Influence 7) Forgery and Counterfeiting 11) Manslaughter by Negligence      15) Weapons
               4) Drug Abuse Violations    8) Fraud                     12) Murder and Nonnegligent Homicide
Set #2 (1):    1) m_17
               1) 1994 6)  2000  11) 2005 16) 2010 21) 2015
               2) 1995 7)  2001  12) 2006 17) 2011 22) 2016
Set #3 (22):   3) 1997 8)  2002  13) 2007 18) 2012
               4) 1998 9)  2003  14) 2008 19) 2013
               5) 1999 10) 2004 15) 2009 20) 2014
```

b)

Fig. 3. a): Three formal concepts from 46 ones acquired by Data-Peeler and containing only boys of 17 years old (m_17) as attributes; b): the coarse-grained dense cluster that equivalent to these formal concepts.

Guided evolution. Our algorithm finds coarse-grained clusters of high density, which at the same time have the maximum volume. However, formal concepts of small volume containing no more than one element in one or several subsets are of particular interest. Among the formal concepts in the juvenile offenses data set there is the following one: {Runaway, f_16, (2007, 2009, 2010)}. This cluster cannot be found when the algorithm is configured to the maximum density and volume of clusters. It is necessary either to look for clusters with a minimum volume and maximum density, or to use the evolution control tools inherent in the algorithm. In our case, we supplement the principle of nondominated sorting with additional conditions for the preservation of chromosomes containing only one unit in the first and second sections. This requires running a separate experiment, in which previously found clusters may be lost, but the desired clusters of small volume are found, reflecting the unique features of the data. Fig. 4 illustrates the result of applying guided evolution on the previous example of a single formal concept.



Objects: Runaway
Attributes: f_16
Conditions: 2007 2009 2010

Density:	1.
Volume:	3
Individual coverage:	1.
Set ♯1 (1):	1) Runaway
Set ♯2 (1):	1) f_16
Set ♯3 (3):	1) 2007 3) 2010 2) 2009

Fig. 4. Formal concept and dense cluster of small volume.

Scaling the algorithm. The NSGA-II algorithm has computational complexity of $O(M*N^2)$ where M is the number of objectives and N is the population size [13]. In the problem of multimodal clustering on formal contexts, it is useful to estimate the performance of algorithms depending on the dimension of the formal context. For our algorithm, in which the dimension of the context determines the size of the chromosomes, such estimates are especially important.

Fig. 5 shows the results of testing algorithm on the randomly generated formal contexts having dimensions from 2 to 6. The population size was constant equal to 100 chromosomes, mutation probability was 0.01.

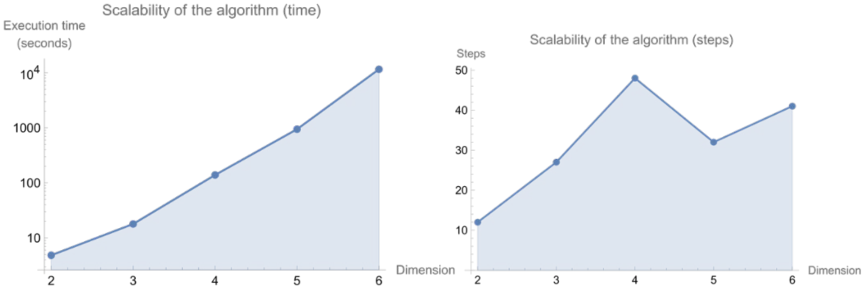


Fig. 5. The dependence of the execution time and the number of steps required to construct the Pareto front on the dimension of the formal context.

The execution time is not an objective performance characteristic of algorithms, especially for evolutionary algorithms which have a population size as a parameter. However, we use the execution time to roughly estimate the scaling of the algorithm. The dependence of the execution time on the dimension of the formal context on the Fig. 5 is approximately estimated as $\sim 10^{(k\text{-}2)}$ where k is dimension on formal context.

The number of steps required to construct the Pareto front has no explicit dependence on the dimension of the context. This corresponds to the well-known properties of evolutionary algorithms, in which the number of evolution steps randomly depends on several parameters of the algorithm.

5 Conclusion

Evolutionary algorithms have certain advantages in implementing Pareto-optimal solutions to multi-objective optimization problems. Among them, there is one important which consists in the fact that the presence of a population of solutions supported by the algorithm allows one to naturally organize the formation of the Pareto front.

In this paper, we propose two innovations, which may be interested in FCA community.

The first innovation is the application of multi-objective optimization for the construction of multimodal clusters on formal contexts.

The second innovation is the ability to control the process of building multimodal clusters through the use of an evolutionary optimization algorithms.

In the future research, we plan to perform a deeper comparison of the work of the evolutionary algorithm with other well-known FCA algorithms [3].

Also we plan to explore several other encodings in evolutionary algorithm to exclude the appearance of extra chromosomes in the population.

We hope that the modeling framework proposed here would be useful for the FCA community.

Acknowledgments. The reported study was funded by Russian Foundation of Basic Research, the research project № 20-07-00055 and RFBR and Tula Region research project № 19-47-710007. Our special thanks is for Robin Gruna for visualization of Pareto fronts.

References

1. Voutsadakis, G. Polyadic concept analysis. – Order. Vol. 19 (3). Pp. 295–304. (2002).
2. Ganter, Bernhard; Stumme, Gerd; Wille, Rudolf, eds., Formal Concept Analysis: Foundations and Applications, Lecture Notes in Artificial Intelligence, No. 3626, Springer-Verlag. Berlin (2005)
3. Dmitry V. Gnatyshak, Dmitry I. Ignatov, Sergei O. Kuznetsov: From Triadic FCA to Triclustering: Experimental Comparison of Some Triclustering Algorithms. In: Proceedings of the Tenth International Conference on Concept Lattices and Their Applications (CLA'2013), La Rochelle: Laboratory L3i, University of La Rochelle, pp. 249-260. (2013).

4. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Closed Patterns Meet *N*-ary Relations. In: ACM Trans. Knowl. Discov. Data. 3, 1, Article 3, 36 p. (2009)
5. Ignatov D. I., Semenov A., Komissarova D. V., Gnatyshak D. V. Multimodal Clustering for Community Detection, in: Formal Concept Analysis of Social Networks / Ed. by R. Missaoui, S. Kuznetsov, S. Obiedkov. Springer, P. 59-96. (2017)
6. Ignatov D. I., Gnatyshak D. V., Kuznetsov, S. O., Mirkin, B., G.: Triadic Formal Concept Analysis and triclustering: searching for optimal patterns. Mach. Learn. 101:271–302. (2015)
7. Mirkin, B. G., Kramarenko, A. V.: Approximate bicluster and tricluster boxes in the analysis of binary data. In: Rough sets, fuzzy sets, data mining and granular computing, LNCS, Vol. 6743, pp. 248–256. (2011)
8. Ignatov D. I., Egurnov D. Triclustring Toolbox, in: Supplementary Proceedings ICFCA 2019 Conference and Workshops Vol. 2378. CEUR Workshop Proceedings, 2019. P. 65-69. (2019)
9. Neznanov A., Ilvovsky D., Kuznetsov S. FCART: A New FCA-based System for Data Analysis and Knowledge Discovery, in: Contributions to the 11th International Conference on Formal Concept Analysis. Dresden : Qucoza,. P. 31-44. (2013)
10. Mikhail Bogatyrev, Dmitry Orlov and Tatyana Shestaka. Multimodal Clustering with Evolutionary Algorithms. Proc. of the 9th Int. Workshop "What can FCA do for Artificial Intelligence?" co-located with the 30th Int. Joint Conference on Artificial Intelligence (IJCAI 2021). Montréal, Québec, Canada, August 21, 2021. CEUR Proceedings, Vol. 2972. Pp. 71-86. (2021)
11. Li, K., Wang, R., Zhang, T., et al. Evolutionary many-objective optimization: A comparative study of the state-of-the-art. IEEE Access, vol. 6, pp. 26194-26214. (2018)
12. Hruschka E., Campello R., Freitas A., de Carballo A.: A Survey of Evolutionary Algorithms for Clustering. IEEE Transactions on Evolutionary Computation. V. 39. P. 133–155. (2009)
13. Deb, A. Pratap, Agrawal, S. and Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation, vol. 6, pp. 182–197. (2002)
14. Kis, L.L., Sacarea, C., Troanca, D.: FCA Tools Bundle - a Tool that Enables Dyadic and Triadic Conceptual Navigation. In: Proceedings of the 5th International Workshop “What can FCA do for Artificial Intelligence?” Co-located with the European Conference on Artificial Intelligence, The Hague, The Netherlands, 30 August 2016, pp. 42–50 (2016)

Lazy Classification of Underground Forums Messages Using Pattern Structures

Abdulrahim Ghazal[1] and Sergei O. Kuznetsov[2]

[1] National Research University Higher School of Economics, Pokrovsky boulevard,
11, 109028, Russia, Moscow
agazal@hse.ru

[2] National Research University Higher School of Economics, Pokrovsky boulevard,
11, 109028, Russia, Moscow
skuznetsov@hse.ru

Abstract. Underground forums are monitored platforms where hackers announce attacks and tools to carry on attacks on businesses or organizations. In this paper, we will experiment on assessing the risk of a dataset of these messages, using pattern structures and a lazy classification scheme, with some introduced complexity-reducing elements and natural language analysis techniques. The results show promising application for this method for this problem, and serve as an introductory step for deeper investigation.

