information

Article

Towards Automated Semantic Explainability of Multimedia Feature Graphs

Stefan Wagenpfeil [1,*], Paul Mc Kevitt [2], and Matthias Hemmje [1]

[1] Faculty of Mathematics and Computer Science, University of Hagen, Universitätsstrasse 1, D-58097 Hagen, Germany; felix.engel@fernuni-hagen.de (F.E.); Matthias.hemmje@fernuni-hagen.de (M.H.)
[2] Academy for International Science & Research (AISR), Derry/Londonderry, Ireland; p.mckevitt@aisr.org.uk
* Correspondence: stefan.wagenpfeil@fernuni-hagen.de

Abstract: Multimedia feature graphs are employed to represent features of images, video, audio, or text. Various techniques exist to extract such features from multimedia objects. In this paper, we describe the extension of such a feature graph to represent the *meaning* of such multimedia features and introduce a formal context-free PS-grammar (Phrase Structure grammar) to automatically generate human-understandable natural language expressions based on such features. To achieve this, we define a semantic extension to syntactic multimedia feature graphs and introduce a set of production rules for phrases of natural language English expressions. This explainability, which is founded on a semantic model provides the opportunity to represent any multimedia feature in a human-readable and human-understandable form, which largely closes the gap between the technical representation of such features and their semantics. We show, how this explainability can be formally defined and demonstrate the corresponding implementation based on our generic multimedia analysis framework. Furthermore, we show how this semantic extension can be employed to increase the effectiveness in precision and recall experiments.

Keywords: indexing, retrieval, explainability, semantic, multimedia, feature graph, graph code

check for updates

Citation: Wagenpfeil, S.; Mc Kevitt, P.; Hemmje, M. Towards Automated Semantic Explainability of Multimedia Feature Graphs. *Information* **2021**, *1*, 0. https://doi.org/

Received: 15 December 2020
Accepted: 15 January 2021
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction and Motivation

Bridging the semantic gap has been a research goal for many years. Narrowing down the gap between detected features from multimedia assets (i.e., images, video, audio, text, and Social Media) and their semantic representation has led to numerous investigations and research in the field of Multimedia Information Retrieval (MMIR) [1]. With the extraction of MMIR supporting features and the integration of these features into data models, internal representations of these features are created. MMIR applications process these representations e.g., w.r.t. indexing, retrieval, and querying and employ them to analyze relationships between or within assets. In particular, the topic of querying becomes highly important, as e.g., every single minute, 147.000 photos are uploaded to Facebook, 41.6 million Whatsapp messages are sent, or 347.000 stories are posted by Instagram [2] and the users still expect highly accurate query results. Due to higher resolutions of images and video, the Level-Of-Detail (LOD) in Multimedia assets has increased significantly. Current professional cameras like the *Sony α7R IV 35* are equipped with a resolution of 61.0 Megapixel [3], and Smartphones like the *Xiaomi Redmi Note 10 Pro* even push that boundary to 108 Megapixel [4]. This high LOD is also reflected by other Multimedia types, e.g., text, where News agencies maintain huge archives of textual information, enriched by user comments, web information, or social media [5].

The figures given above demonstrate that it becomes more and more important to semantically *understand* MMIR supporting features to increase the precision and recall of quasi-semantic MMIR querying. The Semantic Web [6] and all its related technologies and concepts are a sound basis for knowledge representation, reasoning, inferencing,

and truth maintenance. To bridge the gap between detected features and their semantic representation, a machine-readable and formal approach is required, as well as a human-understandable explanation of the corresponding processing steps. However, the requirement of a high LOD, the increasing number of assets, and the demand for fast and accurate semantic querying contradict each other and lead to further challenges in the area of explainability in a human-understandable way.

In this paper, we present a solution for the automated explainability of MMIR processing steps in the form of human-understandable natural language texts based on a semantic modeling, which also supports inferencing and reasoning.

2. State of the Art and Related Work

This section gives an overview of current techniques and standards for semantic indexing and retrieval and discusses related work. We introduce the *Multimedia Feature Graph (MMFG)* and the *Graph Code* concept, also discussing semantic analysis and intelligent information retrieval methods.

In previous work, we already introduced the *Generic Multimedia Analysis Framework (GMAF)* [7][8] [9][10] as a unifying framework that can integrate various Multimedia features into a single data structure. The GMAF utilizes selected existing technologies as plugins to support various Multimedia feature detection algorithms for text (e.g., *social media posts, descriptions, tag lines*) [11][12][13], images (especially object detection and spatial relationships including the use of machine learning) [14][15][11][16][11], audio (transcribed to text) [17] [18] [15], and video including metadata [19] and detected features [20][21][18].

The GMAF produces a *Multimedia Feature Graph (MMFG)*, which is defined in [7] and represents various integrated Multimedia features. Within an application, these MMFGs are typically represented as a collection. For the remainder of this paper, some internal structures, sets, and definitions of the MMFG are relevant (see also [22][7]):

- $FVT_{MMFG} = \{ft_1, ..., ft_n\}$ is the feature term vocabulary, i.e., the set of feature term labels, which represent detected features. Elements of FVT_{MMFG} are represented by $Nodes(n)$ in the MMFG graph structure.
- $FVT_{Coll} = \bigcup_{i=1}^{n} FVT_{MMFG_i}$ is the feature term vocabulary of the collection of MMFGs within a MMIR application.
- the set $FRT_{MMFG} = \{cn, s, sr\}$ representing the feature relationship types of a MMFG, where *cn* represents the "child" relationship, *s* represents the "synonym" relationship, and "sr" represents the "spacial" relationship between feature vocabulary terms. Elements of FRT_{MMFG} are represented by links between *Nodes* in the MMFG graph structure.

As the integration of Multimedia features within MMFGs produce a much higher level-of-detail (LOD), effective and efficient algorithms are required to process these feature graphs. Hence, we introduced *Graph Codes*, which are a 2D projection of MMFGs and corresponding algorithms, performing calculations based on matrices instead of graph traversal and also support the higher LOD of MMFGs [7]. For these *Graph Codes*, we introduced a metric for similarity calculation, the mathematical concepts of the indexing and retrieval algorithms, including a detailed evaluation regarding performance, precision, and recall. Figure 1 briefly summarizes this concept as a foundation for subsequent sections of this paper. Figure 1(a) shows a snippet of an exemplary MMFG visualized in a graph editing tool [23]. Figure 1(b) illustrates a part of the MMFG in an object diagram including various node and relationship types, which can be represented as a *Graph Code* table (see Figure 1(c)) based on the graph's valuation matrix. A *Graph Code's* matrix representation is shown in Figure 1(d), where the correspondence to mathematical matrix calculations is obvious[1].

[1] It is notable for Multimedia, that due to the current object detection algorithms, MMFGs and their corresponding *Graph Codes* contain semantic information (e.g., "is a"), spatial information (e.g., "above"), and also temporal information (by the temporal ordering of sub-collections of *Graph Codes*).

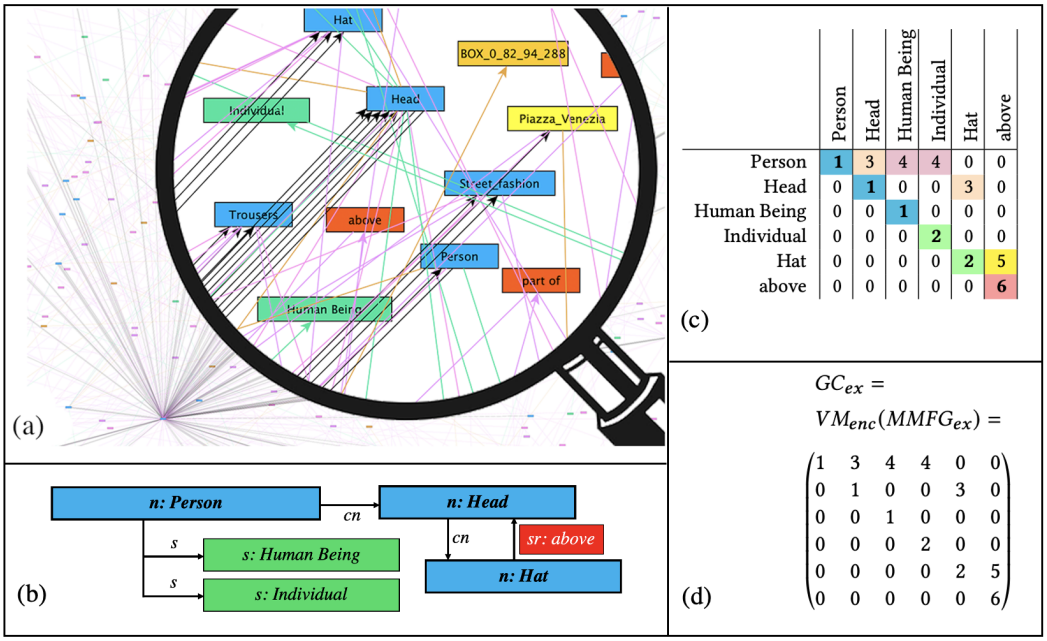

Figure 1. Exemplary *MMFG* and its related representations.

To calculate MMIR results based on *Graph Codes*, a *Graph Code Metric* is defined, which can be applied for similarity algorithms. In general, every detected feature can be regarded as a Multimedia indexing term. The indexing term of any relevant feature thus becomes part of the vocabulary of the overall retrieval index. In Multimedia Indexing and Retrieval (MMIR), typically these terms have structural and/or semantic relationships to each other and represent the basis for semantic query construction and result representation. In [7], we already defined a metric for similarity of *Graph Codes*, which is a triple $M_{GC} = (M_F, M_{FR}, M_{RT})$ containing a *feature-metric* M_F based on the vocabulary, a *feature-relationship-metric* M_{FR} based on the possible relationships, and a *feature-relationship-type-metric* M_{RT} based on the actual relationship types. This metric can be applied for result representation and has to be considered when constructing corresponding queries. Querying and result presentation based on *Graph Codes* is discussed in [24].

Current *Graph Code Query Construction* technologies employ structured query languages (e.g., SQL, OQL, XML-Query), including Visual query languages (VQLs) and natural language querying (NL) [25]. A Meaning Driven Data Query Language (MDDQL) [25] [26], can support query construction by system-made suggestions of natural language based terms. In the field of Natural Language Processing (NLP), there have been several approaches to automatically translate natural language into structured queries, e.g., NLP to SPARQL processing [27] [28]. Typically, results of this kind of queries are represented in the form of ranked lists. All these query construction methodologies require semantic modeling.

