A Logic-based Tool for Dynamic Generation and Classification of Musical Content

Antonio Lieto[1 (✉) [0000−0002−8323−8764]], Gian Luca Pozzato[1 (✉) [0000−0002−3952−4624]], Alberto Valese[1], and Mattia Zito[1]

Dipartimento di Informatica, Università di Torino, Italy
{antonio.lieto, gianluca.pozzato, alberto.valese}@unito.it, mattia.zito@edu.unito.it

Abstract. In this work we present NERVOUS, an intelligent recommender system exploiting a probabilistic extension of a Description Logic of typicality to dynamically generate novel contents in AllMusic, a comprehensive and in-depth resource about music, providing data about albums, bands, musicians and songs (https://www.allmusic.com). The tool can be used for both the generation of novel music genres and styles, described by a set of typical properties characterizing them, and the reclassification of the available songs within such new genres.

1 Introduction

The ability of generating new knowledge via conceptual combination concerns high-level capacities associated to creative thinking and problem solving, and it represents an open challenge for artificial intelligence [2]. Indeed, dealing with this problem requires, from an AI perspective, the harmonization of two conflicting requirements: on the one hand, the need of a syntactic and semantic compositionality; on the other hand, the need of capturing typicality effects. However, such requirements can be hardly accommodated in standard symbolic systems, including formal ontologies [4]. According to a well-known argument [18], prototypes, namely commonsense conceptual representations based on typical properties, are not compositional. Consider a concept like *pet fish*: it results from the composition of the concept *pet* and of the concept *fish*, however, the prototype of *pet fish* cannot result from the composition of the prototypes of a pet and a fish. For instance, a typical pet is furry, whereas a typical fish is grayish, but a typical pet fish is neither furry nor grayish (typically, it is red). This is a paradigmatic example of the difficulty to address when building formalisms and systems trying to imitate this combinatorial human ability. Examples of such difficulties concern handling exceptions to attribute inheritance and handling the possible inconsistencies arising between conflicting properties of the concepts to be combined.

In this work we continue our activity started in [10,9] with the definition of a Typicality Description Logic for concept combination ($\mathbf{T}^{\mathsf{CL}}$, typicality-based compositional logic), that we have exploited in order to build a goal-oriented framework for knowledge invention in the cognitive architecture of SOAR [8,12,11], as well as for the generation and the suggestion of novel editorial content in multimedia broadcasting [3] and in the artistic domain of paintings, poetic content [15], and museum items [13]. In the Description Logic $\mathbf{T}^{\textsf{CL}}$, "typical" properties can be directly specified by means of a "typicality" operator $\mathbf{T}$ enriching the underlying DL, and a TBox can contain inclusions of the form $\mathbf{T}(C) \sqsubseteq D$ to represent that "typical Cs are also Ds". As a difference with standard DLs, in the logic $\mathbf{T}^{\textsf{CL}}$ one can consistently express exceptions and reason about defeasible inheritance as well. Typicality inclusions are also equipped by a real number $p \in (0.5, 1]$ representing the probability/degree of belief in such a typical property: this allows us to define a semantics inspired to the DISPONTE semantics [20] characterizing probabilistic extensions of DLs, which in turn is used in order to describe different *scenarios* where only some typicality properties are considered. Given a KB containing the description of two concepts C_H and C_M occurring in it, we then consider only *some* scenarios in order to define a revised knowledge base, enriched by typical properties of the combined concept $C \sqsubseteq C_H \sqcap C_M$ by also implementing a HEAD/MODIFIER heuristics coming from the cognitive semantics.

In this work we exploit the logic $\mathbf{T}^{\textsf{CL}}$ in order to dynamically generate novel knowledge by means of a mechanism for commonsense combination, that we apply to data extracted from AllMusic (https://www.allmusic.com), a comprehensive and in-depth resource about music. In particular, we introduce NERVOUS (dyNamic gEneratoR of noVel cOntent in mUSic), a tool which is able to compute the following activities:

- it builds the prototypical description of 18 basic musical genres (Blues, Classical, Country, Easy Listening, Holiday and so on), by extracting data about musical genres and songs in AllMusic by means of a crawler. Such prototypes are formalized by means of a $\mathbf{T}^{\textsf{CL}}$ knowledge base, whose TBox contains both *rigid* inclusions of the form
$$BasicGenre \sqsubseteq Concept$$
in order to express essential desiderata but also constraints, for instance $Childrens \sqsubseteq \neg Sex$ (due to law restrictions, sexual contents for kids are forbidden), as well as *prototypical* properties of the form
$$p \ :: \ \mathbf{T}(BasicGenre) \sqsubseteq TypicalConcept,$$
representing typical concepts of a given genre, where p is a real number in the range $(0.5, 1]$, expressing the degree of belief of such a concept in items belonging to that genre: for instance, $0.84 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq Cerebral$ is used to express that typical songs belonging to the Avant-garde genre are Cerebral (in some sense) with a probability/degree of belief of the 84%, and such a degree is automatically extracted by NERVOUS from the data available on AllMusic for that genre;
- it allows the generation of new musical genres by exploiting the reasoning capabilities of the logic $\mathbf{T}^{\textsf{CL}}$ in order to generate new *derived* genres as the result of the creative combination of two basic or derived ones;
- it implements a mechanism of reclassification of the available songs of AllMusic within new genres generated in the previous phase. Intuitively, a song is classified as belonging to the new genre if its moods and themes match the typical properties of the prototype of such a genre, obtaining a *score of compatibility* higher than 0. A positive matching, namely the same property has a high score in the song and is

a typical property in the genre, provides a positive score, whereas a negative one, e.g. the song has a high score for a property which is negated in the prototype of the genre, produces a negative score. Songs having at least one positive match and having no negative ones has an overall positive score and is then recommended by NERVOUS for that genre.