Keywords: Formal concept analysis (FCA) · Threat intelligence · Underground forums · Pattern structures.

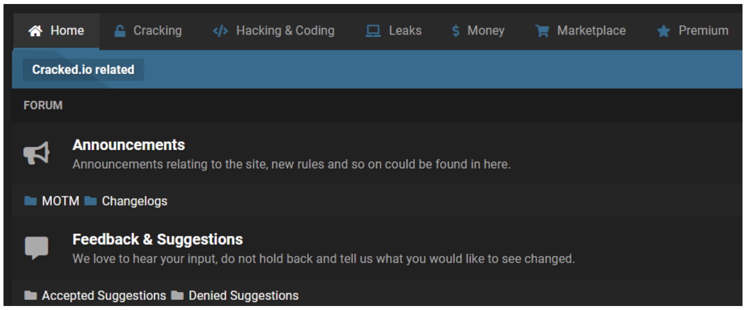
1 Introduction

1.1 Threat Intelligence

Threat Intelligence constitutes a very critical part of the cybersecurity world nowadays, with the intensifying cases of cyber attacks that are costing millions of dollars to many industries and governments [1], and improving every day in quantity and quality.

The practice of collecting information about attacks or offers to attack a target from various sources has attracted a lot of research and business attention. This is done by monitoring online forums and instant messaging services, where threat actors post to let other members know that they have some unauthorized access to a network, database or that they made a new tool that can help to achieve the above goals. Most of the serious criminal activity is done for money [2], but sometimes, it has other motives (political statements [3], sabotage or even corporate espionage).

The forums consist of sub-forums and sections that deal with different topics and other administrative sections to control the announcement, sales, and membership processes. In addition to that, many forums have a direct marketplace section, where more strict rules about content are enforced and make these



Announcements
Announcements relating to the site, new rules and so on could be found in here.

Feedback & Suggestions
We love to hear your input, do not hold back and tell us what you would like to see changed.

Fig. 1. An example of forum sections (partial structure).

sections designated for sale/buying of illegal products, ranging from cracked software to confidential databases of entities.

The most prestigious forums also have an "escrow" service that works like a mediator between buyers and sellers on the forum and a type of guarantee of impartial control over the interaction. Many forums also have tiers of membership (VIP, Premium, Golden, etc.). Some forums have no free membership tiers, meaning that one have to pay to access their content. The content posted in paid sections or forums is supposed to have more credibility and be written by more interesting members, depending on the prices, which range from 10$ up to 300$. Most transactions are paid with cryptocurrency.

1.2 Threat Intelligence Workflow

Detecting threats is usually performed with the help of human analysts. The analysts' workflow has three main phases: discovery, reaction and analysis. The most time consuming and difficult to do is the first phase, where the analyst has to browse through thousands of messages posted daily to find credible threats to report. This is even harder when there is a constant stream of spam messages flooding the forums.

This work will focus on helping the human analysts in the discovery phase, using textual analysis of the messages, and learning approaches to help filter out the irrelevant messages, and tag all relevant messages with the right threat category to further ease the next two phases of the workflow.

Now to avoid confusion, we need to define some terms as they are mentioned in the system vs. how they are usually mentioned in the real world. Table 1 gives the terminology.

We need a system that can detect threats without capturing many false positive examples, or missing a true positive case, which carries a high business cost. Another constraint on the needed system is to be time sensitive, as many threats are very volatile, meaning that the threat actor will post about the sale of access for example, in several minutes or hours, some other criminal entity (or in some cases the victims) will contact him about the sale, pay him and request removing the sale announcement. The last important feature of the system is explainability, as human analysts need to understand the reasoning behind a classification result. This will be achieved by building a learning-based system.

Table 1. Terminology Disambiguation.

Usual terminology	Our Terminology	Meaning
Post	Thread	The group of messages that are posted under one topic.
Comment, message, reply	Message	The one item posted in a thread.
-	Last post	The Html tags that contain all the relevant information for extracting a message.
Author, Hacker, original poster	Threat Actor	The person who wrote the message.
Title, Headline	Topic	The thread title.

The rest of the paper is organized as follows: In Section 2 we recall basic definitions in formal concept analysis and pattern structures. In Section 3 we describe the lazy classification method using pattern structures. Section 4 describes the experimental setting. In Section 5, we discuss the preliminary results of applying the lazy classification to underground forum messages. We conclude the work in section 6.

2 Formal Concept Analysis

2.1 Main Definitions

Formal Concept Analysis (FCA) as defined in [4] is a mathematical theory that is based on concepts and conceptual hierarchy. It is applied for knowledge discovery and data analysis [5, 6].

Let G be a set of objects, M a set of attributes or descriptions of these objects, and $I \subseteq G \times M$ a binary relation between G and M. We call the triple (G,M,I) a formal context. If $g \in G$ has the attribute $m \in M$, then $(g, m) \in I$. We then define the derivation operators $(.)^{'}$ on $A \subseteq G$ and $B \subseteq M$:

$$A^{'} = \{m \in M \mid \forall g \in A : gIm\} \tag{1}$$

$$B^{'} = \{g \in G \mid \forall m \in B : gIm\} \tag{2}$$

We call a pair (A,B) such that $A^{'} = B$ and $B^{'} = A$, a formal concept, and A is called its extent and B is its intent. A partial order $\leq$ is defined on the set of concepts: (A,B) $\leq$ (C,D) iff $A \subseteq C$ and $B \subseteq D$. In this case, (A,B) is called the subconcept and (C,D) a superconcept. This partial order gives rise to a complete lattice on the set of all formal concepts. We call this the concept lattice ζ of the formal context (G,M,I).

The hierarchical structure of concept lattices can be applied at mining association rules [5, 7] ontology design [8, 9], and recommendation systems, because of the ability to explain the rationale behind the recommended item [10].

There is a large focus in FCA domain on building concept lattices and extracting the concepts from a given formal context in an efficient manner, for it to become more applicable in the real world [11]. In [12] researchers compared the performance of several algorithms to build concept lattices and gave insights on which algorithm would be the one to choose depending on the data. There is also a number of works to discuss other applications of FCA in learning [13, 14].

2.2 Pattern Structures

To increase the applications of Formal Concept Analysis, it had to be able to represent more complex data structures, like graphs or non-binary data. This need led to the development of pattern structures [15].

Let G be a set of objects and $(D, \sqcap)$ be a meet-semi-lattice of possible object descriptions or patterns (for standard FCA, it would be the powerset of attribute set) with the similarity operator $\sqcap$. Elements of D are ordered by a subsumption relation $\sqsubseteq$ such that a,$b \in D$, then one has $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$. We also define $\delta : G \rightarrow D$ as a mapping between objects and their attributes. We call $(G, \underline{D}, \delta)$ where $\underline{D} = (D, \sqcap)$ a pattern structure. We can define the operators $(\cdot)^{\diamond}$ on $A \subseteq G$ and $d \in (D, \sqcap)$ making Galois connection between the powerset of objects and ordered set of descriptions:

$$A^{\diamond} = \sqcap_{g \in A} \delta(g) \tag{3}$$

$$d^{\diamond} = \{g \in G \mid d \sqsubseteq \delta(g)\} \tag{4}$$

These operators will give us back the maximal set of patterns shared by the objects in A and the maximal set of objects that share the description d, respectively.