Semantic Representation is covered by the concepts and standards defined by the Semantic Web [29], where the manual, semiautomatic, and automated generation of annotations is defined. The basis for these representations and annotations is a set of domain-oriented vocabulary terms. Once a basic "subclass" relationship is introduced between vocabulary terms, taxonomies can be built, which structure these terms in the form of class / subclass relationships. Typically, taxonomies also contain a consistent set of predefined textual labels and synonyms. Thesauri can be used to model further relationships (e.g. "broader", "narrower") for additional structuring, scoping, and increased expressiveness. Resource Description Framework (RDF) [6] can serve as a foundation. It covers the description of any kind of resource by employing XML Syntax. The Resource Description Framework Schema (RDFS) provides domain specific extension points and a standardized model of exchanging RDF documents. As RDF is based on XML, it can automatically be represented in the form

of a graph model, which provides the opportunity to employ a mapping to the MMFG on a structural level. RDF-techniques like publishing or linking, with a shared data model can act as a base layer for other technologies [6] [30] [31] [29]. Finally, ontologies describe arbitrary relationships between taxonomy terms (now called concepts) going beyond the hierarchical taxonomy structure. OWL [29] represents these concepts and relationships as classes and properties [32]. Once such a well-defined formal semantic model is in place, also reasoning and inference algorithms can be applied to such semantic representations [33].

But to clearly define, how the concepts of an ontology can be combined automatically (e.g., during automatic inferencing), a *Well Defined Grammar* [34] is required. Based on such grammars, an algorithmic implementation can distinguish between valid and invalid statements of a given formal or natural language. According to [34], a grammar $G = (V, T, P, S)$ for a language L is defined by the tuple of vocabulary terms V, the list of terminal symbols T, which terminate valid sentences of L, production rules P, which describe valid combinations of vocabulary terms and a set of starting symbols S for sentences of L.

In [36], *PS-grammars (Phrase Structure grammars)* are employed as a specialized form to generate language terms by production rules, in which the left side of the rule is replaced by the right side. If, e.g., $\alpha \rightarrow \beta$ is a production rule in P, and ϕ, ρ are literals in V, then $\phi\alpha\rho \rightarrow \phi\beta\rho$ is a direct replacement. Rules of *PS-Grammars (PSG)* are further detailed by four types, *0: unlimited PS-rule, 1: context sensitive PS-rule, 2: context-free PS-rule, 3: regular PS-rule*, which denote systematic restrictions of the production rules. These restrictions lead to a hierarchy of formal languages and the corresponding calculation and validation complexities (see Table 1).

restriction	PSG class	language class	complexity
type 3	regular PSG	regular language	linear
type 2	context free PSG	context free language	polynomial
type 1	context sensitive PSG	context sensitive language	exponential
type 0	unrestricted PSG	recursive enumerable language	unsolvable

Table 1: PS-grammar (Phrase Structure grammar) hierarchy of formal languages according to [36].

Typically, when defining grammars, the set V will contain additional classes to structure the possible production rules (typically defined as Chomsky rules [34]), e.g., classes to describe *Nominal Phrases (NP), Verbal Phrases (VP), Prepositional Phrases (PP)*, or other word types like *Adjectives (ADJ)*, and their location in validly produced sentences [36]. In many cases, grammars are designed, that $V \cap T = \emptyset$. As an example, the sentence "The hat is above the head" can be represented by the context-free grammar $G_{en} = (V_{en}, T_{en}, P_{en}, S_{en})$ for simple English sentences:

- $V_{en} = \{S_{en}, NP, VP, V, N, DET, PR\}$ representing the variables of the grammar
- $T_{en} = \{the, hat, is, above, head\}$ is the set of terminal symbols, with $V \cap T = \emptyset$
- the production rules for this grammar can be defined as follows:

$$\begin{aligned} P_{en} = \{&S_{en} \rightarrow NP\ VP, \\ &VP \rightarrow V\ PP, \\ &NP \rightarrow DET\ N, \\ &PP \rightarrow PR\ NP\} \end{aligned}$$

The production rules of $PS - Grammars$ can be visualized in form of so called *PS-trees* [36]. Figure 2 shows such a PS-tree with the sentence, "The hat is above the head" based on the exemplary MMFG of Figure 1(b). In this example, the use of *Nominal Phrases (NP), Verbal Phrases (VP), Prepositional Phrases (PP),* and the *Start Symbol S* are also illustrated. For example:

- the NP, "The hat", consists of the determiner "The", and the noun "hat"
- the NP, "the head", is built by the determiner "the", and the noun "head"
- the PP, "above the head", is constructed by the preposition "above", and the NP "the head"
- the VP, "is above the head", consists of the verb "is", and the PP "above the head"
- the starting symbol for this sentence is constructed by a NP, "The hat" and the VP, "is above the head"

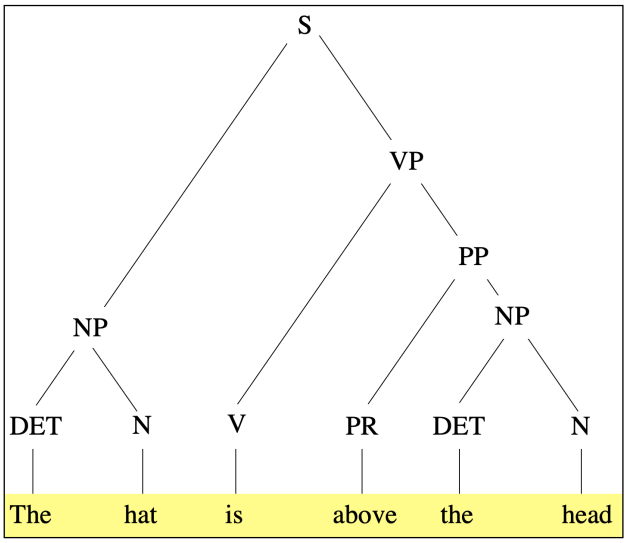

Figure 2. Production rules for the example "The hat is above the head" visualized as Phrase-Structure tree (PS-tree).

By applying these production rules, both construction and the analysis of well-formed sentences can be approved. The introduction of a grammar is also a prerequisite for reasoning [36], where additional semantic features are generated by calculating inferences and conclusions [35]. A formal grammar for MMFGs and *Graph Codes* and the corresponding concepts based on this grammar, is defined in section 3 (modeling) and will be the basis for constructing Semantic MMFGs (SMMFG).

To manage and maintain such semantic information based on MMFGs, several more technical *Knowledge Representation and Processing Systems* have been specified and introduced. For example, the W3C introduced the *Simple Knowledge Organization System (SKOS)* [37] as a standard way to represent knowledge organization systems with RDF [6]. Reasoning (i.e., the automated extension and maintenance of information), Truth Maintenance (i.e., the automated calculation of information validity), and Inference Systems (i.e., deducting new information based on logical rules) also contribute to improving the semantic information of a MMIR collection [33][38][39].

One common approach for *Automated Reasoning and Inferencing* is the concept of *Non-monotonic Reasoning* [38], which is based on *justified beliefs* and *reasonable assumptions*. Typically, so called *Default Logics* are employed to represent knowledge in form of rules. For example, the rule $A : b/C$ is intended to state that, "if A is believed, and each b $\in$ B can be consistently believed, then C should be believed". A is a *prerequisite*, b a set of *justifications* [38][33]. For the calculation and representation of *Default Logic*, two major approaches serve as a foundation and are named after their inventors Reiter [40] and Poole [41]. Both approaches result in the common concept of Knowledge Extensions, which represent the set of possible rules, which are assumed (or calculated) to be believed.

Semantic Querying and Retrieval can be performed by employing, e.g., SPARQL [42], which is a standardized query language, that operates on RDF, RDFS, or OWL representations and also supports the inclusion of semantic features. Semantic reasoning, which is applied to the underlying semantic representation of an ontology, is often called, "intel-

ligent retrieval". This means, that automatic reasoning can derive new semantic feature annotations from existing feature representations. This further means, that newly derived features have not been detected, but are actually derived by means of reasoning. Thus, the Multimedia object is annotated with additional features, that are not a result of extraction or detection, but derived by logical and semantic reasoning. The resulting feature index is extended and retrieval results will be more accurate where additional features will be attributed to MMFGs automatically [35]. In the remainder of this paper, we will call the originally detected or extracted features, *Internal Feature (F_{Int})*, and the features that have been derived by reasoning *External Features (F_{Ext})*.

However, to follow this systematic approach, typically a substantial manual effort is required to map syntactic feature representations to semantic representations. Also the introduction of rules, a basic logic, and truth maintenance criteria must be performed manually. Hence, in section 3 we define a formal, standardized, and automated approach for the integration of these systems.

In summary, we can state that current technologies provide a sufficient set of appropriate algorithms, tools, and concepts for semantic modeling, representation, indexing and retrieval. However, concepts like RDF, RDFS, OWL, SPARQL, or the Semantic Web mostly rely on graph-based semantic representation structures and thus underlie similar constraints regarding efficiency as syntactic feature graphs. The introduction of grammars provides a standardized way of constructing and analyzing well formed sentences. As *Graph Codes* provide a set of algorithms to significantly increase effectiveness, LOD, and efficiency for graph-based IR algorithms, we now present their application and extension into *Semantic Graph Codes* and the corresponding processing algorithms.

3. Modeling and Design

In this section, we define and introduce several semantic extensions of syntactic MMFGs, which serve as a basis for optimized *Semantic Graph Code* processing and the corresponding application of semantic concepts such as Annotation, Topic Modeling, Reasoning, or Inferencing. We also introduce *Explainable Semantic Graph Codes*, providing a human-readable representation of multimedia feature graphs.

The MMFG has been designed to represent MMIR features on a pure syntactical basis, containing only, *Internal Features* F_{Int}. To support semantic extensions for MMFGs, we apply Semantic Web concepts [29]. In addition, we introduce a context-free PS-grammar G_{MMFG} for the construction of human-readable, i.e., valid natural language textual phrases and statements based on MMFG features. This enables the construction of a formal semantic representation on the one hand, and establishes the basis for natural language textual explanations on the other hand. This combination leads to a well-defined semantic representation of MMFGs, *Semantic Multimedia Feature Graph (SMMFG)*, and to the *Explainable Semantic Multimedia Feature Graph (ESMMFG)*. In particular, ESMMFGs can serve as a basis for inferencing and thus support the production of additional *External Features* F_{Ext}. The employment of a PS-grammar in addition to the semantic extension has the advantage, that the representation of such a MMFG is not only machine-readable, but also "human-readable", i.e., the representation supports transparency, explainability, and reproducibility for humans.

The structure of this section follows the logical sequence of extensions from simple MMFGs to semantic SMMFGs and explainable ESMMFGs. Hence, in subsection 3.1, we discuss the initial formal foundation of the chosen approach. In subsection 3.2, annotations for MMFGs are introduced, which are then employed to define the semantic extension in the form of SMMFGs in subsection 3.3. Finally, explainability is introduced by the definition of a PS-grammar resulting in ESMMFGs in subsection 3.4.