We have tested NERVOUS by reclassifying the available songs in the highlights of AllMusic with respect to the new generated genres, as well as with an evaluation, in the form of a controlled user study experiment, of the feasibility of using the obtained reclassifications as recommended contents. The obtained results are encouraging and pave the way to many possible further improvements and research directions.

2 Combining Concepts: the Description Logic $\mathbf{T}^{\mathsf{CL}}$

The tool NERVOUS exploits the Description Logic $\mathbf{T}^{\mathsf{CL}}$ [9,10] for the generation of new genres as the combination of two existing ones. The language of $\mathbf{T}^{\mathsf{CL}}$ extends the basic DL $\mathcal{ALC}$ by *typicality inclusions* of the form

$$p \ :: \ \mathbf{T}(C) \sqsubseteq D$$

where $p \in (0.5, 1]$ is a real number representing its degree of belief, whose meaning is that "we believe with degree p that, normally, Cs are also Ds". We avoid degrees $p \leq 0.5$ since it would be misleading for typicality inclusions, since typical knowledge is known to come with a low degree of uncertainty.

We define a knowledge base $\mathcal{K} = \langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{R}$ is a finite set of rigid properties of the form $C \sqsubseteq D$, $\mathcal{T}$ is a finite set of typicality properties of the form $p \ :: \ \mathbf{T}(C) \sqsubseteq D$ where $p \in (0.5, 1] \subseteq \mathbb{R}$ is the degree of belief of the typicality inclusion, and $\mathcal{A}$ is the ABox, i.e. a finite set of formulas of the form either $C(a)$ or $R(a, b)$, where $a, b \in \mathtt{O}$ and $R \in \mathtt{R}$.

The Description Logic $\mathbf{T}^{\mathsf{CL}}$ relies on the DL of typicality $\mathcal{ALC} + \mathbf{T_R}$ introduced in [5], which allows one to describe the *prototype* of a concept, in this case a musical genre. As a difference with standard DLs, in the logic $\mathcal{ALC} + \mathbf{T_R}$ one can consistently express exceptions and reason about defeasible inheritance as well. For instance, a knowledge base can consistently express that "typical students are young persons", whereas "normally, senior students are not young persons" by $\mathbf{T}(\mathit{Student}) \sqsubseteq \mathit{Young}$ and $\mathbf{T}(\mathit{SeniorStudent}) \sqsubseteq \neg \mathit{Young}$, given a knowledge base also containing the standard inclusion $\mathit{SeniorStudent} \sqsubseteq \mathit{Student}$, representing that all senior students are students. The semantics of the $\mathbf{T}$ operator is characterized by the properties of *rational logic* [7], recognized as the core properties of nonmonotonic reasoning. The Description Logic $\mathcal{ALC} + \mathbf{T_R}$ is characterized by a minimal model semantics corresponding to an extension to DLs of a notion of *rational closure* as defined in [7] for propositional logic: the idea is to adopt a preference relation among $\mathcal{ALC} + \mathbf{T_R}$ models, where intuitively a model is preferred to another one if it contains less exceptional elements, as well as a notion of *minimal entailment* restricted to models that are minimal with respect to such preference relation. As a consequence, the operator $\mathbf{T}$ inherits well-established properties like *specificity* and *irrelevance*; in the example, the Description Logic $\mathcal{ALC} + \mathbf{T_R}$

allows one to infer that $\mathbf{T}(\mathit{Student} \sqcap \mathit{Italian}) \sqsubseteq \mathit{Young}$ (being Italian is irrelevant with respect to being young) and, if one knows that Rachel is a typical senior student, to infer that she is not young, giving preference to the most specific information.

A model $\mathcal{M}$ of $\mathbf{T}^{\mathsf{CL}}$ extends standard $\mathcal{ALC}$ models by a preference relation among domain elements as in the logic of typicality [5]. In this respect, $x < y$ means that x is "more normal" than y, and that the typical members of a concept C are the minimal elements of C with respect to this relation. An element $x \in \Delta^{\mathcal{I}}$ is a *typical instance* of some concept C if $x \in C^{\mathcal{I}}$ and there is no C-element in $\Delta^{\mathcal{I}}$ *more normal* than x.

Definition 1 (Model of $\mathbf{T}^{\mathsf{CL}}$). *A model $\mathcal{M}$ is any structure $\langle \Delta^{\mathcal{I}}, <, .^{\mathcal{I}} \rangle$ where: (i) $\Delta^{\mathcal{I}}$ is a non empty set of items called the domain; (ii) $<$ is an irreflexive, transitive, well-founded and modular (for all x, y, z in $\Delta^{\mathcal{I}}$, if $x < y$ then either $x < z$ or $z < y$) relation over $\Delta^{\mathcal{I}}$; (iii) $.^{\mathcal{I}}$ is the extension function that maps each atomic concept C to $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each role R to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and is extended to complex concepts in the standard way for standard connectives, whereas for the typicality operator we have $(\mathbf{T}(C))^{\mathcal{I}} = Min_<(C^{\mathcal{I}})$, where $Min_<(C^{\mathcal{I}}) = \{x \in C^{\mathcal{I}} \mid \nexists y \in C^{\mathcal{I}}$ s.t. $y < x\}$.*

In order to perform useful nonmonotonic inferences, in [5] the above semantics is strengthened by restricting entailment to a class of minimal models. Intuitively, the idea is to restrict entailment to models that *minimize the atypical instances of a concept*. The resulting logic corresponds to a notion of *rational closure* on top of $\mathcal{ALC} + \mathbf{T_R}$. Such a notion is a natural extension of the rational closure construction provided in [7] for the propositional logic. This nonmonotonic semantics relies on minimal rational models that minimize the *rank of domain elements*. Informally, given two models of KB, one in which a given domain element x has rank 2 (because for instance $z < y < x$), and another in which it has rank 1 (because only $y < x$), we prefer the latter, as in this model the element x is assumed to be "more typical" than in the former. Query entailment is then restricted to minimal *canonical models*. The intuition is that a canonical model contains all the individuals that enjoy properties that are consistent with KB.