A pair (A, d), $A \in G$ and $d \in (D, \sqcap)$ that satisfies $A^{\diamond} = d$ and $d^{\diamond} = A$ is called a *pattern concept*, where A is called the *extent* and d is called the *pattern intent* of (A, d).

A partial order $\leq$ is defined on the set of concepts: $(A, d_1) \leq (B, d_2)$ iff $A \subseteq B$ (or, equivalently, $d_2 \sqsubseteq d_1$). This partial order forms a complete lattice on the set of all pattern concepts. We call this the pattern concept lattice of the pattern structure $(G, \underline{D}, \delta)$.

Pattern structures are helpful in applications with complex data, like graphs, many-valued attributes [5], intervals or interval vectors [6].

For classification tasks we do not need to extract the full hidden knowledge from a dataset in terms of implications, hypotheses or association rules, but a so-called lazy classification can be applied [16, 17].

2.3 Lazy Classification with Pattern Structures

In classification problems we have a target attribute, which, in the simplest case of two classes, has two values, denoted by + and −. By G_+ we denote the set of objects that have the target attribute (positive examples) and by G_- we denote the set of objects that do not have the target attribute (negative examples), so that $G_+ \cap G_- = \emptyset$. Elements of G that do not belong to any of these subsets are called unclassified examples G_τ.

A version of the lazy classification method [16, 17] is described in Algorithm 1.

Algorithm 1: Lazy Classification with Pattern Structures

```
Requires: pattern structure $(G, \underline{D}, \delta)$, test example $g_t \in G_\tau$ with description $\delta(g_t)$,
parameter $0 \leq \alpha \leq 1$.
for $g \in G_+ \cup G_-$ :
compute sim $= \delta(g) \sqcap \delta(g_t)$
extsim $=$ (sim)$^\diamond$
        if $\alpha$% of objects in extsim have target attribute, classify $g$ positive
        if $\alpha$% of objects in extsim do not have target attribute, classify $g$ negative
classify undetermined (the algorithm terminates without classification).
```

This algorithm takes $O \mid G \mid \cdot (p(\sqcap + \mid G \mid p(\sqsubseteq)))$ time, where $p(\sqcap)$, $p(\sqsubseteq)$ are times for computing $\sqcap$, $\sqsubseteq$, respectively.

3 Experiments

The examples dataset used in the following experiments is composed of underground forum messages as the objects and features extracted from these messages as attributes. After building the concept lattice from this dataset, we construct another dataset for testing the classification. Both datasets contain positive and negative examples. The target attribute is a simple flag stating whether the message is a real threat or not.

The procedure realizing the lazy classification scheme is as follows: we go through the objects of the formal context and check the extension of the intersection of the test object and the concept object, if all objects of this extension have the attribute value, then the test object is classified positively, if all of the objects of the extension do not have the attribute value, the test object is classified negatively, otherwise, the model cannot classify the object. To avoid object selection bias, we shuffle the objects randomly.

Now we move to describe the datasets used in this setting, then talk about the results of the experiments.

3.1 The Datasets

The training examples revolve around messages that are classified as real threats by human analysts. These messages were collected using a system which is a part of a commercial product[3]. The only constraint on these examples is being published on the underground forums in 2021 and on. These examples constitute the positive training dataset part. We checked the underground forums where these messages were written, and the list of these forums is used to generate the negative examples, in a way such that for each positive example X of a forum, Y=n*X negative examples are collected randomly w.r.t the date constraint, where n is a factor, called *balance ratio*, which will be controlled in the experiments. The number of positive training examples is 595. The testing was performed on 960 positive examples and the negative testing examples were constructed in the same manner as training negative examples. These messages are collected from 12 different forums. Several experiments will be run with changing several parameters that might affect the final results. These factors are:

1. Dataset size and distribution: the only constant we have is the number of positive examples used. We test several sizes of negative/positive balance ratio values. In addition to this, we try to make a less random choice of negative examples, by filtering messages that might be spam messages (for instance, messages like "thanks" or "up" are considered to be of low value). We can also choose messages posted only in subsections of forums that are relevant to the classification (if we are trying to find messages about database leaks, we do not need messages coming from the credit cards section). Filtering based on message length is needed also, because some messages are articles, and while they would contain a lot of triggering keywords, they are not actual threats.

2. The intersection operator: we used set-theoretic intersection, then we use interval intersection (We will look into two-sided and one-sided intervals).

3. The number of attributes: the attributes are the values of tf-idf for the words of the messages, and changing the number of words to be in the attributes list is examined, and for which popularity of the keyword we can ignore it and remove it from the list is also considered. The values of attributes in case of theoretic set intersection are binary (the message either have the keyword or not). In case of interval intersection, the value of the attribute would be an interval beginning and ending with the tf-idf value.

4. The tolerance factor α, which represents the probabilistic relaxation allowed for counter examples.

3.2 Assessment Methods

We will have the usual classification assessment methods, but we need to be attentive to the case of unclassified examples. Handling unclassified examples can be done in three different ways:

[3] Provided by the cybersecurity firm Group-IB.

- Ignoring them by considering them negative examples, and that depends on how harsh our model is about classifying new examples. i.e., if the model classifies almost all positive examples as positive (high recall) and leaves only a very small number of unclassified examples relative to the number of new examples, then it might be possible to classify unclassified examples as negative, as the probability of them being actually positive is very low. This approach is not acceptable in cases where the cost of missing a positive example is high, which is the case for us, as missing a real threat is of a high cost.
- Moving all the unclassified examples to a human analyst, so they can assess their credibility, which can be done only in case the number of unclassified examples is low.
- Removing unclassified examples all together, by having a probabilistic relaxation of the classifier, so that it classifies examples based on how close they are to a class, not as zero-one state.

In addition to the usual definitions of assessment, we define another measure of the improvement this brings to human workers, by how much the messages they have to check shrunk in size. We call it "**saved effort**" and define as $1 - \frac{|G_{uncl}|}{|G_\tau|}$, where G_{uncl} is the set of unclassified examples $G_{uncl} \subseteq G_\tau$.

3.3 Testing Parameters

We first state our parameters that would be used in the future. We have $|G_+| = trp$ positive examples and $|G_-| = trn$ negative examples for the training dataset. We have zero unclassified examples in the training data. The number of negative examples $trn = trp * n$ where n is a positive integer.

The number of attributes is labeled as *att*. In the test dataset, we have *tsp* positive examples and *tsn* examples. The model might have a number of unclassified examples of the test dataset. This set of examples is labeled *uncl* and it is divided in two subsets: true positive examples that were left unclassified by the model (labeled *pos-uncl*) and true negative examples that were left unclassified by the model (labeled *neg-uncl*).

In the method we described above, the explanation of the model is the intersection of attributes of the test object and the concept object that gave us the result of the classification (meaning that the extension of this intersection of attributes all have/do not have the attribute value). This set is denoted by *sim*.

While we can control the balance of the training dataset in our case, this might not be the real world situation, which means that we would have a varying range of the values of saved effort and F1 measure, presenting us with a trade-off between coverage and model predictions correctness.

4 Results

We performed multiple experiments. The first one would have only binary attributes. The next would have intervals as attributes. In the next two experiments, the data is represented by one-sided intervals. After this, we repeat the pattern structures experiments but with varying values for α. Due to space limitations we present the results at https://github.com/abdulrahimGhazal/FCA-results

4.1 Binary Attributes

The attribute values here are TT-IDF values for the keywords contained in the vectorizer's vocabulary resulting from building the tf-idf model. The value of the attribute in an object would be represented here as:

$$att\_value(keyword) = \begin{cases} 1 & keyword \in vectorizer\ vocab \quad (5) \\ 0 & otherwise \quad (6) \end{cases}$$

We test a range of 5 values of the ratio between the positive to negative examples in the training dataset (from 1 to 5). We also test 5 different values of *min_df* which is a factor that controls the number of attributes that we would have. It specifies the threshold at which the TT-IDF builder ignores the term if it has less frequency in the documents. Theoretic set intersection is used.