3.1. Formal Representation of a MMFG

As shown in the state of the art discussion (see section 2), MMFGs are purely syntactical structures based on a data model, which forms a multimedia feature graph with nodes and edges. The formal model of such a syntactic MMFG representation is shown in Figure 3(a). Here, and in subsequent sections, we further detail and extend the exemplary MMFG from section 2, which serves as a exemplary syntactic representation of a simple MMFG (see Figure 3(b)). To support the formal representation of such a MMFG with a formal language (i.e., grammar), an additional element - the root node S_{MMFG} - must be defined, which acts as a starting symbol for valid formal language expressions (see section 1). This is shown in Figure 3(b).

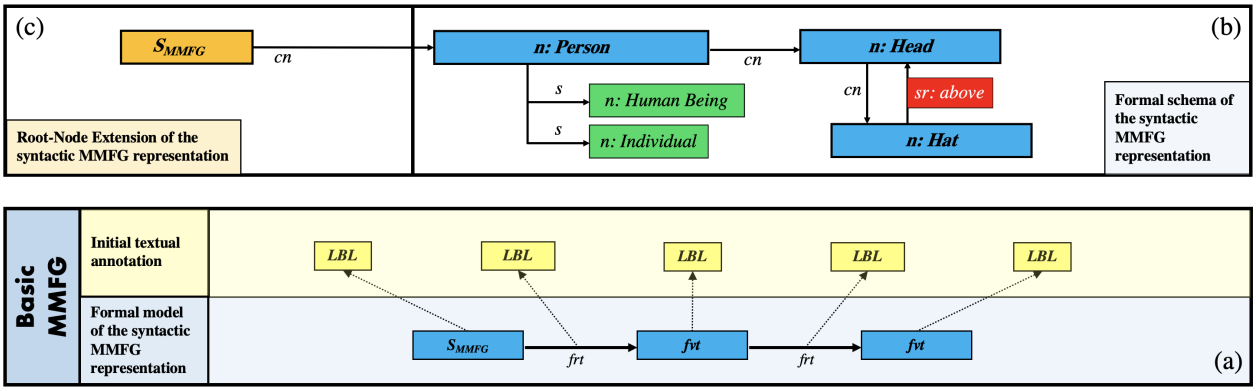

Figure 3. MMFG overview: (a) formal model of the syntactic MMFG representation, (b) formal syntactic schema of the MMFG representation of $MMFG_{ex}$. (c) additional starting symbol S_{MMFG}.

A basic grammar $G_{MMFG} = (V_{MMFG}, T_{MMFG}, P_{MMFG}, S_{MMFG})$ for the construction of valid MMFG-sentences (see section 2) can be formalized as

- $V_{MMFG} = FRT \cup FVT$ is the set of feature relationship terms FRT and the set of feature vocabulary terms (FVT).
- T_{MMFG} is a set of textual labels (LBL) for elements in FRT and FVT
- P_{MMFG} is the set of production rules, which produce sentences based on, FRT and FVT. In its simplest form, P can be defined as:

$$\begin{aligned} P = \{ \\ S \rightarrow FVT\ FRT\ FVT \\ FVT \rightarrow LBL \\ FRT \rightarrow LBL \\ \} \end{aligned}$$

- S_{MFG} is the root node of an MMFG

When such an initial grammar is applied to the exemplary MMFG of Figure 3(c), formal language expressions like, "person has-child head. head has-child hat. hat is-above head.", can be produced based on feature vocabulary terms and feature relationship terms. All these sentences are built on the pattern, *node - relationship - node*. Although this initial grammar could be employed for a basic representation of syntactical MMFGs, further extensions must obviously be constructed, particularly in respect of supporting higher level semantics and improving human readability. This initial grammar leaves the following open challenges:

- the initial grammar does not yet represent the syntactic structure of MMFGs
- the initial grammar is not a context-free PS-grammar
- the initial grammar for MMFGs does not reflect the structural elements of MMFGs and their corresponding production rules. For example, the structural element *cn*

(i.e., child node) should be transformed into the grammatical structure, "has a child named", represented by several textual labels. Another example would be the spacial relationship, $sr : above$, which should be represented by a set of textual labels forming the phrase, "is above of".

- the initial grammar does not provide a semantic description of the syntactic relationships.

However, this initial definition illustrates the overall approach of representing MMFG structures with a formal language grammar with an approximation approach. Based on this, we now formally model semantic and explainable MMFGs and hence introduce several extensions to MMFGs and the corresponding grammars.

3.2. Enabling annotation of formal MMFG representations

Modeling a formal representation for MMFGs, we initially apply an annotation pattern to support the linking of External Semantic Annotations with a special node type *Annotation Anchor (aa)* to represent a link to an external semantic annotation. Such *Annotation Anchors* can be linked to syntactic MMFG nodes or syntactic relationships with a special relationship type, the *Annotation Relationship (ar)*. To support this, also the representation of all MMFG relationships has been extended by corresponding annotation anchors to allow their semantic annotation. Thus, any syntactic MMFG node or relationship can be linked with an *Annotation Relationship* to an *Annotation Anchor*. This means, that any syntactic resource within a MMFG can be semantically annotated.

The formal representation of such a basic MMFG including annotation anchors is shown in Figure 4(a) and will be further detailed in the following sections. Figure 4(b) shows the extension of an exemplary MMFG by *Annotation Anchors (aa)* and *Annotation Relationships (ar)*. To clearly distinguish between a feature and its textual representation, the introduced variable LBL is employed, which allows the production of human understandable textual representations of MMIR features.

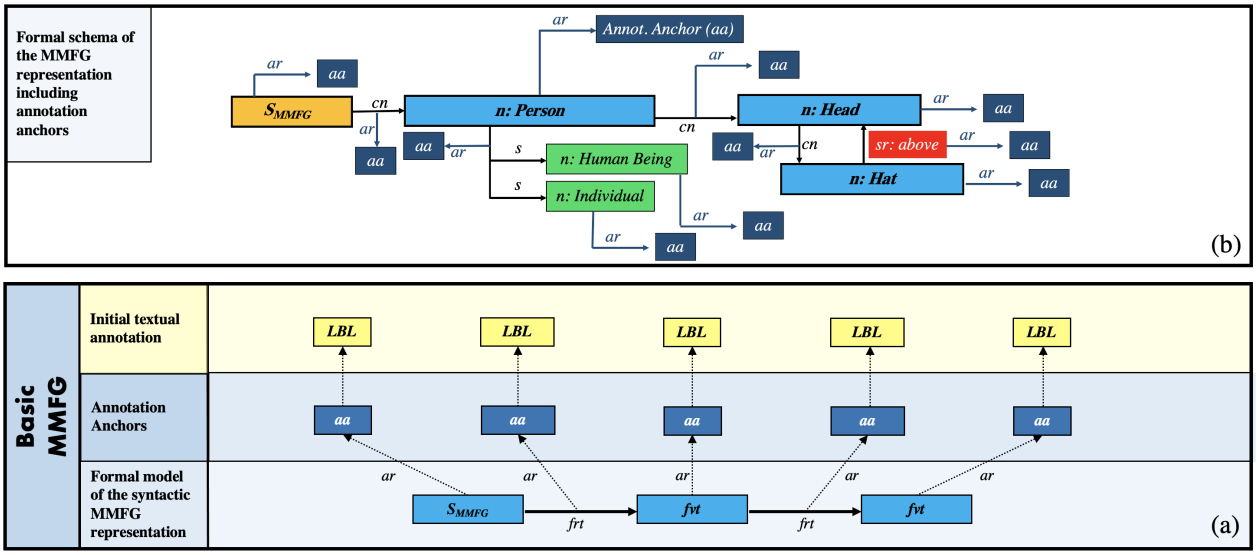

Figure 4. MMFG overview including *Annotation Anchors (aa)* and *Annotation Relationships (ar)*: (a) formal model, (b) formal syntactic schema of the MMFG representation $MMFG_{ex}$.

With these extensions, MMFG nodes and relationships can initially be linked to elements of existing semantic representations of vocabularies, taxonomies, or ontologies, and their corresponding machine-readable representations in the Semantic Web. The grammar G_{MMFG} can be extended to support the construction of formal language expressions in-

cluding the annotation pattern by adding the elements *ar* and *aa* to the set V_{MMFG}, so that $V_{MMFG} = FRT \cup FVT \cup ar, aa$ and by refining the production rules as follows:

$$\begin{aligned} P_{MMFG} = \{S_{MMFG} &\rightarrow FVT\ FRT\ FVT, \\ FVT &\rightarrow ar, \\ FRT &\rightarrow ar, \\ ar &\rightarrow aa, \\ aa &\rightarrow LBL\} \end{aligned}$$

Now, the grammar G_{MMFG} supports the construction of additional language expressions like, "person is-annotated-with-the-semantic-concept rdf:person. has-child is-annotated-with-the-semantic-concept rdf:related. head is-annotated-with-the-semantic-concept rdf:head". This annotation pattern will now be further employed as a basis for the semantic representation of MMFGs.

3.3. Semantic annotation of formal MMFG representations

The introduction of *Annotation Anchors* and *Annotation Relationships* is a purely syntactic extension. However, this syntactic structure must be annotated semantically. Hence, we now introduce a semantic annotation, which means, that each syntactical element of a MMFG will be annotated with *semantics* and the purely syntactic MMFG becomes a semantically annotated MMFG - a *Semantic Multimedia Feature Graph (SMMFG)*. For such a SMMFG, we define the following additional elements or structures:

- a *Semantic Node Representation (snr)* for each MMFG node, and a *Semantic Relationship Representation (srr)* for each MMFG relationship. These elements are required to represent the *meaning* of both nodes and relationships semantically.
- a set of *Semantic Feature Vocabulary Terms* $SFVT_{SMMFG}$ and a set of *Semantic Relationship Vocabulary Terms* $FRVT_{SMMFG}$. In previous work [7][43], we already defined the set FVT_{MMFG} as the representation of all syntactic MMFG vocabulary terms (i.e., the textual labels of detected features). In a SMMFG, the semantic of each syntactic vocabulary term, $fvt_i \in FVT_{MMFG}$, is now represented by a semantic feature vocabulary term $sfvt_i \in SFVT_{SMMFG}$. Analogously, the set, $SRVT_{SMMFG}$, is defined as the set of semantic representations of labels related to syntactic MMFG relationships.
- while in a MMFG, relationships are simply represented by their relationship type (e.g., cn, sr, s), in SMMFGs, these relationship types are modeled by *Semantic Relationship Vocabulary Terms (srvt)*, which represent the semantics of vocabulary terms describing the relationship type.
- in a MMFG, each node and each relation is linked by an *Annotation Relationship (ar)* to an *Annotation Anchor (aa)*, which now represents a *Semantic Node Representation (snr)* or a *Semantic Relation Representation (srr)*. An *Annotation Anchor (aa)* is an URI for the node or relation it represents and used to link these MMFG nodes or relations to semantic node representations and semantic relationship representations.
- in a SMMFG, the *hasName* relation links *Semantic Node Representations* with *Semantic Feature Vocabulary Terms* and *Semantic Relation Representations* with *Semantic Relationship Vocabulary Terms*. Each *srr* is linked via the *hasDomainNode* and *hasRangeNode* relations to the corresponding *snr's*.