The Description Logic $\mathbf{T}^{\mathsf{CL}}$ considers a distributed semantics similar to DISPONTE [21] for probabilistic DLs. This logic allows one to label inclusions $\mathbf{T}(C) \sqsubseteq D$ with a real number between 0.5 and 1, representing its degree of belief, assuming that each axiom is independent from each others. Degrees in typicality inclusions allow one to define a probability distribution over *scenarios*: intuitively, a scenario is obtained by choosing, for each typicality inclusion, whether it is considered as true or false. In an extension of the above example, we could have the following KB:

(1) $\mathit{SeniorStudent} \sqsubseteq \mathit{Student}$
(2) $0.70\ ::\ \mathbf{T}(\mathit{Student}) \sqsubseteq \mathit{Young}$
(3) $0.95\ ::\ \mathbf{T}(\mathit{SeniorStudent}) \sqsubseteq \neg \mathit{Young}$
(4) $0.85\ ::\ \mathbf{T}(\mathit{SeniorStudent}) \sqsubseteq \mathit{Married}$

We consider eight different scenarios, representing all possible combinations of typicality inclusion, for instance $\{((2), 1), ((3), 1), ((4), 0)\}$ represents the scenario in which (2) and (3) hold, whereas (4) is not considered.The standard inclusion (1) holds in every scenario, representing a rigid property not admitting exceptions. We equip each scenario with a probability depending on those of the involved inclusions: the scenario of the example has probability $0.7 \times 0.95 \times (1 - 0.85)$, since 2 and 3 are involved, whereas 4 is not. Such probabilities are then taken into account in order to select the most adequate scenario describing the prototype of the combined concept.

Last, the logic $\mathbf{T}^{\textsf{CL}}$ exploits a method inspired by cognitive semantics [6] for the identification of a dominance effect between the concepts to be combined: for every combination, we distinguish a HEAD, representing the stronger element of the combination, and a MODIFIER. The basic idea is: given a KB and two concepts C_H (HEAD) and C_M (MODIFIER) occurring in it, we consider only *some* scenarios in order to define a revised knowledge base, enriched by typical properties of the combined concept $C \sqsubseteq C_H \sqcap C_M$.

Given a KB $\mathcal{K} = \langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ and given two concepts C_H and C_M occurring in $\mathcal{K}$, the logic $\mathbf{T}^{\textsf{CL}}$ allows defining a prototype of the combined concept C as the combination of the HEAD C_H and the MODIFIER C_M, where the typical properties of the form $\mathbf{T}(C) \sqsubseteq D$ (or, equivalently, $\mathbf{T}(C_H \sqcap C_M) \sqsubseteq D$) to ascribe to the concept C are obtained by considering blocks of scenarios with the same probability, in decreasing order starting from the highest one. We first discard all the inconsistent scenarios, then: (1) we discard those scenarios considered as *trivial*, consistently inheriting all the properties from the HEAD from the starting concepts to be combined. This choice is motivated by the challenges provided by task of commonsense conceptual combination itself: in order to generate plausible and creative compounds it is necessary to maintain a level of surprise in the combination. Thus both scenarios inheriting all the properties of the two concepts and all the properties of the HEAD are discarded since they prevent this surprise; (2) among the remaining ones, we discard those inheriting properties from the MODIFIER in conflict with properties that could be consistently inherited from the HEAD; (3) if the set of scenarios of the current block is empty, i.e. all the scenarios have been discarded either because trivial or because preferring the MODIFIER, we repeat the procedure by considering the block of scenarios, having the immediately lower probability. Remaining scenarios are those selected by the logic $\mathbf{T}^{\textsf{CL}}$.

The output of this mechanism is a knowledge base in the logic $\mathbf{T}^{\textsf{CL}}$ whose set of typicality properties is enriched by those of the compound concept C. Given a scenario w satisfying the above properties, we define the properties of C as the set of inclusions $p \ :: \ \mathbf{T}(C) \sqsubseteq D$, for all $\mathbf{T}(C) \sqsubseteq D$ that are entailed from w in the logic $\mathbf{T}^{\textsf{CL}}$. The probability p is such that: (i) if $\mathbf{T}(C_H) \sqsubseteq D$ is entailed from w, that is to say D is a property inherited either from the HEAD (or from both the HEAD and the MODIFIER), then p corresponds to the degree of belief of such inclusion of the HEAD in the initial knowledge base, i.e. $p \ : \ \mathbf{T}(C_H) \sqsubseteq D \in \mathcal{T}$; (ii) otherwise, i.e. $\mathbf{T}(C_M) \sqsubseteq D$ is entailed from w, then p corresponds to the degree of belief of such inclusion of a MODIFIER in the initial knowledge base, i.e. $p \ : \ \mathbf{T}(C_M) \sqsubseteq D \in \mathcal{T}$.

The knowledge base obtained as the result of combining concepts C_H and C_M into the compound concept C is called *C-revised* knowledge base, and it is defined as:

$$\mathcal{K}_C = \langle \mathcal{R}, \mathcal{T} \cup \{p \ : \ \mathbf{T}(C) \sqsubseteq D\}, \mathcal{A} \rangle,$$

for all D such that either $\mathbf{T}(C_H) \sqsubseteq D$ is entailed in w or $\mathbf{T}(C_M) \sqsubseteq D$ is entailed in w, and p is defined as above.