The highest F1 in these experiments occurred on the first experiments where the dataset had the most keywords (lowest *min_df* step) and the same goes for the saved effort measure. The highest value was F1 = 98.8 and saved effort = 88.8 at (ratio, *min_df*) = (1, 0.01).

We can observe peaks that happen when we reset the *min_df* step in the training dataset, then a decline in the F1 afterwards until the next peak. This is a natural observation, since the *min_df* step increases between the peaks, meaning that we increase the number of ignored keywords that might play a role in classifying the messages correctly.

4.2 Relaxed Binary Attributes

We now repeat the same experiment, using only the lowest balance ratio, varying the values of *min_df*, and also varying values of α to allow for a certain small amount of counterexamples (objects of the other class matching the intersections used as classifiers).

The highest value was F1 = 98.6 at (α, ratio, *min_df*) = (90, 1, 0.01) and saved effort = 92.5 at (α, ratio, *min_df*) = (75, 1, 0.01).

Now we look at using pattern structures to classify underground messages by representing textual data as intervals (two-sided and one-sided).

4.3 Interval Representation

In this next part, we represent the values of the tf-idf as intervals of the floating point value such as if the tf-idf value is x, the attribute value would be [x,x]. In this setting, the intersection operator is defined as an interval that starts with the minimum of the two intervals' starts and ends with the maximum of the two intervals' ends.

$$[a_1, b_1] \sqcap [a_2, b_2] = [min(a_1, a_2), max[b_1, b_2)] \tag{7}$$

The highest F1 in these experiments occurred on the first experiments where the dataset had the most keywords (lowest *min_df* step) and the same goes for the saved effort measure. The highest value was F1 = 88.0 and saved effort = 87.7 at (ratio, *min_df*) = (1, 0.01). The values of F1 and saved effort decreases after that with increasing values of *min_df*.

4.4 One-Sided Interval Representation (Max)

Here, we represent the values of the tf-idf as intervals of the floating point value such as if the tf-idf value is x, the attribute value would be [x,∞].

In this setting, the intersection operator is defined as an interval that starts with the maximum of the two intervals' starts and ends with infinity.

$$[a_1, \infty] \sqcap [a_2, \infty] = [max(a_1, a_2), \infty] \tag{8}$$

The highest F1 in these experiments occurred on the first experiments where the dataset had the most keywords (lowest *min_df* step) and the same goes for the saved effort measure. The highest value was F1 = 88.0 and saved effort = 87.7 at (ratio, *min_df*) = (1, 0.01). The values of F1 and saved effort decreases after that with increasing values of *min_df*.

4.5 One-Sided Interval Representation (Min)

Here, we represent the values of the tf-idf as intervals of the floating point value such as if the tf-idf value is x, the attribute value would be [x,∞].

In this setting, the intersection operator is defined as an interval that starts with the minimum of the two intervals' starts and ends with infinity.

$$[a_1, \infty] \sqcap [a_2, \infty] = [min(a_1, a_2), \infty] \tag{9}$$

The highest F1 in these experiments occurred on the first experiments where the dataset had the most keywords (lowest *min_df* step) and the same goes for the saved effort measure. The highest value was F1 = 89.7 and saved effort = 87.6 at (ratio, *min_df*) = (1, 0.01). The values of F1 and saved effort decreases after that with increasing values of *min_df*.

4.6 Interval Representation With Probabilistic Relaxation

In this next part, we represent the values of the tf-idf as intervals of the floating point value such as if the tf-idf value is x, the attribute value would be [x,x]. As in before, the intersection operator is defined as an interval that starts with the minimum of the two intervals' starts and ends with the maximum of the two intervals' ends.

The difference now is testing several α values (see Algorithm 1).

$$[a_1, b_1] \sqcap [a_2, b_2] = [min(a_1, a_2), max[b_1, b_2)] \tag{10}$$

The highest value was F1 = 94.2 at (α, ratio, *min_df*) = (80, 1, 0.01) and saved effort = 94.1 at (α, ratio, *min_df*) = (75, 1, 0.01). The pattern noticed here is the F1 peaks we get when values of *min_df* increases locally at constant values of α, Then when we increase *min_df* significantly, the values of F1 decreases again.

4.7 One-Sided Interval Representation (Max) With Probabilistic Relaxation

Here, we represent the values of the tf-idf as intervals of the floating point value such as if the tf-idf value is x, the attribute value would be [x,∞]. The difference now is testing several α values (see Algorithm 1).

In this setting, the intersection operator is defined as follows:

$$[a_1, \infty] \sqcap [a_2, \infty] = [max(a_1, a_2), \infty] \tag{11}$$

The highest value was F1 = 91.7 at (α, ratio, *min_df*) = (95, 1, 0.01) and saved effort = 94.5 at (α, ratio, *min_df*) = ((85,90,95), 1, 0.01). The pattern noticed here is the F1 peaks we get when values of *min_df* increases locally at constant values of α, Then when we increase *min_df* significantly, the values of F1 decreases again.

4.8 One-Sided Interval Representation (Min) With Probabilistic Relaxation

Here, we represent the values of the tf-idf as intervals of the floating point value such as if the tf-idf value is x, the attribute value would be [x,∞]. The difference now is testing several α values (see Algorithm 1).

In this setting, the intersection operator is defined as an interval that starts with the minimum of the two intervals' starts and ends with infinity.

$$[a_1, \infty] \sqcap [a_2, \infty] = [min(a_1, a_2), \infty] \tag{12}$$

The highest value was F1 = 94.1 at (α, ratio, *min_df*) = (75, 1, 0.01) and saved effort = 94.2 at (α, ratio, *min_df*) = ((75,80), 1, 0.01). The pattern noticed here is the F1 peaks we get when values of *min_df* increases locally at constant values of α, Then when we increase *min_df* significantly, the values of F1 decreases again.

4.9 Discussion

Looking at the results, we can see that the binary values of the attributes model performed the best, because the attributes values and intersection method there (the theoretical set intersection) is the most restrictive among all others, giving a little room for error, but the issue with such method is that it suffers from this restrictions when totally new keywords are introduced, as it does not accept any keywords which were not used in the exact intersection.

We can also see that the less restrictive the conditions of the intersection, the less accurate it will be. The best conditions of pattern structures in non-binary representation of the data in our example is the Minimum one-sided intervals, because by its nature, it is the most inclusive of values that are non-zero in the classification scheme.

The next best one is the Maximum one-sided interval. While worse than the Minimum one-sided interval, because it is restricting attribute values to a smaller interval, it still outperforms the interval representation, because of how limited the values of interval representation are.

When relaxation is introduced, we can see that the values of F1 and the saved effort are higher than in the case with no relaxation, supporting that flexibility is useful in case of text messages classification.

The sizes of the *pos-uncl* were always smaller than *neg-uncl*, because of the limited set of keywords the model would have compared to the large amount of possible negative messages' keywords.

5 Conclusion

Underground forum messages are hackers announcements that are shared on the internet about attacks or tools used to carry out attacks. We presented an FCA-based approach for classifying forum messages based on their probability of being risky. The results of experiments show that the use of binary attributes (standard FCA) gave better accuracy, while the use of interval pattern structures gave better saved effort.