Figure 5(a) shows, how *Annotation Anchors* can now be linked with *snr* and *srr* to semantic concepts described in the sets, $SFVT_{SMMFG}$, and, $SRVT_{SMFG}$. As already outlined, the semantic representation of the syntax of a MMFG contains relationships itself. For example, each *srr* has two relationships, *hasDomainNode* and *hasRangeNode*, which link to corresponding *snr* elements. They form the 1st level semantic annotation of a MMFG and thus a *Semantic Multimedia Feature Graph (SMMFG)*. The 2nd level semantic annotation is modeled by the elements of $sfvt$ and $srvt$, which represent the semantic information of the

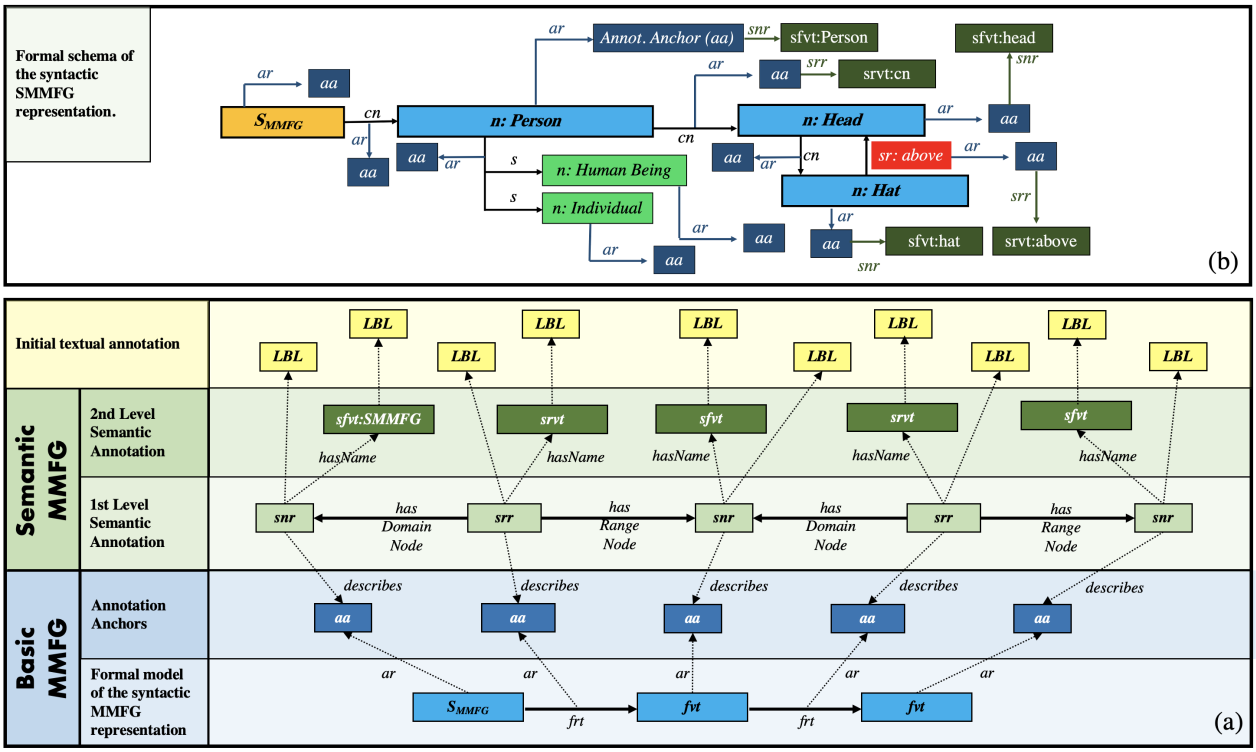

Figure 5. SMMFG overview: (a) formal model of the syntactic SMMFG. (b) formal syntactic schema of the SMMFG representation $SMMFG_{Ex}$.

feature vocabulary terms. Figure 5(b) shows the semantic extension applied to the above example.

Based on these syntactic extensions, also the formal grammar, G_{MMFG} can be extended, resulting in a formal grammar, G_{SMMFG} for the representation of valid formal language expressions of SMMFG structures. $G_{SMMFG} = (V_{SMMFG}, T_{SMMFG}, P_{SMMFG}, S_{SMMFG} = S_{MMFG})$, where

- $V_{SMMFG} = V_{MMFG} \cup SNR \cup SRR \cup SFVT_{SMMFG} \cup SRVT_{SMMFG}$ is the set of semantic representations of descriptions of MMFG nodes and relationships (see also Figure 5).
- T_{SMMFG} is an extension to, T_{MMFG}, and includes additional textual descriptions of the semantic relationships: "hasName", "hasDomainNode", "hasRangeNode", "describes".
- P_{SMMFG} extends the production rules, P_{MMFG}, as follows:

$$
\begin{aligned}
P_{SMMFG} = P_{MMFG} \cup \{ \\
srr \rightarrow snr\ snr\ SRVT\ aa \\
snr \rightarrow SFVT\ aa \\
aa \rightarrow LBL* \\
\}
\end{aligned}
$$

The grammar G_{SMMFG} supports the construction of additional language expressions like, "the-semantic-concept rdf:person hasName person. the-semantic-concept rdf:head hasName head. there-is-a-semantic-relationship-between rdf:person and rdf:hat which has-DomainNode rdf:person and hasRangeNode rdf:head. the-semantic-relationship between rdf:person and rdf:head is-described-by rdf:relation and hasName rdf:related".

Although these sentences describe further details of a SMMFG in a formally correct way and increase the readability for machines, *human-readability* decreases due to the mixture of syntactic, semantic, and structural labels. This means, that until now human-readability depends on the selection of adequate readable and understandable textual

labels. To eliminate this dependency, an automated construction of human-readable textual expressions must be achieved. Summarizing this, until now we defined a formal way to represent syntactic and semantic elements of MMFGs and SMMFGs by introducing the formal grammars, G_{MMFG} and G_{SMMFG}, with which not only the syntactic structure of a MMFG, but also the semantic enrichment of SMMFGs, can be represented by formal language expressions. To achieve our final goal of *human-understandable* i.e., natural language expressions, we introduce a *PS-grammar* in the next section, which transforms these machine-readable expressions into human-readable expressions.

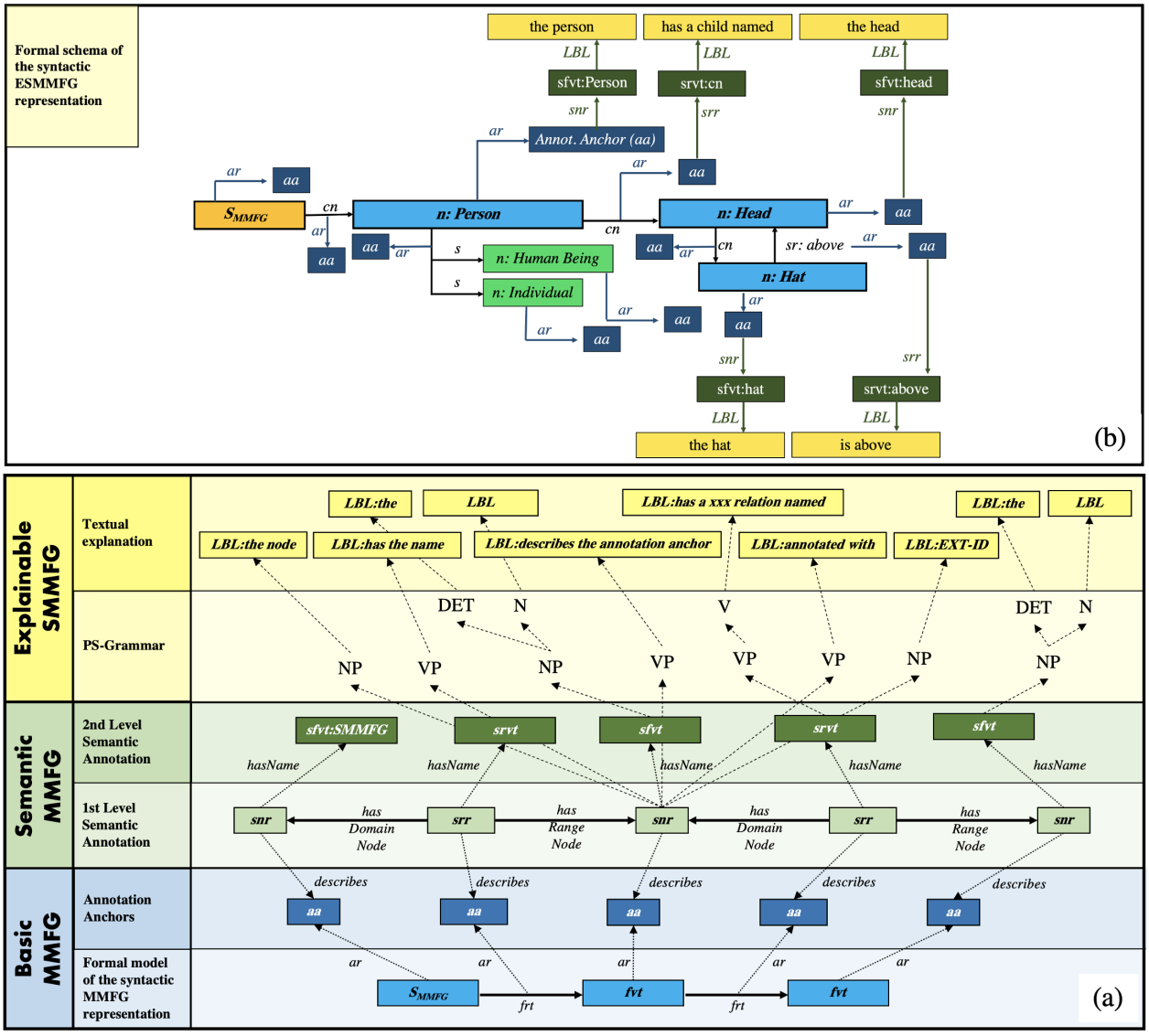

Figure 6. ESMMFG overview: (a) formal model of the syntactic ESMMFG representation. (b) formal schema of the syntactic ESMMFG representation.

3.4. Explainable SMMFGs

In section 2 we introduced the concept of *context-free PS-grammars*. This type of grammar is typically employed for the production of natural languages consisting of e.g., *Nominal Phrases (NP), Verbal Phrases (VP), nouns (N), verbs (V), adjectives (A), determiners (DET)*. The grammar $G_{en} = (V_{en}, T_{en}, P_{en}, S_{en})$ from section 2 can be employed to produce valid english sentences (as illustrated in Figure 2).