It turns out that reasoning in $\mathbf{T}^{\textsf{CL}}$ is EXPTIME-complete, namely it remains in the same complexity class of standard Description Logic $\mathcal{ALC}$ [10].

3 The Tool NERVOUS: Automated Generation of Prototypical Descriptions of Musical Genres

The tool NERVOUS is implemented in Python, with a web interface in Flutter, and it makes use of the library `owlready2` (https://pythonhosted.org/Owlready2/) for relying on the services of efficient DL reasoners (like HermiT). NERVOUS first builds a prototypical description of basic musical genres available in AllMusic, like blues, classical, country, folk, jazz, rap, pop-rock. A screenshot of the platform AllMusic is reported in Figure 1.

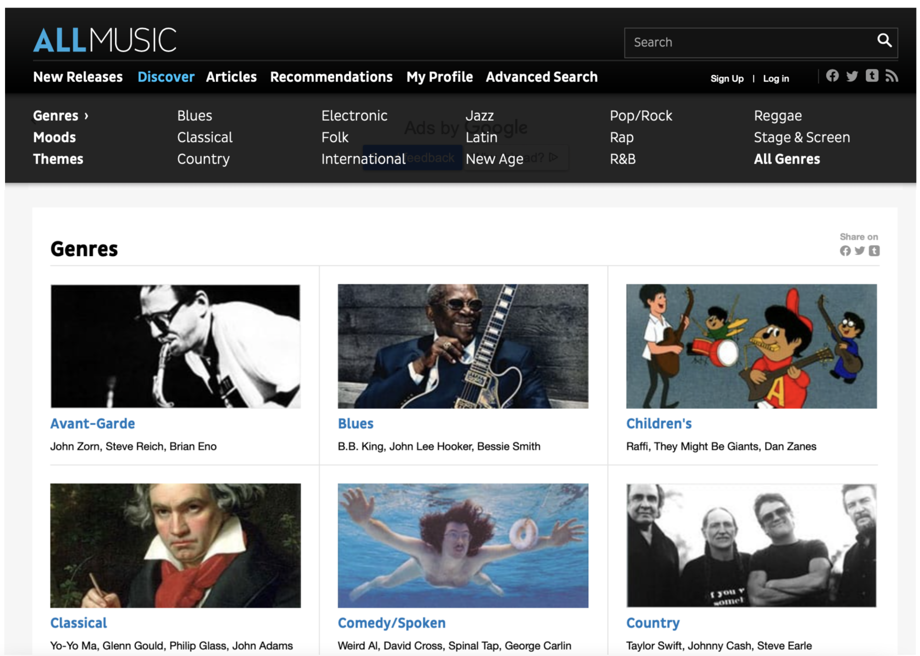

Fig. 1: A screenshot of the AllMusic platform.

To this aim, a web crawler extracts metadata from the information available on AllMusic by means of the Python library BeautifulSoap 4. More in detail, for each basic genre, the crawler extracts metadata of the 50 songs belonging to the "highlight" section for that genre; for each song the crawler then extracts a list of properties, that AllMusic calls Styles, Moods, and Themes: these properties are those that will be used to describe the prototype of genres. More in detail, for each property the system counts the number of songs having that property, then the properties are considered in a descending order and equipped by a normalized probability. As an example, consider the genre Blues and the 50 songs belonging to such a genre: the property with the highest frequency is "Regional Blues", occurring in 40 songs over 50, followed by "Earthy" (35 over 50), "Gritty" (35 over 50), "Passionate" (29 over 50) and so on. These information are used in order to provide a description of each basic genre in terms of its typical properties in the logic $\mathbf{T}^{\mathsf{CL}}$, where the frequency of a property for a genre is obtained from the number of occurrences of such property in descriptions of the songs belonging to that genre. The six properties with the highest frequency are included in the prototypical description of each basic genre, as well as the two properties with the lowest probabilities, that are added as negated properties. Formally, we have:

Definition 2. *Given a basic genre $Genre$, let $\mathcal{MI}$ be the set of songs classified in $Genre$, and let $\mathcal{S}_{Genre}$ be the set of the concepts occurring in the songs classified in that genre by AllMusic, i.e. $\mathcal{S}_{Genre} = \bigcup_{m \in \mathcal{MI}} \mathcal{S}_m$, where $\mathcal{S}_m$ is the set of properties extracted for m by the web crawler. Given a concept $Concept \in \mathcal{S}_{Genre}$, let $n_{Genre,Concept}$ the number of songs in $\mathcal{MI}$ whose description contain $Concept$, we define the frequency of a concept $Concept$ for a genre $Genre$, written $f_{Genre,Concept}$, as follows:*

$$f_{Genre,Concept} = \frac{n_{Genre,Concept}}{\mid \mathcal{MI} \mid}.$$

The prototypical description of a basic $Genre$ in the logic $\mathbf{T}^{\mathsf{CL}}$ is defined as the set of inclusions $p_1 :: \mathbf{T}(Genre) \sqsubseteq TypicalConcept_1, p_2 :: \mathbf{T}(Genre) \sqsubseteq TypicalConcept_2$, $\ldots, p_6 :: \mathbf{T}(Genre) \sqsubseteq TypicalConcept_6, \ldots$, where $TypicalConcept_1$, $TypicalConcept_2$, $\ldots$, $TypicalConcept_6$ are the six concepts in $\mathcal{S}_{Genre}$ with the highest frequencies; frequencies are then also normalized and used as degrees of belief of the respective inclusions. The two properties with the lowest frequencies are included as negated ones with a fixed probability of 0.9.