Acknowledgements The article was prepared within the framework of the Basic Research Program at HSE University, RF

References

1. McAfee: The Economic Impact of Cybercrime: No Slowing Down. McAfee, Santa Clara, CA, USA (2021)
2. Pastrana, S., Thomas, D. R., Hutchings, A., Clayton, R.:CrimeBB: Enabling Cybercrime Research on Underground Forums at Scale. In: International World Wide Web Conference, pp. 1845–1854. IW3C2, Lyon, France (2018)
3. Zone-H Homepage, http://www.zone-h.org/. Last accessed 12 May 2022

4. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag Berlin Heidelberg, Berlin, Germany (1999)
5. Kaytoue, M., Kuznetsov, S., Napoli, A., Duplessis, S: Mining gene expression data with pattern structures in formal concept analysis. Inf. Sci. **181**, 1989—2001 (2011)
6. Masyutin, A., Kashnitsky, Y., Kuznetsov, S: Lazy classification with interval pattern structures: application to credit scoring. In: FCA4AI@ IJCAI., pp. 43-–54. CEUR-WS.org, Buenos Aires, Argentina (2015)
7. Ben Boubaker Saidi, O., Tebourski, W: Formal Concept Analysis Based Association Rules Extraction. IJCSI International Journal of Computer Science Issues **8**(4), 490–497 (2011)
8. Obitko, M., Snasel, V., Smid, J: Ontology Design with Formal Concept Analysis. In: CLA., pp. 111–119. CLA (2004)
9. Jiang, G., Ogasawara, K., Endoh, A., Sakurai, T: Context-based ontology building support in clinical domains using formal concept analysis. International Journal of Medical Informatics **71**(1), 71–81 (2003)
10. Vilakone, P., Xinchang, K., Park, D. S: Movie recommendation system based on users' personal information and movies rated using the method of k-clique and normalized discounted cumulative gain. Journal of Information Processing Systems **16**(2), 494–507 (2003)
11. Dias, S. M., Vieira, N. J: Concept lattices reduction: Definition, analysis and classification. Expert Systems with Applications **42**(20), 7084–7097 (2015)
12. Kuznetsov, S. O., Obiedkov, S. A: Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence **14**(2-3), 189–216 (2002)
13. Kuznetsov, S. O: Machine Learning and Formal Concept Analysis. In: International Conference on Formal Concept Analysis., pp. 287–312. Springer, Berlin, Heidelberg (2004)
14. Kuznetsov, S. O., Makhazhanov, N., Ushakov, M: On neural network architecture based on concept lattices. In: International Symposium on Methodologies for Intelligent Systems., pp. 653–663. Springer, Cham (2017)
15. Ganter, B., Kuznetsov, S. O:Pattern Structures and Their Projections. In: International conference on conceptual structures., pp. 129–142. Springer, Berlin, Heidelberg (2001)
16. Kuznetsov, S. O:Scalable Knowledge Discovery in Complex Data with Pattern Structures. In: International Conference on Pattern Recognition and Machine Intelligence., pp.30–39. Springer, Berlin, Heidelberg (2013)
17. Kuznetsov, S. O:Fitting pattern structures to knowledge discovery in big data. In: International conference on formal concept analysis., pp.254–266. Springer, Berlin, Heidelberg (2013)

Organizing Contexts as a Lattice of Decision Trees for Machine Reading Comprehension

Boris Galitsky[1][0000−0003−0670−8520], Dmitry Ilvovsky[2][0000−0002−5484−372X], and Elizaveta Goncharova[2,3][0000−0001−8358−9647]

[1] Knowledge Trail Inc, San Jose, CA, USA
[2] Higher School of Economics, Moscow, Russia
[3] AIRI, Moscow, Russia

Abstract. Supported decision trees that have been first proposed to boost the performance and the explainability of the expert systems built upon the texts can become a great basis for the machine reading comprehension (MRC) systems. The supported decision tree is based on building and combining the corresponding discourse trees for the text passage. In this work, we build an environment of supported decision trees for the MRC task. Each answer is represented by a path of a supported decision tree and the whole corpus of answers is then form a lattice of supported decision trees. This environment gives a boost to MRC performance, handling cases where it is nontrivial to determine which document/passage MRC needs to be applied to.

1 Introduction

Machine reading comprehension (MRC) is a question answering task where the goal of the model is to read and understand text passages and answer the question about them. MRC is designed to check the language model's ability to understand text written in natural language, thus, in some cases, an answer cannot be retrieved from the given text passage, or it requires some world knowledge to answer a question. In such cases, a model should have access to the external knowledge base to retrieve the correct answer. The most promising technique to answer such questions is to augment the language model with the external knowledge database, such as Wikipedia, and to ensemble it with the additional retrieval component that supports the system with the relevant documents [10, 11].

Thus, in the cases, when an answer cannot be retrieved directly from a text, the system is split into two core components [1], where the former is an information retrieval system designed to identify useful pieces of text from the knowledge source (the retriever); and a system to produce the answer given the retrieved documents and the question (the reader).

The existing models that perform this problem are based on the transformer architecture and retrieve the relevant text passages based on the constructed latent representations. The main drawback of these techniques is the lack of explainability and limitation of the external documents that a model has access to (e.g., Wikipedia only). In this work, we propose to use supported decision trees DecTSup, first presented in [4] for expert systems as the basis for the relevant passages retrieval. The system allows to enrich the current textual corpora with external documents and use the organized rule-based structure in order to retrieve the relevant text passages that contain an answer to the asked question.

In comparison to our previous work, we show that the DecTSups can be built for the texts of the arbitrary genre. For the experimental evaluation, we refer to the MRC problem in the medical domain and show that utilizing DecTSup-based retrieval procedure improves the performance of the QA model over the standard MRC pipelines by up to 6% for the F1-score.

In a conventional MRC architecture, documents are not organized in any structure, and once a passage deemed most relevant is retrieved, the other documents are ignored. However, when humans answer questions, the answer is backed up by supporting documents so that the answer can be explained. In the real world question answering, the corpus of available documents serves the purpose of providing additional information and clarification on the answer topic. In this work we attempt to reproduce this feature of systematized MRC and organize the documents and passages into such structure as *lattice*. As multiple questions for a set of documents are answered, the involved documents are embedded into this lattice to form chains of explanation for obtained answer. We analyze the advantages of this lattice-based MRC architecture for delivering precise and explainable answers in a systematic way.

2 Background

2.1 Supported Decision Trees

DecTSups have been first presented in [4] as the basis for the expert systems that should retrieve some instructions from the textual data based on the input query. There, the authors claim that a flow of potential recommendations can be easily retrieved from the textual data and organized in the format of the decision tree (DecT) as for the standard numerical attributes. This flow of recommendations or instructions is extracted in the form of a discourse tree [2], where the nodes of the DecT are the elementary discourse units (EDUs) obtained from the discourse tree expressing some condition, and the edges are some type of the rhetorical relations that connect EDUs corresponding to different nodes. The DecT constructed from textual data can be enriched with the additional knowledge retrieved from the specific rhetorical relation and, thus, construct the *supported decision tree* (DecTSup). DecTSup can be easily integrated into the expert system as a basis to perform dialogue management that improves information retrieval procedures.

To turn a DecT into a corresponding DecTSup, each edge should be labeled with the information extracted from text for the given decision step:

1. The extracted entity;
2. The extracted phrase for the attribute for this entity;

3. The rhetorical relation;
4. The full nucleus and satellite EDUs.

3 Decision Chains and Discourse Structure

3.1 Rhetorical Structure Theory and Decision Chains Construction

A discourse tree (DT) is a basis for the DecT and DecTSup respectively. DT is a hierarchical structure that describes the relations that hold between text units in a document. Several theories have been proposed in the past to describe the discourse structure, among which RST [8] is one of the most developed. During discourse parsing, one can segment a document into non-overlapping text spans (contiguous units for clauses) called EDUs. Each of these EDUs can be tagged as either a nucleus or a satellite, where nucleus nodes are more central and satellite nodes more peripheral. Nucleus units consist of the main information the author expresses in the text, and satellite units contain additional information supporting the one presented in a nucleus. The EDU itself can be of different lengths, i.e., it can contain just one word or a word sequence. The discourse units are organized in the hierarchy by rhetorical relations (e.g., *Antithesis*, *Elaboration*, *List*, etc.) that reflect the function of these EDUs. As a result, we can represent the discourse structure of the text as a DT, where the relations at the top level cover the relations at the bottom.

To build the corresponding DecT, first, a *decision chain* should be retrieved from the DT. A decision chain is defined as a sequence of EDUs with rhetorical relations between sequence elements [3]. Elements of a decision chain are connected with $rhetorical_relation$ between a premise and a decision. It can be read as "If ⟨*premise*⟩ then make ⟨*decision*⟩ according to *rhetorical_relation*". In a decision chain, each consecutive member starting from the second one is a ⟨*decision*⟩.