Based on the grammars, G_{en}, G_{MMFG}, and G_{SMMFG}, we can now define such a context-free PS-grammar, G_{ESMMFG}, which transforms any MMFG or SMMFG formally into

human-readable (i.e., explainable) natural language expressions. MMFGs or SMMFGs, that are extended in this way, become *explainable* and will be called *Explainable Semantic Multimedia Feature Graphs (ESMMFG)* in the remainder of this paper. Figure 6 shows the introduction of the PS-grammar and the corresponding schema for the syntactic ESMMFG representation.

Formally, we define, $G_{ESMMFG} = (V_{ESMMFG}, T_{ESMMFG}, P_{ESMMFG}, S_{ESMMFG})$ as follows:

- the variables, V_{ESMMFG} are based on the variables, V_{en}, of the english grammar, G_{en}, and additionally includes the variables of the previously defined grammars:

  $$V_{ESMMFG} = V_{en} \cup V_{MMFG} \cup V_{SMMFG}$$

  It thus represents the union of variables defining the *english grammar* (i.e., $V_{en} = NP, VP, V, N, DET, PR$), the *syntactic elements* of an MMFG (i.e., $V_{MMFG} = FRT \cup FVT \cup \{ar, aa\}$), and the *semantic enrichment* (i.e., $V_{SMMFG} = V_{MMFG} \cup SNR \cup SRR \cup SFVT \cup SRVT$).
- $T_{ESMMFG} \cap V_{ESMMFG} = \emptyset$ is the set of terminal symbols and represented by the labels LBL, which can be regarded as any english word of type noun, verb, determiner, adjective, or preposition. The production of these words is based on the semantic feature and semantic relationship vocabulary. The order, in which such $LBLs$ can be arranged to formulate valid expressions, is given by the following production rules.
- P_{ESMMFG} is the set of production rules and defines how the MMFG and SMMFG structures can be formally transformed into valid natural language expressions. P_{ESMMFG} also contains the simple production rules previously defined in, P_{MMFG} and P_{SMMFG}. However, the phrase structure of, P_{ESMMFG}, leads to various additional and refining elements:

  $$\begin{aligned}
  P_{ESMMFG} = \{ \\
  S_{ESMMFG} &\rightarrow FVT \\
  FVT &\rightarrow FVT\ FRT\ FVT \\
  FVT &\rightarrow ar\ SNR \\
  FRT &\rightarrow ar\ SRR \\
  SRR &\rightarrow SRVT\ SNR\ SNR \\
  SNR &\rightarrow SFVT\ NP\ VP \\
  SRVT &\rightarrow NP\ VP \\
  SFVT &\rightarrow NP\ VP \\
  ar &\rightarrow aa \\
  aa &\rightarrow LBL \\
  VP &\rightarrow V\ PP \\
  PP &\rightarrow PR\ NP \\
  NP &\rightarrow DET\ N \\
  PR &\rightarrow LBL \\
  N &\rightarrow LBL \\
  V &\rightarrow LBL \\
  DET &\rightarrow LBL \\
  \} &
  \end{aligned}$$

- $S_{ESMMFG} = S_{SMMFG} = S_{MMFG}$ is the starting symbol for any valid expression. This means, that any natural language representation of a MMFG or SMMFG starts with

the processing of the root-element. However, as the root-node of a MMFG is a node itself, G_{ESMMFG}, can also be employed to produce expressions of subgraphs of a MMFG or SMMFG.

The application of the production rules P_{ESMMFG} is shown in Figures 7, 8, and 9. For illustration purposes, some of the production rules representing mostly internal structures have been omitted for readability purposes.

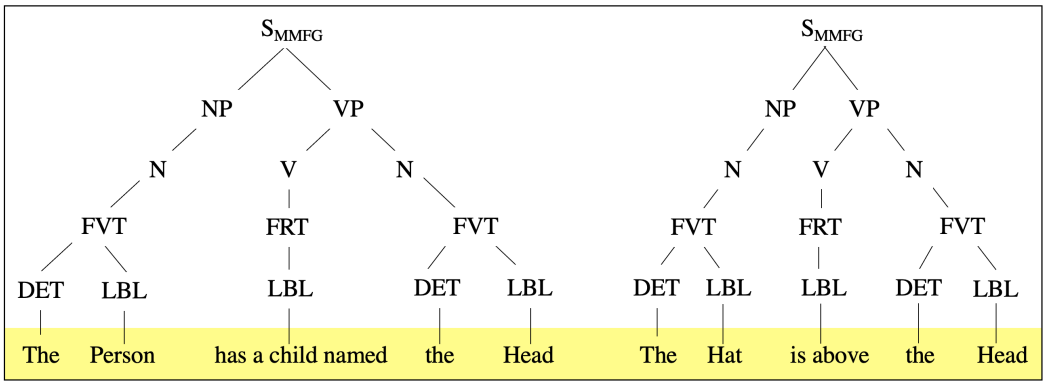

Figure 7. PS-tree with production rules, P_{ESMMFG}, of, G_{ESMMFG}, applied to $MMFG_{ex}$.

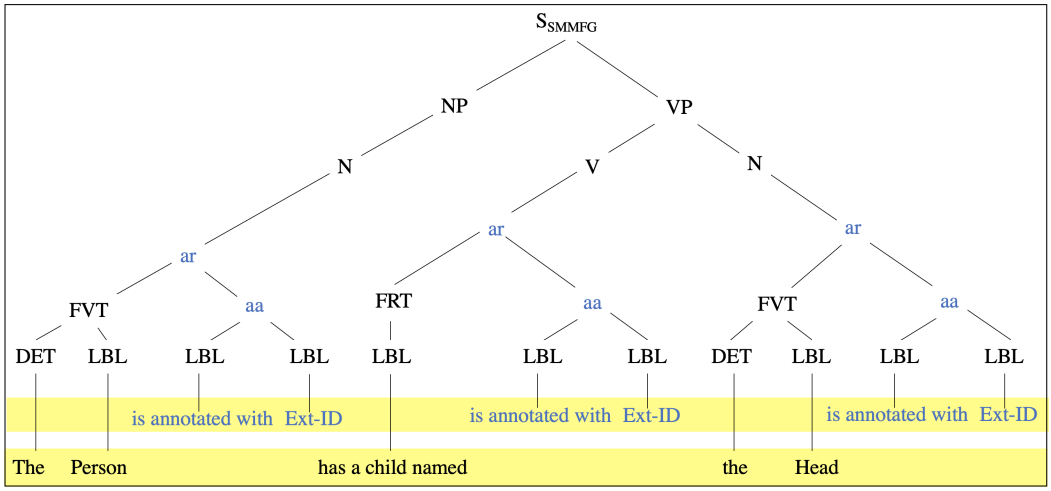

Figure 8. PS-tree with production rules, P_{ESMMFG}, of, G_{ESMMFG}, applied to $SMMFG_{ex}$ including annotation anchors.

These examples show, that the expressiveness of ESMMFGs increased significantly with the introduction of, G_{ESMMFG}, and that natural language sentences can now be built formally based on syntactic and semantic structures of MMFGs.

It is important to note, that *any* natural language expression, that is generated based on, G_{ESMMFG}, is content-wise *true* (i.e., correct) as it purely represents the original multimedia features in a formal, but human-readable way. In addition, G_{ESMMFG}, provides unlimited options for the production of valid natural language expressions due to its underlying phrase structure. This means, that any multimedia feature can now be represented as a natural language, human-readable text. It is up to the application to define, which phrases should be used, which level of abstraction should be applied, or which subset of MMFG-nodes has to be selected for the natural language representation. Examples of the application of, G_{ESMMFG}, to multimedia features of various domains is given in section 5 (evaluation).

The formal definition of G_{ESMMFG} furthermore guarantees, that any element of a MMFG (i.e., any detected multimedia feature) can be structurally and semantically represented. It also ensures, that the semantic information of any multimedia feature can be

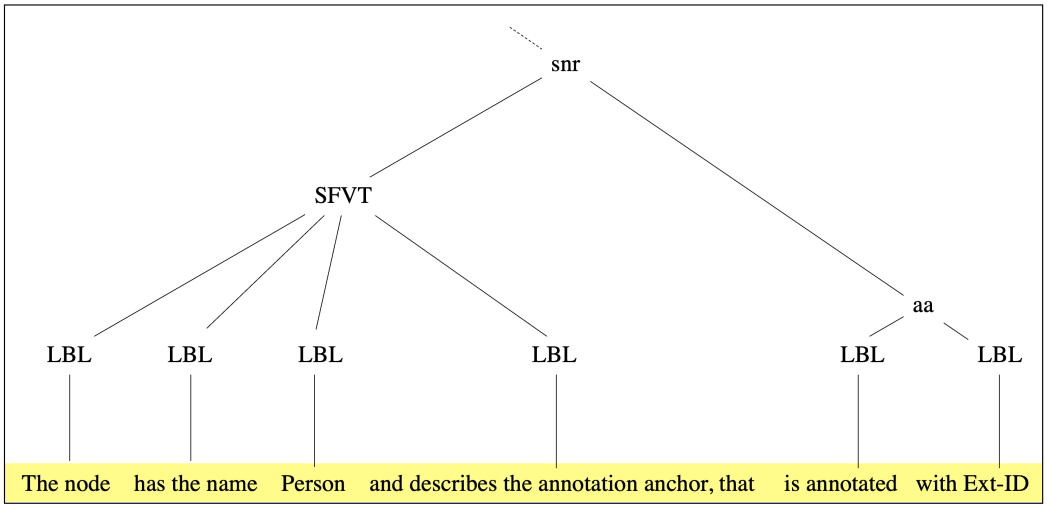

Figure 9. Snippet of the PS-tree with production rules, P_{ESMMFG}, of G_{ESMMFG}, applied to $SMMFG_{ex}$, including semantic feature vocabulary terms.

mapped to semantic systems, interpreted, and employed for inferencing and reasoning. Furthermore, any MMFG can now be represented as a syntactically correct and human readable text, which further supports automatic processing by employing a selection of numerous text-bases algorithms, e.g., for argument extraction. However, the generated text is highly dependent on the construction of phrases based on the detected (or calculated) MMIR features in the original MMFG. As this textual representation might be different depending on the MMIR processing step (e.g., the explanation of query construction, result presentation, or the ranking of an element in the result list), also different strategies for the construction of phrases need to be employed. This is reflected by introducing various subclasses for the corresponding processing steps as illustrated by the implementation samples (see section 4).