Definition 3. *Given a genre $Genre$, let the set of concepts $\mathcal{S}_{Genre}$ of Definition 2 in descending order by the frequencies $f_{Genre,Concept}$ of Definition 2:*

$$\mathcal{S}_{Genre} = \langle C_1, C_2, \ldots, C_k \rangle$$

where $f_{Genre,C_1} \geq f_{Genre,C_2} \geq \cdots \geq f_{Genre,C_k}$. The prototypical description of $Genre$ in the logic $\mathbf{T}^{\mathsf{CL}}$ is defined as the set of inclusions $f_{Genre,C_1} :: \mathbf{T}(Genre) \sqsubseteq C_1$, $f_{Genre,C_2} :: \mathbf{T}(Genre) \sqsubseteq C_2$, $\ldots$, $f_{Genre,C_6} :: \mathbf{T}(Genre) \sqsubseteq C_6$, $0.9 :: \mathbf{T}(Genre) \sqsubseteq \neg C_{k-1}$, $0.9 :: \mathbf{T}(Genre) \sqsubseteq \neg C_k$.

Observed that the least frequent concepts in blues songs are “The Creative Side” and “Day Driving”, the prototype of the genre Blues computed by the tool NERVOUS is therefore as follows:

$$
\begin{array}{l}
0.90 :: \mathbf{T}(Blues) \sqsubseteq RegionalBlues \\
0.86 :: \mathbf{T}(Blues) \sqsubseteq Earthy \\
0.86 :: \mathbf{T}(Blues) \sqsubseteq Gritty \\
0.82 :: \mathbf{T}(Blues) \sqsubseteq Passionate \\
0.82 :: \mathbf{T}(Blues) \sqsubseteq LateNight \\
0.76 :: \mathbf{T}(Blues) \sqsubseteq HangingOut \\
0.90 :: \mathbf{T}(Blues) \sqsubseteq \neg TheCreativeSide \\
0.90 :: \mathbf{T}(Blues) \sqsubseteq \neg DayDriving
\end{array}
$$

As mentioned, the logic $\mathbf{T}^{\text{CL}}$ allows one to also "manually" add rigid properties, for instance to express legal constraints, thus integrating the bottom-up, data-driven, process of prototype formation with top down expert knowledge. However, actually there is no convergence about the identification of rigid properties for describing a music genre, therefore we have chosen to avoid such properties and we have adopted typical properties only, but the reasoning mechanism provided by NERVOUS is already able to deal also with rigid properties.

4 The Tool NERVOUS: Generation of Novel Musical Genres

NERVOUS generates novel genres by combining existing ones by means of the reasoning mechanism provided by the logic $\mathbf{T}^{\text{CL}}$. Given the prototypical description of basic genres, the system NERVOUS combines two basic genres in order to build a prototype of the derived genre, by exploiting the logic $\mathbf{T}^{\text{CL}}$. To this aim, NERVOUS relies on a variant of CoCoS [14], a Python implementation of reasoning services for the logic $\mathbf{T}^{\text{CL}}$ in order to exploit efficient DLs reasoners for checking both the consistency of each generated scenario and the existence of conflicts among properties. More in detail, NERVOUS considers both the available choices for the HEAD and the MODIFIER, and it allows one to restrict its concern to a given and fixed number of inherited properties.

NERVOUS improves its ancestor, DENOTER [3], in several aspects:

- it implements a "smart" generation of scenarios, namely it beforehand discards scenarios that are inconsistent "at a glance", i.e. where P is a rigid property of the HEAD (resp. MODIFIER) and $\neg P$ is a rigid property of the MODIFIER (resp. HEAD). Moreover, similar potential conflicts among typical properties (e.g. P is a typical property of the MODIFIER whereas $\neg P$ is a typical property of the HEAD) are beforehand handled by means of suitable data structures;
- trivial scenarios are beforehand discarded;
- each scenario is equipped by a probability/score, computed by trying to assign a higher score to scenarios containing more properties, since they are considered more significant with statistic probability at hand;
- the user has the opportunity of choosing a scenario with a fixed number of properties for the combined genre rather than the one(s) with the higher probability/score.

As an example, consider the following prototype of genre Avant-garde:

$0.90 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq \neg Freedom$
$0.90 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq \neg Drammatic$
$0.90 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq TheCreativeSide$
$0.84 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq Cerebral$
$0.78 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq Uncompromising$
$0.78 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq Provocative$
$0.78 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq Revolutionary$
$0.78 \ :: \ \mathbf{T}(AvantGarde) \sqsubseteq ModernComposition$

The tool NERVOUS generates the prototype of the combined concept between Avant-garde (HEAD) and Blues (MODIFIER). The new, derived genre has the following $\mathbf{T}^{\text{CL}}$ description:

$$
\begin{array}{l}
0.90 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq \neg DayDriving \\
0.90 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq RegionalBlues \\
0.86 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq Earthy \\
0.86 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq Gritty \\
0.82 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq Passionate \\
0.82 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq LateNight \\
0.76 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq HangingOut \\
0.90 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq \neg Freedom \\
0.90 \ :: \ \mathbf{T}(AvantGarde \sqcap Blues) \sqsubseteq \neg Dramatic
\end{array}
$$

Rigid properties of basic concepts, if any, would be inherited by the derived concept too.

5 The Tool NERVOUS: Reclassifications and Recommendations

The tool NERVOUS is also able to reclassify songs of AllMusic within the novel derived genres. As mentioned, each song is equipped by some information available in AllMusic: NERVOUS extracts such information and then it computes a score, in order to compare them with the properties of a derived genre. Scores are provided by the portal users, who can mark each property of each song. All the metadata extracted by the crawler are stored in a JSON file.