An example of the decision chains and a fragment of DecTSup for a text passage presented below is given in Figure 1.

"Although there is no cure for type 2 diabetes, studies show it is possible for some people to reverse it. Through diet changes and weight loss, you may be able to reach and hold normal blood sugar levels without medication. This does not mean you are completely cured. Type 2 diabetes is an ongoing disease. Even if you are in remission, which means you are not taking medication and your blood sugar levels stay in a healthy range, there is always a chance, that symptoms will return. But it is possible for some people to go years without trouble controlling their glucose and the health concerns that come with diabetes."

elaboration

explanation

contrast

TEXT: Although there is no cure for type 2 diabetes,

attribution

TEXT: studies show

```
      enablement
        TEXT: it is possible for some people
        TEXT: to reverse it.
  evaluation
    condition
      TEXT: Through diet changes and weight loss,
      manner-means
        TEXT: you may be able to reach and hold normal blood sugar levels
        TEXT: without medication.
      TEXT: This does not mean you are completely cured.
    elaboration
      TEXT: Type 2 diabetes is an ongoing disease.
      contrast (RightToLeft)
        elaboration(RightToLeft)
          same-unit
            condition
              TEXT: Even if you are in remission,
              joint
                TEXT: which means you are not taking medication
                TEXT: and your blood sugar levels stay in a healthy range ,
              TEXT: there is always a chance ,
            TEXT: that symptoms will return .
          background
            TEXT: But it is possible for some people to go years
            elaboration
              TEXT: without trouble
              elaboration
                TEXT: controlling their glucose and the health concerns
```

When a text is represented as a discourse tree, it is split into elementary discourse units (EDUs), denoted by 'TEXT' tag. EDUs are organized hierarchically according to rhetorical relations between them. Using the constructed DT, for an arbitrary rhetorical relation, we can define some patterns defining, how EDUs are connected to each other. In particular, relation of *Elaboration*, ⟨satellite⟩ elaborates (provides additional information) on ⟨nucleus⟩. Certain rhetorical relations have an obvious interpretations in terms of what decision ⟨satellite⟩ can be made by means of ⟨nucleus⟩. For example, $Enablement(result) \Rightarrow$ possible to achieve result ⟨nucleus⟩ by the way of ⟨satellite⟨. Based on this logical connection identified by the specific EDUs, we can retrieve possible decision chains for the read passage given bellow:

$\textit{diet changes and weight loss} \Rightarrow^{condition} \textit{without medication} \Rightarrow^{manner-means} \textit{sugar(normal)}$

$\textit{Remission} \Rightarrow^{condition} \textit{not taking medications} \;\; \textit{sugar(normal)} \Rightarrow^{elaboration} \textit{chance}\diamond \; \textit{symptoms(yes)}$

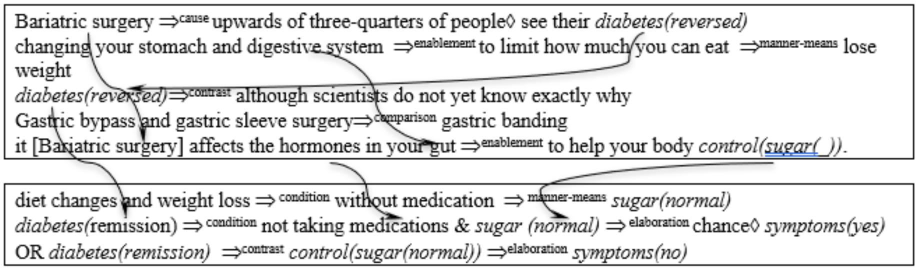
$\textit{OR Remission} \Rightarrow^{contrast} \textit{control(sugar(normal))} \Rightarrow^{elaboration} \textit{symptoms(no)}$

We denote *sugar(normal)* as a formal representation of target values. Formal representations are shown in italic, and original text – in a regular font.

chance◇, possibility◇ are the modalities which do not change the configuration of a DecTSup but control the probability of navigation of the given decision chain.

Formally, a decision chain is defined as a sequence of EDUs with rhetorical relations between sequence elements [3]. Each element is a whole original EDU or its representation as a logic form that can be obtained as a result of a semantic parsing: it depends whether an entity from this EDU occurs in an available ontology or not. We encourage the readers to refer to [6] for a detailed explanation of how to obtain semantic-based representation of the EDU. For formalized elements of decision chains, it is easier to establish a correspondence or synonymy between entities to form a decision navigation graph.

Elements of a decision chain are connected with $\Rightarrow^{rhetorical\_relation}$ between a premise and a decision. It can be read as "If ⟨premise⟩ then make ⟨decision⟩ according to *rhetorical_relation*". In a decision chain, each consecutive member starting from the second one is a ⟨decision⟩. Each previous member is a ⟨premise⟩.



Bariatric surgery $\Rightarrow^{cause}$ upwards of three-quarters of people◊ see their *diabetes(reversed)*
changing your stomach and digestive system $\Rightarrow^{enablement}$ to limit how much you can eat $\Rightarrow^{manner-means}$ lose weight
diabetes(reversed) $\Rightarrow^{contrast}$ although scientists do not yet know exactly why
Gastric bypass and gastric sleeve surgery $\Rightarrow^{comparison}$ gastric banding
it [Bariatric surgery] affects the hormones in your gut $\Rightarrow^{enablement}$ to help your body *control(sugar(_))*.

diet changes and weight loss $\Rightarrow^{condition}$ without medication $\Rightarrow^{manner-means}$ *sugar(normal)*
*diabetes(*remission) $\Rightarrow^{condition}$ not taking medications & *sugar (normal)* $\Rightarrow^{elaboration}$ chance◊ *symptoms(yes)*
OR *diabetes(remission)* $\Rightarrow^{contrast}$ *control(sugar(normal))* $\Rightarrow^{elaboration}$ *symptoms(no)*

Fig. 1. Two sections of decision chains

Figure 1 shows two sections of decision chains extracted from the two texts above. Arrows connect the same (or corresponding) entities, (possibly, parameterized diferently) such as $control(sugar(\_)) \rightarrow sugar(normal))$.

In the first formalized decision expression $control(sugar(\_))$, the outermost predicate is $control(\_)$ that ranges over control subjects such as $sugar(\_)$ with an anonymized variable '_'.

4 DecTSup environment for Machine Reading Comprehension

4.1 Basic Example

MRC is a task to retrieve the correct answer span from the given textual context. However, the context which is given to MRC system usually restricts its applicability in real-life applications. As, in many cases, the given passage may not contain the necessary information. Efforts made in multi-passage MRC research have somewhat broken the limitation of the given context, but there is still a long way to go as how to find the most relevant resources for MRC systems effectively determines the performance of answer prediction. Moreover, the existing DL MRC systems often fail to capture long-range dependencies existing in a text, thus, even if an answer can be retrieved from a context, the system may not be able to find it. It calls for a deeper combination of information retrieval and machine reading comprehension, which we do in this work.

Let us consider the example, where in order to answer a simple question, MRC model should be aware of the additional information.

Passage: There is an ice cube in a glass of water. When the ice cube melts, will the water level have risen, fallen, or remained the same? We have an ice cube floating in the water. If it is floating in equilibrium, then it will have to displace enough water to support its weight. When the ice has melted, it turns into exactly the same volume as it displaced before. So the added volume is the same, so the level of the water will not change.

Question: Into what does ice melt?

Answer: The same volume as it displaced before.

Relying on this text, the transformer-based MRC system cannot answer a very basic question about melting ice into water. Required knowledge is not spelled out in explanation but instead is assumed to be known to the reader. As MRC has no means to acquire it, this knowledge should be added in some way or another. A complete hypothetical decision tree for inferring a solution to a physics problem contains hints on which texts and/or ontology expressions are needed to answer all question about the physical system described in the formulated problem.

We postulate a hypothesis that answering a question, given a passage constitutes navigating a fragment of a decision tree build from this passage. While this approach seems natural when this passage describes some form of a decision explicitly, and certain abstraction is required for an arbitrary text genre where decisions are hypothetical.