Currently, the order of the constructed sentences is based on the order of nodes in the original MMFG data structure. This will produce good results for text based multimedia documents, as the order of explaining texts will follow the document structure. However, for other multimedia types (e.g., images), the order of the descriptions of detected objects will be random. This can be subject to further improvements in future work. It should also me mentioned, that the presented concept is a pure mathematical approach to calculate explaining texts for multimedia features without any need of machine learning tasks as, e.g., in deep LSTM language modeling [49].

In this section, we outlined, how an MMFG can be semantically represented and extended by a *Semantic Multimedia Feature Graph*. We defined how explainability and transparency can be introduced to syntactic data structures based on MMIR features resulting in *Explainable Multimedia Feature Graphs*. To evaluate the full potential of this semantic extension, we apply this concept to the *Generic Multimedia Analysis Framework (GMAF)*, in which MMFGs and now also SMMFGs, and ESMMFGs can be processed employing *Graph Codes*, which are particularly optimized for MMIR calculations. Hence, in the following subsection, we show briefly how the concept of *Graph Codes* can be semantically represented and extended by applying the algorithms of *Graph Codes* to *Semantic Multimedia Feature Graphs*.

3.5. Semantic Graph Codes

Graph Codes are a 2D representation of MMFGs, which are computationally optimized particularly when employed for MMIR. They are calculated from MMFGs by employing an encoding function, f_{enc}, which transforms a MMFG into the *Graph Code* structure. In the following subsections, we define how *Graph Codes* can be transformed into *Semantic Graph Codes* with corresponding operations.

Until now, the dictionary of a *Graph Code (GC)* is based on the feature vocabulary terms of detected features (i.e., textual labels describing detected features) of the MMFG. The dictionary of *Semantic Graph Codes (SGC)* is based on the semantic representation of the meaning of such feature vocabulary terms, i.e., unique labels of elements of SNR. Thus, in a SGC, unique identifiers are used to represent the dictionary. However, this representation can lead to ambiguous representation possibilities.

For example, the feature vocabulary term, *Jaguar*, of a *Graph Code*, GC_{Jaguar}, could be connected to the semantic node representation for an animal, but it could also be connected to the semantic node representation for a car. To solve these problems, a function, $sq(vt_i)$, is introduced to determine the unambiguous semantic representation of the meaning of a feature vocabulary term. $sq(vt_i)$ performs a semantic query for each, vt_i, to the semantic model and receives either a single unique result, $snr_i \in SNR$, representing the meaning of, vt_i, or it receives a list of possible results (e.g., the *Jaguar-animal* or the *Jaguar-car* semantic nodes). To identify the correct element in this list of possible results, we apply the *Graph Code Similarity* algorithm to, GC_{Jaguar}, and each element of the result list. To do this, for each element of the result list, we perform an additional query on the semantic model (e.g., for the *Jaguar-car*) and represent the result as an MMFG, which is then transformed into a *Graph Code*, $GC_{Result_i}, GC_{Result_j}$. The, GC_{Jaguar}, will not just contain the detected feature vocabulary term *Jaguar*, but additional information. As *Graph Codes* support a high Level-Of-Detail (LOD) and are generated by recursive processing, they will also contain numerous additionally detected feature vocabulary terms, like *wheel, road, window, ...* or *whiskers, teeth, furr, ...* Our query to the semantic model also returns relationships of the *Jaguar* to semantic node representations of some of these detected feature vocabulary terms. Thus, if the *Jaguar* in our MMFG is a car, the similarity to the result of the semantic query for the *Jaguar-car* will be more similar to, GC_{Jaguar}, than the result of the *Jaguar-animal* query. In addition, *topic modeling* can be applied to further optimize the selection of semantic query results as unambiguous semantic node representations.

Summarizing this, the construction of *Semantic Graph Codes* utilizes a semantic query, $sq(vt_i)$, for each feature vocabulary term, vt_i, to identify the unambiguous semantic node representation, snr_i. To construct SGCs, the encoding function, f_{enc}, has to be modified to employ, $sq(vt_i)$, when calculating the dictionary for SMMFGs:

$$f_{enc}(MMFG) = GC_{MMFG} \tag{1}$$

$$f_{enc}(SMMFG) = SGC_{SMMFG} \tag{2}$$

$$\forall snr_i \in SMMFG, \forall vt_i \in MMFG : sq(vt_i) = snr_i \tag{3}$$

In addition to the calculation of IDs, f_{enc} will also eliminate synonym and relationship nodes from a SGC-dictionary, as they are represented in the corresponding semantic model. So, f_{enc} will return an empty value for MMFG nodes of the type *Synonym*.

For further illustration of our example (see Figure 1(c)), we define the function, $sq(vt_i)$, in a way, that it returns the following values for the vocabulary terms of this example (see Table 2).

vt_i	$sq(vt_i)$
Person	101
Head	102
Hat	103
above	104
Individual	-
Human Being	-

Table 2: Exemplary function, $sq(vt_i)$, and values for GC_{ex}.

For the vocabulary terms "Individual" and "Human Being" representing MMFG nodes of type *Synonym*, the function, $sq(vt_i)$, does not return any value, as these relationships are already represented by the semantic model and hence don't need to be repeated in each

individual *Semantic Graph Code*. Applying, $sq(vt_i)$, and, f_{enc}, to our example, this would result in a compressed *Semantic Graph Code* SGC_{ex} (see Table 3).

SGC_{ex}	101	102	103	104
101	**1**	3	0	0
102	0	**1**	3	0
103	0	0	**2**	5
104	0	0	0	**6**

Table 3: Table representation of, SGC_{ex2}, including external unique identifiers as row and column descriptors.

Further concepts of *Graph Codes*, like the calculation of similarity, recommendations, querying, or result presentation remain unchanged. However, it should be noted, that *Semantic Graph Codes* lead to a further compression of the *Graph Code* matrix, as synonyms, or common knowledge can be removed from the *Graph Code*, as it is already represented in the external semantic system. Furthermore, new knowledge which exists in the external system, can be employed for *Semantic Graph Codes*, serving as a basis for inferencing and reasoning. As the initial construction of *Graph Codes* is purely based on the detected feature vocabulary terms of a given multimedia object, the *Graph Code* vocabulary is typically very small. For *Semantic Graph Codes*, these feature vocabulary terms are translated into semantic IDs, or even removed, when they exist in the general semantic model of the application, which leads to a further compression, but also requires a representation of the overall semantic model (i.e. the ontology or taxonomy). This model can be represented in a SKOS, which is queried at runtime to identify such vocabulary terms. It is also possible to represent the complete semantic knowledge in the form of a *Semantic Graph Code*. However, this would lead to a very high number of feature vocabulary terms (i.e. any term in the ontology), and to very large *Semantic Graph Codes*. This approach is not recommended, as *Graph Codes* are optimized for indexing and not for knowledge representation.

3.6. Summary

In this section, we discussed our approach to formally defining natural language expressions from multimedia feature graphs. We showed how a context-free PS grammar can be built to generate human-readable English sentences formally and thus are able to close the gap between the technical representation of a multimedia feature and a human-understandable representation of the *meaning* of such a feature. We also showed how semantics can be introduced to the GMAF framework in the form of *Semantic Graph Codes*. Based on this modeling, we now provide further details of the implementation in the next section.

4. Implementation

Basis for the implementation of the semantic extensions and concepts discussed is the current GMAF prototype, which is written in Java and has a Java Swing UI attached. The prototype including the presented code samples of this section is available at [9] frequently updated according to the ongoing progress of this research. Following the Factory Design Pattern [44], the GMAF has been extended to utilize external semantic representation frameworks. In this section, we discuss exemplary implementation details. First of all, in subsection 4.1, the representation of SMMFGs by RDF and RDFS is given.

4.1. RDF and RDFS Representation of MMFGs

As discussed in section 3 (modeling), we can represent any SMMFG with RDF and RDFS. The implementation of this is shown here:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

```
<rdfs:Class rdf:ID="MMFG">
        <rdfs:comment>MMFG Root Node Class</rdfs:comment>
</rfds:Class>

<rdfs:Class rdf:ID="Node">
        <rdfs:comment>MMFG Node Class</rdfs:comment>
</rfds:Class>

<rdfs:Class rdf:ID="Annotation_Anchor">
        <rdfs:comment>Annotation Anchor</rdfs:comment>
</rfds:Class>

<rdfs:Class rdf:ID="Relationship">
        <rdfs:comment>Relationship</rdfs:comment>
</rfds:Class>

<rdf:Property rdf:ID="Feature">
        <rdfs:comment>Feature respresenting nodes of a MMFG
        </rdfs:comment>
        <rdfs:domain rdf:resource="#MMFG"/>
        <rdfs:range rdf:resource="#Node"/>
</rdf:Property>

<rdf:Property rdf:ID="cn">
        <rdfs:comment>child relationship
        representing the level of detail
        </rdfs:comment>
        <rdfs:domain rdf:resource="#Node"/>
        <rdfs:range rdf:resource="#Relationship"/>
</rdf:Property>

<rdf:Property rdf:ID="sr">
        <rdfs:comment>Spacial relationship</rdfs:comment>
        <rdfs:domain rdf:resource="#Node"/>
        <rdfs:range rdf:resource="#Relationship"/>
</rdf:Property>

<rdf:Property rdf:ID="s">
        <rdfs:comment>Synonym relationship</rdfs:comment>
        <rdfs:domain rdf:resource="#Node"/>
        <rdfs:range rdf:resource="#Relationship"/>
</rdf:Property>

<rdf:Property rdf:ID="ar">
        <rdfs:comment>Semantic relationship</rdfs:comment>
        <rdfs:domain rdf:resource="#Relationship"/>
        <rdfs:range rdf:resource="#Annotation␣Anchor"/>
</rdf:Property>

<rdf:Property rdf:ID="aa">
        <rdfs:comment>Semantic relationship</rdfs:comment>
        <rdfs:domain rdf:resource="#Node"/>
        <rdfs:range rdf:resource="#Annotation␣Anchor"/>
</rdf:Property>
</rdf:RDF>
```

This implementation allows the transformation of a pure syntactical vocabulary into a semantic vocabulary, as now the syntactical model of a SMMFG can be represented by a formally well-founded semantic schema. Thus, the syntactic labels can be linked to or

represented as semantic concepts. This means, that any information of the above example can now be represented with RDF-statements as follows:

```
<rdf:RDF>
        <rdf:Statement rdf:about="mmfg:Statement">
        <rdf:subject rdf:resource="Node:Hat"/>
        <rdf:predicate rdf:resource="Relationship:above"/>
        <rdf:object rdf:resource="Node:Head"/>
</rdf:Statement>
...
</rdf:RDF>
```

Additionally, any information can now be linked to external semantic concepts:

```
<rdf:RDF xml:lang="en" ... xmlns:mmfg="mmfg.rdf">
        ...
        <mmfg:Node rdf:about="Node">
        <mmfg:name>Hat</mmfg:name>
</mmfg:Node>

<mmfg:Hat>
        <mmfg:sr rdf:resource="mmfg:Head" rdf:about="above"/>
        <mmfg:ar rdf:resource="mmfg:rel_3"/>
</mmfg:Hat>

<mmfg:Relationship>
        <mmfg:name>rel_3</mmfg:name>
        <mmfg:aa rdf:resource="mmfg:aa_3"/>
</mmfg:Relationship>

<mmfg:AnnotationAnchor>
        <mmfg:name>aa_3</mmfg:name>
        <mmfg:comment>
        URI of external semantic representation
        </mmfg:comment>
</mmfg:AnnotationAnchor>
...
</rdf:RDF>
```

As the MMFG is now represented in RDF and RDFS and thus aligned with semantic concepts, a further integration based on the GMAF framework can be implemented. This is described in the next subsection.