Given a song m and a genre $Genre$, for each property C of m itself, the tool NERVOUS computes a score of compatibility of m with respect to $Genre$: intuitively, this is obtained as the combination of the frequencies of "compatible" concepts, i.e. concepts belonging to both the song and the prototypical description of the genre. More in detail, NERVOUS checks whether the property C is either a rigid or a typical property of the prototype of $Genre$: if this is the case, the score of the song is incremented by a positive number (1 in case of a rigid property, otherwise the product of the probability $f_{Genre,C}$ and the score of C in m). If $\neg C$ belongs to the typical properties of $Genre$, similarly the score is updated by a negative number. NERVOUS re-classifies the song m in the novel genre $Genre$ if the score so obtained is higher than 0, then it suggests the set of classified contents, in a descending order of compatibility.

Definition 4. *Given a song m, let $DerivedGenre$ be a derived genre as defined in Section 3 and let $\mathcal{S}_m$ be the set of properties occurring in the description of m and, given a concept $C \in \mathcal{S}_m$, let s_{m_C} be the score of C in the description of m. We define the score (rank) of compatibility as $r = \sum_{C \in \mathcal{S}_m} \theta_{C_{m,Genre}}$, where:*

- $\theta_{C_{m,Genre}} = p \times s_{m_C}$, *if* $p \ :: \ \mathbf{T}(Genre) \sqsubseteq C \in \mathcal{K}_C$
- $\theta_{C_{m,Genre}} = 1$, *if* $Genre \sqsubseteq C \in \mathcal{K}_C$
- $\theta_{C_{m,Genre}} = -p \times s_{m_C}$, *if* $p \ :: \ \mathbf{T}(Genre) \sqsubseteq \neg C \in \mathcal{K}_C$
- $\theta_{C_{m,Genre}} = -999$, *if* $Genre \sqsubseteq \neg C \in \mathcal{K}_C$

As an example, consider the above derived genre $AvantGarde \sqcap Blues$, and the song "The Things That I Used to Do" by Guitar Slim. It is reclassified in the novel, generated genre $AvantGarde \sqcap Blues$, since its score is ... > 0. This song will be then recommended by NERVOUS, as it can be seen in Figure 2, where a picture of NERVOUS's

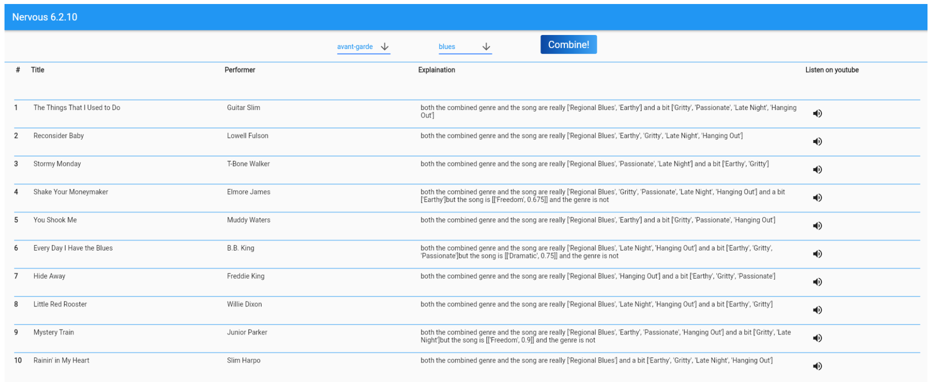

Fig. 2: A screenshot of the interface of NERVOUS.

interface is shown. It is worth noticing that, in order to provide a "white-box" recommender system, each recommended song is equipped by an explanation, relying on the pipeline implemented by system of concept combination.

Let us conclude this section by observing that the fact that a recommended song belongs to both original, basic genres that have been combined is far from being obvious: indeed, the system NERVOUS suggests also the song "Moanin" by Art Blakey & the Jazz Messengers, which is classified by AllMusic as belonging to the genre Jazz. In our opinion, this is a further interesting mechanism providing the required component of surprise in the recommendation, justified by the fact that the description of the song matches the one of the novel genre, the last one only partially inheriting properties from the basic genres whose combination lead to such a new genre.

The tool NERVOUS is available at https://github.com/Mattia98779/Nervous. A preliminary version of a web interface is available at https://mattia98779.github.io/#/: by means of such a web interface, a user can select two basic genres and then obtain the list of suggested songs, together with an explanation.

6 Evaluation and Discussion

In this section we provide a preliminary evaluation of our tool NERVOUS. We have tested it in two different ways. The first evaluation is completely automatic and inheres the capability of the system of generating novel hybrid genres that are able to be populated by the original content of the AllMusic platform via a re-classification mechanism involving the 599 songs of the platform. In this case, the success criterion concerns the avoidance of the creation of empty boxes corresponding to the new generated combined genres. More in detail, at least 69 songs are re-classified by the tool NERVOUS for each derived music genre (the second genre containing "few" songs contains 138 items), with an average of 307 songs per derived genre. This is summarized in Figure 3, picture in the left, whereas from the picture on the right we can observe that only 7 out of 599 songs on AllMusic (with very few attributes) are not re-classified in any genre by the system, whereas all the other ones (98.83%) are re-classified in at least one genre.