4.2 DecTSup MRC Architecture

We propose to treat any text as some kind of problem formulation so that a respective decision tree would tell the MRC system which knowledge is necessary to have complete domain coverage. Usually, a text describes just a single path in a decision tree. Mining for other texts helps reconstruct texts for other paths. Otherwise, it is unclear if even basic questions can be answered (see 4.1). Hence forming a DecTSup for a given passage assure better domain coverage so that the user can expect any relevant question to be answered reasonably well.

If a question requires building a logical chain of facts or forming a decision, the construction of a decision tree in the course of question-answering seems natural. The assumption we make in this study is that a logical chain of facts and respective decision structure is associated with any text, of an arbitrary genre. Considering a given text passage within a decision structure is necessary to determine which other passages need to be involved. This approach is expected to be much more robust and precise than a traditional information retrieval (IR) based, where candidate passages are determined based on keywords.

Under regular MRC architecture, an IR component first finds a document and a passage, and then MRC finds an exact answer in this passage. In the proposed MRC architecture, the IR system finds candidate documents along with DecTSup built for these documents, if available. Then the DecTSup-MRC components use the lattice of DecTSups to decide on which passages need to be involved in the answer.

We draw the feature-by-feature comparison of three MRC environments: default, retrieval-augmented [9] and DecTSup-based in Table 1.

In Figure 2, we present the architecture for the standard MRC pipeline and MRC with the DecTSup.

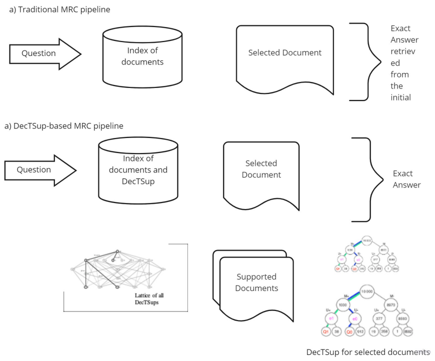

a) Traditional MRC pipeline
Question
Index of documents
Selected Document
Exact Answer retrieved from the initial
a) DecTSup-based MRC pipeline
Question
Index of documents and DecTSup
Selected Document
Exact Answer
Lattice of all DecTSups
Supported Documents
DecTSup for selected documents

Fig. 2. MRC pipelines: a) classical NRC pipeline; b) retrieval-based MRC pipeline based on DecTSup

4.3 An Algorithm for building DecTSup Environment

The essence of building DecTSup environment for MRC is organizing mapping between set of passages P and DecTSups. This algorithm is given a set of documents/passages P_s in a corpus and a set of external documents/passages E_s, and also given a sequence of queries Q_s against P_s, builds a set of supported decision trees DecTSups.

An answer a is obtained as a function MRC($\langle q, p\rangle$). A true answer a_{true} may need another passage, not necessarily p, including other passages p' external document h belong to E_s. The whole passage p corresponds to a DecTSup(p) and an answer a – to a path of this DecTSup(p) $\leftarrow path_{DecTSup(p)}(a)$. There is a many-to-many mapping between multiple passages and DecTSups.

An algorithm for building this DecTSup is as follows.

Table 1. Comparison of the IR-based MRC environments

Feature / Architecture	Default MRC	Retrieval-augmented approach	DecTSup-based
How passages and answers are organized	Unordered set of passages and answers	A model decides which passages to involve	Passages are linked with each other via DecTSups which are in turns organized in a lattice
Treatment of texts describing decisions	–	No special treatment	Decision structure is reflected in how passages are organized. Relying on an assumption that a question concern a single decision step
Inter-connection between answers / passages / document	–	Implicit	Answers are inter-connected
How links between passages and documents can be expressed	–	Implicit	Via common parts of a path in a DecTSup
Explanation for why the exact answer is chosen	–	Implicit, via optimization	Via the full path in DecTSup
Explanation for why the documents containing the answer are selected	–	Implicit, via optimization	Via the lattice of DecTSups
Explanation for why certain entities occur in the answer	–	–	–
Training for document retriever	–	marginalization over sets of retrieved documents is approximated using an expectation-maximization algorithm.	–
Providing background info	–	–	A search session can be followed by a navigation session

Initially, DecTSup = $\emptyset$. Given a corpus of passages P_s, iterate through all available queries Q_s and build a set $\cup_{p \in P} DecTSup(p)$.

For each $\langle q, ? \rangle$

1. Compute $a = MRC(\langle q, p \rangle)$.

2. Compute a_{true} checking if passage p is suitable. If not find another p' or a suitable external doc $p = h$, using the lattice L. Decide whether to substitute or to augment p.
3. Build a chain of phrases from a_{true} for DecTSup(a).
4. Build $path_{DecTSup(a)}$ from this chain of phrases. Only $path_{DecTSup(p)}$ is currently available, not the whole tree.
5. Extend $path_{DecTSup(a)}$ towards a DecTSup by turning all phrases into negations.
6. Update the DecTSups system, identifying a location for $path_{DecTSup(a)}$:

a. Find existing path $path_{DecTSup(p)}$ for $\langle q, p\rangle$ and verify $path_{DecTSup(p)}$ covers $path_{DecTSup(a)}$.
b. If not, find current DecTSup(p) and augment it with $path_{DecTSup(a)}$. Check for overlapping nodes and do the path merge if appropriate;
c. If no such DecTSup(p) exists, form a fragment of new DecTSup(p) from $path_{DecTSup(a)}$.

Once all available queries $q \in Q_s$ are ran, organize a set of DecTSup(p) into a lattice (see 4.4). When a new batch of queries arrive, re-apply this algorithm and then recompute the lattice.

The procedure of extension of $path_{DecTSup(a)}$ towards a $DecTSup$ is implemented by turning all phrases into negations works as follows. For each phrase associated with a path node, we consider its negation and branch it off this node. If a phrase in $path_{DecTSup(a)}$ is already a negation, remove this negation.

This algorithm is also naturally applied when P is a book split into passages, the result can be viewed as a book hyperlink graph.

During MRC environment construction, each query produces an answer that maps into a path in the DecTSup- based MRC environment. Initially, most queries form new paths which are then merged into a part of the future DecTSups. Once new batches of queries does not construct new DecTSups, we form a lattice from them.

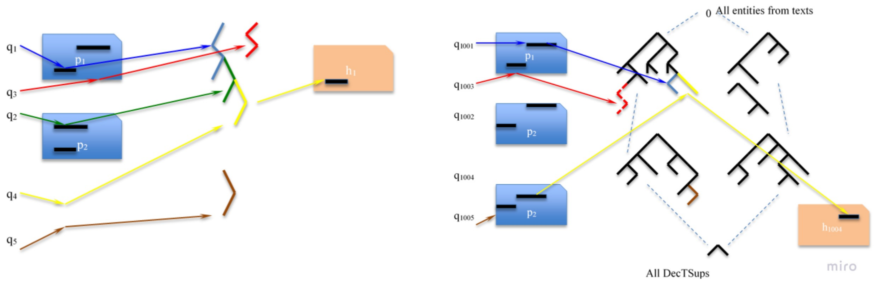


Fig. 3. Building the lattice of DecTSup.

In Figure 3, we show the end of the learning session for building the lattice of DecTSups. Most individual DecTSups are now built and new queries do not initiate additions of new paths to DecTSups. Most of the DecTSups are completed and now form a lattice (shown by dotted lines). Black nodes of DecTSup show nodes that have been formed by the time queries q1001, ... are launched.

Parts of the DecTSups corresponding to the given queries are shown in color. For example, query q1001 is connected by the blue arrow with its answer and then with its blue path in the top-left DecTSup. This blue path has just been added. On the contrary, query q1002 is mapped into the existing part of the DecTSup (shown in dotted red). Query q1001 requires a new external passage h1004.

4.4 Decision Trees and Concept Lattices

The following phase is responsible for retrieving the relevant text passages from the obtained DecTSupp of the documents. In this work, we propose to form a concept lattice of all DecTSups for a corpus of documents to enable this corpus with a structure assuring effective MRC. For a text and its DecTSup, we extract entities and their attributes used for decisions and form the formal context.