4.2. Semantic Extension of the GMAF

For each MMIR application based on the GMAF, an external semantic framework can be attached by implementing (and configuring) an adapter class, which is defined by the interface *Semantic Extension* (see Figure 10). The detected vocabulary term is then passed to the external semantic framework.

Whenever *Semantic Graph Codes* must be calculated (i.e., for indexing, retrieval, querying, filtering) in the GMAF, this extension is called to provide information from or to an external semantic representation. In our prototype, we built two implementations for this. One serves as a reference for the connection of external semantic representations and is implemented to employ the Semantic Web (class *SemWebExtension*), the other serves as an internal default implementation to illustrate and validate the concepts of section 3, in particular Relevance and Topic calculation and inferencing (class *DefaultExtension*).

4.3. Semantic Representation

As external semantic representation, several tools, databases, or services can be applied. In our prototypical implementation, we chose Wikidata [45][46], as it not just serves as a basis for many other MMIR applications, it also provides a fully functioning SPARQL

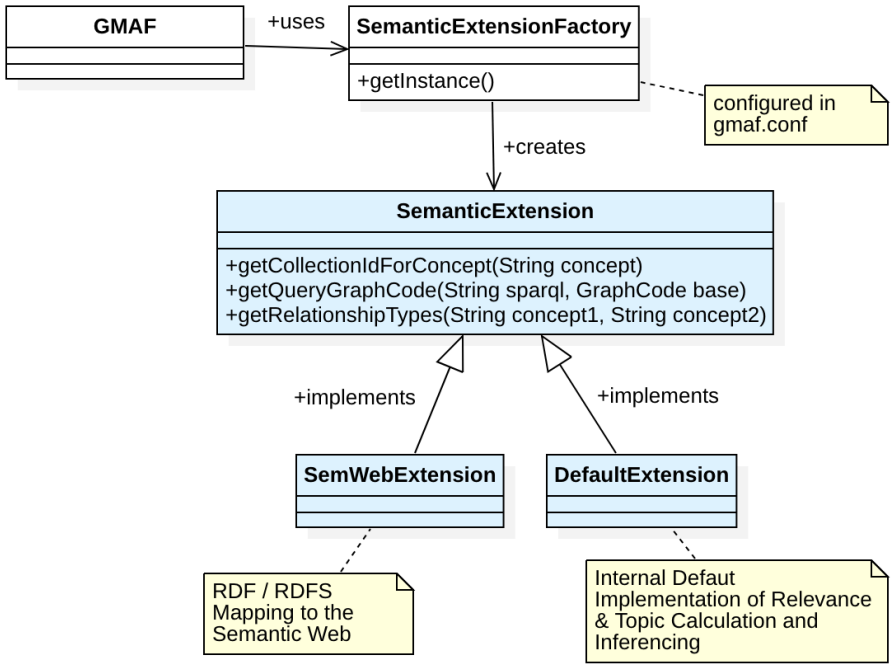

Figure 10. Factory implementation to attach external semantic frameworks.

interface [42], which can be utilized for semantic querying. Hence, attaching Wikidata to the GMAF is straightforward as illustrated in Figure 11.

As Wikidata already supports SPARQL, GMAF-SPARQL-Queries can simple be forwarded and fused with already existing Query Graph Codes. The resulting *Semantic Graph Codes* are displayed in the GMAF-UI as shown in Figure 12 and can be applied to all kind of queries (including Manual Querying, Query By Example, and Query Refinement) [24] and result presentation.

As illustrated in this subsection, connecting external semantic representation systems to the GMAF is quite straightforward. In the case of external systems, the effectiveness of semantic results in terms of Precision and Recall experiments mainly depends on the external system. In our evaluation, we wanted to compare our internal default representation and the algorithms for topic modeling and intelligent information retrieval with these external systems, as semantic extensions are handled transparently within the GMAF (i.e., the GMAF itself does not perform any enrichment or modification). Hence, in the next subsection, we discuss the implementation of the defaults, which will also be employed later for evaluation purposes.

Summarizing this, we showed in this section, that the semantic extension to the GMAF can be implemented based on an interface extension point, which generically provides access to external systems. If no such external system is available, the internal default algorithms for semantic extensions can serve as a good alternative, as *Graph Codes* and their metrics provide a well-defined mathematical model for intelligent information retrieval. An evaluation of our implementation including experiments is given in section 5.

4.4. Explainability

For the implementation of *Explainable Semantic Multimedia Feature Graphs*, we apply the design patterns, *Interpreter*, *Composite*, and *Facade*. The *Composite* pattern is employed to recursively process ESMMFG nodes and to construct the final human-readable text. The *Interpreter* pattern is chosen to represent the phrase structure of the underlying grammar, where each existing element (e.g., NP, VP, FVT, etc.) is encapsulated by a subclass being responsible for the correct construction of valid expressions. Finally, the *Facade* pattern serves as a wrapper and provides a simple API, with which explanations of a given MMFG can be generated. A simple call of this module can be made as follows:

```
public class SemWebExtension implements SemanticExtension {
    private static final String endpointUrl = "https://query.wikidata.org/sparql";


    public String getCollectionIdForConcept(String concept) {
        try {
            String querySelect = "SELECT ?item ?itemLabel WHERE {\n" +
                    "  SERVICE wikibase:mwapi {\n" +
                    "      bd:serviceParam wikibase:endpoint \"www.wikidata.org\";\n" +
                    "        wikibase:api \"EntitySearch\";\n" +
                    "        mwapi:search \"" + concept + "\"; \n" +
                    "        mwapi:language \"en\".\n" +
                    "      ?item wikibase:apiOutputItem mwapi:item.\n" +
                    "  }\n" +
                    "  SERVICE wikibase:label {bd:serviceParam wikibase:language \"en\".}\n" +
                    "}\n" +
                    "ORDER BY ?item \n" +
                    "LIMIT 1";

            URI endpoint = new URI(endpointUrl);
            SparqlClient sc = new SparqlClient(false);
            sc.setEndpointRead(endpoint);
            sc.setMethodHTTPRead(Method.GET);

            SparqlResult sr = sc.query(querySelect);
            String id = "" + sr.getModel().getValueAt(0, 0);
            id = id.substring(id.lastIndexOf("/") + 1, id.length());
            return id;
        }
        catch (Exception x) {
            x.printStackTrace();
        }
        return concept;
    }
```

Figure 11. Querying an external semantic system for unique IDs with SPARQL.

```
public class Explainer {
        public static String explain(MMFG mmfg, int levelOfDetail,
                                     int languageLevel) {
                SMMFG smmfg = new SMMFG(mmfg);
                ESMMFG esmmfg = new ESMMFG(smmfg);
                LanguageModel model =
                        LanguageModel.getInstance(languageLevel);
                String text = model.produceText(esmmfg, levelOfDetail);
                return text;
        }
        ...
}
...
public abstract class LanguageModel {
        public static final int SIMPLE = 0;
        public static final int NORMAL = 1;
        public static final int COMPLEX = 2;

        public static LanguageModel getInstance(int languageModel) {
                if (languageModel == SIMPLE)
                        return new SimpleLanguageModel();
                else if (languageModel == COMPLEX)
                        return new ComplexLanguageModel();
                else return new DefaultLanguageModel();
        }

        public abstract PSTree producePSTree(
                ESMMFG esmmfg, int levelOfDetail);
        public abstract PSTree produceQueryPSTree(
                ESMMFG esmmfg, int levelOfDetail);
        public abstract PSTree produceResultPSTree(
                ESMMFG esmmfg, int levelOfDetail);
        public abstract PSTree produceComparisonPSTree(
```

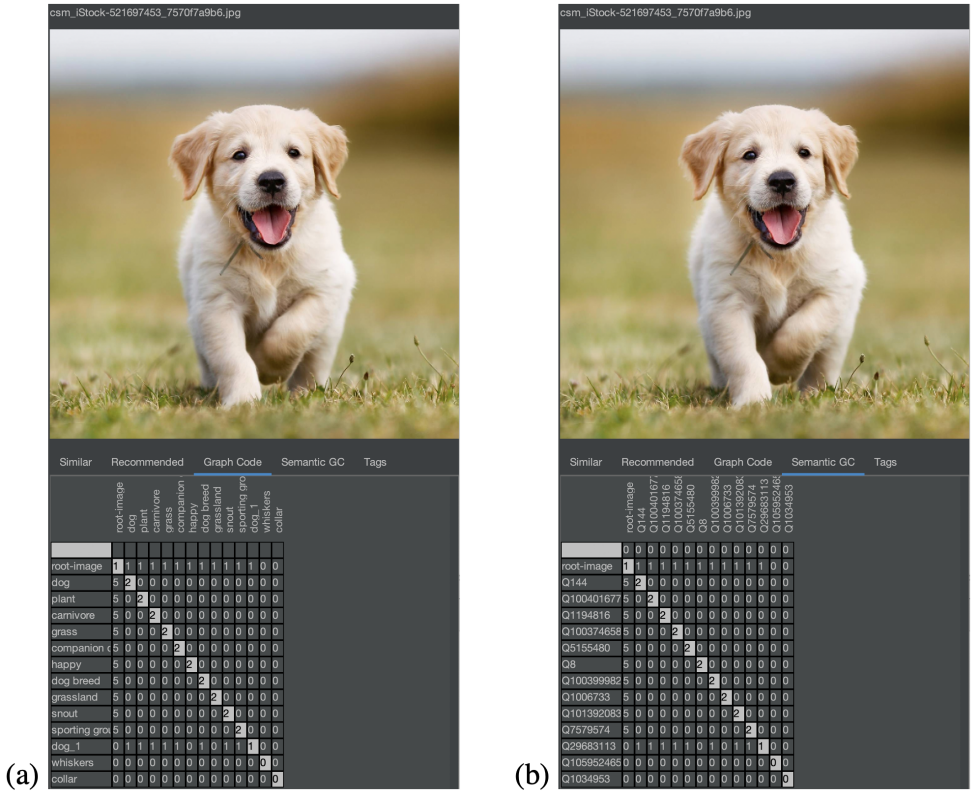

Figure 12. GMAF-UI displaying image (a) *Graph Codes* and (b) the corresponding *Semantic Graph Code*.

```
                ESMMFG esmmfg1, ESMMFG esmmfg2);

        public final String produceText(ESMMFG esmmfg,
                        int levelOfDetail) {
                PSTree ps = producePSTree(esmmfg, levelOfDetail);
                return ps.createSentence();
        }
}
```

The parameter *levelOfDetail* is used to define the number of recursions, that should be applied for the generation of natural-language text. This directly corresponds to the level of detail of the detected MMIR features. The parameter *languageLevel* can define the style of the produced natural language text. Currently, there is a selection of, *simple, medium, complex*. Furthermore, for each step of the MMIR process, a different PS-Tree can be constructed. For example, the method *produceComparisonPSTree* will calculate a PS-Tree with phrases to explain, why *esmmfg1* has been ranked before *esmmfg2* in a result list. The method *produceResultPSTree* calculates a PS-Tree with phrases, that explain, *why* an element is part of the result list. And, the method *produceQueryPSTree* would construct a PS-tree with phrases to outline, which query has been calculated e.g. from a given keyword list or from a query by example pattern. The solution has been implemented in an extendable way, so that further subclasses of *LanguageModel* can be employed to refine, extend, or newly define human understandable natural language phrases. Figure 13 shows the result for a given image with settings, $levelOfDetail = 2$ and $languageLevel = Simple$. Figure 13 also shows the integration of explainability into the GMAF user interface.