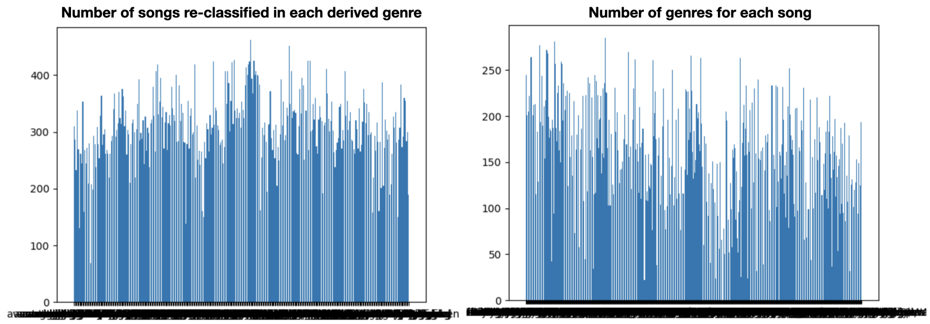

Fig. 3: Some statistics about the re-classification of NERVOUS.

The second evaluation consisted in a user study involving 22 persons (11 females, 11 males, aged 14-72) that evaluated a total of 260 recommendations generated by the system. It is worth observing that this is one of the most commonly used methodology for the evaluation of recommender systems based on controlled small groups analysis [22]. The idea was to estimate the satisfaction of the potential users of the platform when exposed to the contents of the novel categories suggested by NERVOUS: all the participants were voluntary people using an availability sampling strategy. Participants were all naive to the experimental procedure and to the aims of the study. This evaluation was carried out as a classical "*one to one*" lab controlled experiment (i.e. one person at time with one expert interviewer) and we adopted a thinking aloud protocol, consisting in recording the verbal explanations provided by the people while executing a given laboratory task [16,17]. In this setting, the users had to start the interview by indicating a couple of preferred genres among those available in AllMusic. This selection triggered both the activation of a novel hybrid prototypical genre by NERVOUS and the corresponding reclassification of the AllMusic songs based on such selection. The output of the system, pruned to show the top 10 best results, was then evaluated with a 1-10 voting scale expressing the satisfaction of the received recommendations.

The results we have obtained seem promising: the average score assigned by the users to the recommendations of the reclassified elements is 7.44 out of 10. This score was calculated by considering, for each new category, the score assigned to the top 10 reclassified songs, since they were provided, to the users, as recommendations for the novel genres.

It is worth observing that, in few cases, the creative classification performed by the tool NERVOUS has lead to counter-intuitive results. As an example, the song "I'm eighteen" by Alice Cooper, known as "The Godfather of Shock Rock", is classified as belonging to the derived genre result of the combination between Rap and Avant-garde. We strongly conjecture that these situations could be easily avoided by introducing constraints on some genres by means of rigid negated properties.

Furthermore, most of the people we have interviewed observed that AllMusic adopts a debatable choice of basic genres, in particular concerning the fact that Pop and Rock, two of the most popular music genres in the world, are grouped in a single category. This immediately implies some difficulties in combining its prototype with the one of another basic genre. Moreover, some of the (low ranked) items corresponded to old songs. This follows immediately from the fact that few recent songs belong to the highlights of AllMusic, since they have received a lower number of scores by the portal's users.

Notably the first two of the above mentioned issues are not directly related to NERVOUS, since: i) the system can not know if the association description/item is coherent, but it just provides (for the recommended output) the correspondence already in place in AllMusic; ii) the recommendations of old editorial contents is based on the actual dataset of AllMusic (collecting about six hundred songs). This element can be overcome by simply adding an additional filter about the period preferences of the users.

7 Conclusions and Future Works

In this work we have presented NERVOUS, a knowledge-based system for the dynamic generation of novel contents about music, exploiting the reasoning mechanism of the logic $\mathbf{T}^{\mathsf{CL}}$ in order to generate, reclassify and suggest novel content genres in the context of AllMusic, an online platform collecting in-depth information about music genres, albums, musicians and songs. The core component of the system NERVOUS relies on CoCoS, a tool for combining concepts in the logic $\mathbf{T}^{\mathsf{CL}}$.

According to [23] recommender systems "try to identify the need and preferences of users, filter the huge collection of data accordingly and present the best suited option before the users by using some well-defined mechanism". The literature is rich of proposals, that we can partition in three main groups of recommender systems:

- collaborative filtering, which exploits similarities of usage patterns among mutually similar users;
- content-based filtering, which exploits content similarity;
- hybrid filtering, which combines the two approaches.

It is easy to observe that the tool NERVOUS could be considered an hybrid recommender system, since in its current form it makes use of content description as the input. However, it differs from the state of the art approaches since it exploits the reasoning power of a logic framework capable of representing new intuitive principles influencing user preferences and usage attitudes which cannot be derived from the pure analysis of content and/or the comparison of similar users.

The system NERVOUS has been tested in a twofold evaluation showing promising results for both the automatic evaluation and the user acceptability of the recommended items. With evaluation results at hand, we can observe that NERVOUS represents a good approach at addressing the very well known filter bubble effect [19], since it introduces mechanisms that add a sort of "plausible creativity" and a "reasonable serendipity" in content discovery by users.

In future research, we aim at extending our work in several directions. On the one hand, we aim at studying the application of optimization techniques in [1] in order to improve the efficiency of CoCoS and, as a consequence, of the proposed knowledge generation system. On the other hand, we aim at conducting a large scale experiment to further validate the effectiveness of the proposed approach, including people with sensory impairments, with the objective of promoting empathy, cohesion and inclusion across social groups, partially neglected by state-of-the-art recommender systems.