We use the essential ideas of Lindig [7] algorithm which efficiently generates formal concepts together with their subconcept-superconcept hierarchy in the form of concept lattice. The algorithm builds the concept lattice by iteratively generating the neighbor concepts of a concept $\langle A, B\rangle$, either top-down the lattice by adding new attributes to concept intents or bottom-up by adding new objects to concept extents.

For each $A \subseteq X$ and $B \subseteq Y$ we denote by $A^{'}$ a subset of Y and by $B^{'}$ a subset of X defined by

$A^{'} = \{y \in Y | \texttt{ for each } x \in A : hx, yi \in I\}$,

$B^{'} = \{x \in X | \texttt{ for each } y \in B : hx, yi \in I\}$.

That is, $A^{'}$ is the set of all attributes (text entities) from Y shared by all objects (DecTSups) from A (and similarly for $B^{'}$).

A formal concept consists of a set A (so-called extent) of objects which fall under the concept and a set B (so-called intent) of attributes that fall under the concept such that A is the set of all objects sharing all attributes from B and, conversely, B is the collection of all attributes from Y shared by all objects from A.

The algorithm is based on the fact that a concept $\langle C, D\rangle$ is a neighbor of a given concept $\langle A, B\rangle$ if D is generated by $B \cup \{y\}$, i.e. $D = (B \cup \{y\})^{''}$, where $y \in Y \backslash B$ is an attribute such that for all attributes $z \in D$ - B it holds that $B \cup \{z\}$ generates the same concept $\langle C, D\rangle$, i.e. neighbors of $\langle A, B\rangle$ = $\{\langle C, D\rangle | D = (B\{y\})^{''}, y \in Y \backslash B$ such that $(B \cup \{z\})^{''} = D$ for all $z \in DB\}$.

Then a selection of a tree of concepts from the part of the concept lattice occurs. First, for each concept $c = \langle A, B\rangle$ the number L_c of all of its lower concepts is computed. Each lower concept is counted for each different attribute added to the concept c, l_c. For instance, if a concept $d = \langle C, D\rangle$ is generated from concept c by adding either attribute x or attribute y (i.e. $D = (B \cup \{x\})^{''}$ or $D = (B \cup \{y\})^{''}$, respectively), the concept d is counted twice and l_c is increased by two.

Then a tree of concepts is chosen from the part of the concept lattice by iteratively going from the greatest concept (generated by no attributes or, equivalently, by all objects) to minimal concepts. The selection is based on the number l_c of lower concepts of the currently considered concept c.

4.5 Online selection of passages for a query

At search time, to select the passage, we match the query with a chain of phrases for a path in a DecTSup. We try to find such DecTSup so that as many chain phrases path(i) match query phrases as possible.

$$DecTSup : \sum_i score(q \wedge path(i)_{DecTSup}) \rightarrow max$$

$\wedge$ is a syntactic generalization operator [5].

Once a single DecTSup is identified, we select the passages p_s associated with matched nodes of DecTSup.

In some cases, the best match between the query and path phrases can be distributed through multiple DecTSups. Then the condition of connectedness in the lattice L of DecTSups must be maintained:

$$DecTSups(j) : \sum_{i,j} score(q \wedge path(i) DecTSup(j)) \rightarrow max \ \texttt{\&}$$

$$\texttt{\&} \ node(DecTSups(j) \in DecTSups(k)) \in L.$$

To summarize the preference for the occurrence of an answer in a corpus of documents, we prefer it to be in a single passage. If it is not possible, we extend this passage towards other passages connected vis a DecTSup. If it is still insufficient for answer identification, we further extend passages towards foreign DecTSups linked in a lattice. Finally, in the worst case scenario, passages come from unrelated documents associated with DecTSups anywhere in the lattice.

Linked lattice nodes correspond to passages containing the same entities plus minus one or two, therefore, closely related. This is maintained by how the lattice is defined on the set of all DecTSups.

5 Evaluation

We check the MRC model performance on the collection of medical instruction collected from the WebMD website and syntactically generate the question that could be asked regarding the retrieved passages. We select ten classes of diseases to diversify the experiments, and track each processing step to identify the performance bottleneck. It should be mentioned that the F1-score reported to assess the model's performance is applied for each topic independently assessing whether a retrieved answer is correct or not.

In the baseline, we rely on IR to identify passages in the fixed corpus of documents, so the main source of errors is an improperly selected passage or a lack

Table 2. Evaluation results for the IR in medical domain

Search domain	Baseline F1 (IR-based)	F1 with synt. gen. between q and p	F1 with a set of DecTSups	F1 with a lattice of DecTSups
Bloating	76.3	77.0	79.3	80.4
Cough	75.2	76.8	76.9	82.4
Diarrhea	77.0	75.9	78.0	79.3
Dizziness	77.9	77.1	79.8	81.9
Fatigue	72.6	75.4	80.4	82.7
Fever	71.9	74.5	78.0	79.3
Headache	74.6	74.8	81.5	82.7
Muscle Cramp	77.3	76.0	80.6	81.0
Nausea	72.1	76.4	78.4	80.7
Throat irritation	75.9	73.2	79.5	82.3
Average	75.1	75.7	79.2	81.3

of appropriate passage in the corpus. Our second baseline in the third column is syntactic generalization between query q and passage p which is better than keyword frequency maintained by IR but lags behind the approach developed in this paper. In the fourth column, we show F1 of DecTSup-enabled MRC environment. The fifth column shows the contribution of ideal, corrected DecTSups and the last, sixth column should the contribution of the lattice of DecTSups, where individual trees are organized to systematically identify additional passages which need to be involved in finding the exact answer.

The second baseline is only 0.6% better then the IR-based document identification, which makes the syntactic generalization insufficient for the robust identification of relevant passages to form the exact comprehensive answer. We observe that an unstructured MRC can have a boost of 4% F1-score by finding the best document by DecTSup environment. Proceeding from IR passage selection to the one based on independent DecTSups turns out to be an ultimate win for the MRC environment. A further upgrade from a set of DecTSups to a lattice gives further 2% boost in search performance.

6 Conclusion

In this work, we explored a way to build decision trees from text relying on discourse analysis and use this environment to boost the MRC model's performance. We use DecTSup environment for retrieving the relevant documents and passages, where the answer can be found. We compare the DecTSup-based retrieval procedures with several baselines including syntactic generalization for matching questions and passages and keywords matching on the set of the medical instructions texts. We show that utilizing the DecTSup as the description of texts and combining them into the concept lattice improves the performance of MRC by up to 6% on average. In future studies, we will consider building a concept lattice from a textual description of data [5], instead of a decision tree for authors' instructions on how to do things and make decisions in the course of it.

References

1. Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics.
2. Boris Galitsky. Matching parse thickets for open domain question answering. *Data Knowledge Engineering*, 107:24–50, 2017.
3. Boris Galitsky. *Managing Customer Relations in an Explainable Way*, pages 309–377. Springer International Publishing, Cham, 2020.
4. Boris Galitsky. *Chapter 3 - Obtaining supported decision trees from text for health system applications*. Academic Press, 2022.
5. Boris Galitsky, Gabor Dobrocsi, Josep Rosa, and Sergei Kuznetsov. Using generalization of syntactic parse trees for taxonomy capture on the web. volume 6828, pages 104–117, 07 2011.
6. Boris A. Galitsky and Dmitry Ilvovsky. Validating correctness of textual explanation with complete discourse trees. In *FCA4AI@IJCAI*, 2019.
7. Christian Lindig. Fast concept analysis. In *Working with Conceptual Structures – Contributions to ICCS 2000*, pages 152–161. Shaker Verlag, 2000.
8. William Mann and Sandra Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8, 1988.
9. Devendra Singh Sachan, Siva Reddy, William Hamilton, Chris Dyer, and Dani Yogatama. End-to-end training of multi-document reader and retriever for open-domain question answering. In *NeurIPS*, 2021.
10. Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.
11. Yizhong Wang, Kai Liu, Jing Liu, Wei He, Yajuan Lyu, Hua Wu, Sujian Li, and Haifeng Wang. Multi-passage machine reading comprehension with cross-passage answer verification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1918–1927, Melbourne, Australia, July 2018. Association for Computational Linguistics.