References

1. Alberti, M., Bellodi, E., Cota, G., Riguzzi, F., Zese, R.: cplint on SWISH: probabilistic logical inference with a web browser. Intelligenza Artificiale **11**(1), 47–64 (2017). https://doi.org/10.3233/IA-170106, https://doi.org/10.3233/IA-170106
2. Boden, M.A.: Creativity and artificial intelligence. Artificial Intelligence **103**(1-2), 347–356 (1998)
3. Chiodino, E., Di Luccio, D., Lieto, A., Messina, A., Pozzato, G.L., Rubinetti, D.: A knowledge-based system for the dynamic generation and classification of novel contents in multimedia broadcasting. In: De Giacomo, G., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (eds.) ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020. Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 680–687. IOS Press (2020). https://doi.org/10.3233/FAIA200154, https://doi.org/10.3233/FAIA200154
4. Frixione, M., Lieto, A.: Representing and reasoning on typicality in formal ontologies. In: Ghidini, C., Ngomo, A.N., Lindstaedt, S.N., Pellegrini, T. (eds.) Proceedings of the 7th International Conference on Semantic Systems. pp. 119–125. ACM International Conference Proceeding Series, ACM (2011). https://doi.org/10.1145/2063518.2063534, https://doi.org/10.1145/2063518.2063534
5. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: Semantic characterization of Rational Closure: from Propositional Logic to Description Logics. Artificial Intelligence **226**, 1–33 (2015). https://doi.org/10.1016/j.artint.2015.05.001
6. Hampton, J.A.: Inheritance of attributes in natural concept conjunctions. Memory & Cognition **15**(1), 55–71 (1987)
7. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence **55**(1), 1–60 (1992). https://doi.org/http://dx.doi.org/10.1016/0004-3702(92)90041-U
8. Lieto, A., Perrone, F., Pozzato, G.L., Chiodino, E.: Beyond subgoaling: A dynamic knowledge generation framework for creative problem solving in cognitive architectures. Cognitive Systems Research **58**, 305–316 (2019). https://doi.org/10.1016/j.cogsys.2019.08.005, https://doi.org/10.1016/j.cogsys.2019.08.005
9. Lieto, A., Pozzato, G.L.: A description logic of typicality for conceptual combination. In: Ceci, M., Japkowicz, N., Liu, J., Papadopoulos, G.A., Ras, Z.W. (eds.) Foundations of Intelligent Systems - 24th International Symposium, ISMIS 2018, Limassol, Cyprus, October 29-31, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11177, pp. 189–199. Springer (2018). https://doi.org/10.1007/978-3-030-01851-1_19, https://doi.org/10.1007/978-3-030-01851-1_19
10. Lieto, A., Pozzato, G.L.: A description logic framework for commonsense conceptual combination integrating typicality, probabilities and cognitive heuristics. Journal of Experimental and Theoretical Artificial Intelligence **32**(5), 769–804 (2020). https://doi.org/10.1080/0952813X.2019.1672799, https://doi.org/10.1080/0952813X.2019.1672799
11. Lieto, A., Pozzato, G.L., Perrone, F.: A dynamic knowledge generation system for cognitive agents. In: 31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019, Portland, OR, USA, November 4-6, 2019. pp. 676–681. IEEE (2019). https://doi.org/10.1109/ICTAI.2019.00099, https://doi.org/10.1109/ICTAI.2019.00099
12. Lieto, A., Pozzato, G.L., Perrone, F., Chiodino, E.: Knowledge capturing via conceptual reframing: A goal-oriented framework for knowledge invention. In: Proceedings of the 10th ACM Conference on Knowledge Capture, K-CAP 2019, Marina del Rey. pp. 109–114. ACM (2019)
13. Lieto, A., Pozzato, G.L., Striani, M., Zoia, S., Damiano, R.: Degari 2.0: A diversity-seeking, explainable, and affective art recommender for social inclusion. Cognitive Systems Research

77, 1–17 (2023). https://doi.org/https://doi.org/10.1016/j.cogsys.2022.10.001, https://www.sciencedirect.com/science/article/pii/S1389041722000456

14. Lieto, A., Pozzato, G.L., Valese, A.: COCOS: a typicality based concept combination system. In: Felli, P., Montali, M. (eds.) Proceedings of the 33rd Italian Conference on Computational Logic, Bolzano, Italy, September 20-22, 2018. CEUR Workshop Proceedings, vol. 2214, pp. 55–59. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2214/paper6.pdf
15. Lieto, A., Pozzato, G.L., Zoia, S., Patti, V., Damiano, R.: A commonsense reasoning framework for explanatory emotion attribution, generation and re-classification. Knowledge-Based Systems **227**, 107166 (2021)
16. Newell, A., Shaw, J.C., Simon, H.A.: Report on a general problem solving program. In: IFIP congress. vol. 256, p. 64. Pittsburgh, PA (1959)
17. Newell, A., Simon, H.A.: Human problem solving, vol. 104, n. 9. Prentice-Hall Englewood Cliffs, NJ (1972)
18. Osherson, D.N., Smith, E.E.: On the adequacy of prototype theory as a theory of concepts. Cognition **9**(1), 35–58 (1981)
19. Parisier, E.: The Filter Bubble: What the Internet Is Hiding from You (2012)
20. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R.: Probabilistic description logics under the distribution semantics. Semantic Web **6**(5), 477–501 (2015). https://doi.org/10.3233/SW-140154, http://dx.doi.org/10.3233/SW-140154
21. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R.: Reasoning with probabilistic ontologies. In: Yang, Q., Wooldridge, M. (eds.) Proceedings of IJCAI 2015. pp. 4310–4316. AAAI Press (2015), http://ijcai.org/proceedings/2015
22. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender systems handbook, pp. 257–297. Springer (2011)
23. Sohail, S.S., Siddiqui, J., Ali, R.: Classifications of recommender systems: A review. Engineering Science and Technology Review **10**(4), 132–153 (2017)