5. Evaluation

In this section, we discuss concept and algorithm evaluation. In previous work [7], we evaluated *Graph Codes* retrieval against existing graph-traversal-based algorithms and were

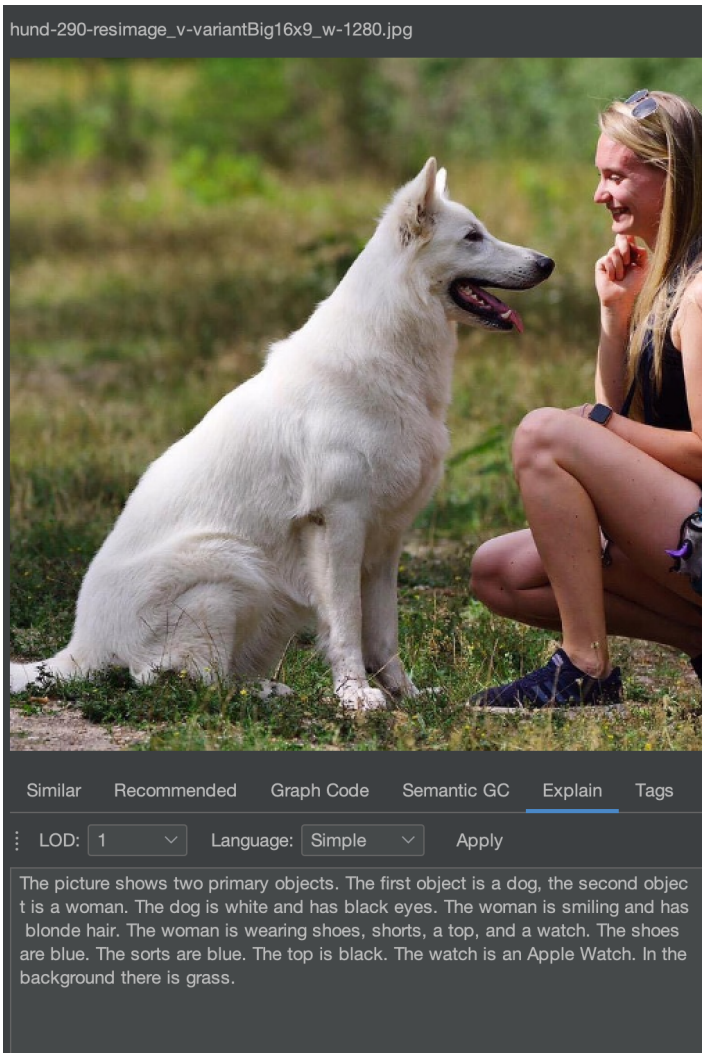

Figure 13. GMAF-UI displaying a human-readable explanation of an image.

able to prove, that their efficiency and effectiveness are superior to graph-based solutions. In the first part of this section, we extend this previous evaluation by experiments based on *Semantic Graph Codes* based on images from the Flickr30k and the DIV2K dataset. In the second part of this evaluation section, we chose to employ the text sample dataset of the 2021 TREC conference's News evaluation [47] with 600.000 full-text articles from the Washington Post [5] to illustrate semantic retrieval and inference.

5.1. Semantic Retrieval

To determine the effectiveness of the *Graph Code Algorithm*, we selected 5 test-queries from the annotations of a random set of 1000 Flickr30k images and calculated values for precision and recall for these [7]. When *Graph Codes* are transformed into *Semantic Graph Codes*, the same evaluation employs synonyms and "is-a" relations of the external (or internal) semantic model. In the following experiment, we compared our previous results with results based on external and attached internal semantic models. Table 4 shows the measured results for queries based on data, which contain, *rel*, relevant results, *sel* matching results. The columns, Precision P_B, Recall R_B, and the F1-score ($F1 = 2 * \frac{P_B * R_B}{P_B + R_B}$), $F1_B$ contain values for the basis experiment without any semantic extension. The values in columns, P_I, R_I, and $F1_I$ are calculated for the internal semantic analysis, and the values in columns P_E, R_E, and $F1_E$ are calculated when using an external semantic model, in this case the Wikidata extension.

	Dog	Man	Golf	Guitar	Bicycle	Avg
Basic Experiment						
tp_B	188	119	5	17	54	
tn_B	2	85	2	13	24	
P_B	0.98	0.53	0.71	0.56	0.69	0.71
R_B	0.91	0.25	0.45	0.89	0.94	0.69
$F1_B$	0.94	0.35	0.55	0.69	0.80	0.67
Attachment of External Framework						
tp_E	188	309	6	26	63	
tn_E	2	104	2	6	19	
P_E	0.98	1.51	0.85	0.86	0.80	1.00
R_E	0.91	0.67	0.54	1.36	1.10	0.92
$F1_E$	0.94	0.92	0.66	1.06	0.93	0.90

Table 4: Effectiveness based on the Flickr30k dataset

The findings of this experiment can be summarized as follows:

- any semantic enrichment increases the values for precision and recall (summarized by their F1 value) by 18% (see blue cells in Table 4)
- additional 4% increase can be achieved, when an external semantic system is connected

These results for effectiveness are currently applied to an image dataset [48]. However, in many MMIR applications, text retrieval rates of effectiveness are also important. Hence, in the next subsection, we discuss the evaluation of our algorithms based on text datasets.

5.2. Text Retrieval and Inference

For the evaluation of text retrieval, we employed the TREC2020 dataset of the Washington Post News Archive [47][5], and followed the evaluation criteria of the TREC2021 challenge, which is based on *Similarity (Top-10)* and *Recommendations (Top-10)*. For both tasks, the calculation of a semantic model of each text is required, in our case a MMFG and the corresponding SGC, including the application of a metric. As discussed in [24], similarity can be calculated by applying, M_F, based on semantic vocabulary terms, and recommendations are calculated by applying, M_{RT}. For this evaluation, we measured effectiveness based to the published results of previous years. In the first test-scenario, we applied a standard "Bag-Of-Words" algorithm without any semantic enhancement. The second test-scenario then employs the full semantic analysis and features described in section 3, but does not yet include reasoning and inferencing. This is added in the third experiment. In the fourth scenario, we attached an external framework (Wikidata) and compared all the results to the TREC reference results. We measured the values, P_{Sim}, and, R_{Sim}, as Precision and Recall of Similarity, i.e., if the retrieved documents are in the Top-10-List, P_{Rec}, and, R_{Rec}, as Precision and Recall for the Recommendation Top-10, and the corresponding, $F1_{Sim}$, and, $F1_{Rec}$, values.

Experiment	P_{Sim}	R_{Sim}	$F1_{Sim}$	P_{Rec}	R_{Rec}	$F1_{Rec}$
Bag-Of-Words	0.3	0.3	0.3	0.2	0.2	0.2
Internal Impl.	0.7	0.7	0.7	0.6	0.6	0.6
Inferencing	0.8	0.8	0.8	0.8	0.8	0.8
External Impl.	0.7	0.8	0.7	0.5	0.5	0.5

Table 5: Result of text analysis based on the Washington Post News dataset

This experiment shows, that the introduction of semantics to text analysis provides an increase in effectiveness of 150%. The difference between an external and internal implementation exists, but is not very significant and highly dependent on the dataset and

the external system. However, a more detailed evaluation of external systems can provide further insight. But, in the context of this paper, we are able to prove that the concepts of section 3 are also valid for text retrieval and provide a significant increase in effectiveness.

5.3. Explainability

In addition, to these results in the area of efficiency and effectiveness, *Explainability* has also been evaluated. For this evaluation, we generated various texts describing a number of typical MMIR asset types and their corresponding features. An exemplary text is shown in Figure 13.

Thus, in this evaluation section, we were able to show, that semantic enrichment of MMIR applications provides a significant increase in retrieval effectiveness. We also showed that explainability provides a huge potential, in particular, in combination with *Semantic Graph Codes*, topic modeling, and intelligent information retrieval, improvements of up to 150% can be achieved. Finally, in the next section, we summarize our results.

6. Conclusion and Future Work

In this paper, we discussed concepts and algorithms for narrowing the semantic gap [1], i.e., the gap between detected features in Multimedia objects and the meaning of these features. We introduced a well-defined semantic representation of the MMFG and enhanced the concept of *Graph Codes* to fully support semantic querying, filtering, reasoning, and inferencing. Both external and internal implementations of the semantic model can be attached to the GMAF, which opens a varied range of extensions to existing MMIR applications and standards. In addition, we showed, that our internal default algorithms are also highly effective, and can serve as a solid basis for further implementations. In our evaluation of both image- and text-based datasets, the results of our *Semantic Graph Code* algorithms and the corresponding concepts give evidence for our modeling approach. Hence, *Semantic Graph Codes* are an effective and efficient foundation for automated reasoning and inferencing for any MMIR application.

However, there are still some remaining challenges: the presentation of inferencing conflicts to the user, the implementation of additional integration with existing SKOS systems, further implementation of language models, and the evaluation of our implementation with further datasets. This will be part of our ongoing and future work, which is also frequently updated in our GitHub repository [9].

Author Contributions: Conceptualization and methodology: Stefan Wagenpfeil and Matthias Hemmje. Software, validation, formal analysis, investigation, resources, data curation, writing: Stefan Wagenpfeil. Review, editing and supervision: Paul Mc Kevitt and Matthias Hemmje. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in [9].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kwasnicka H. **(2018)**, *Bridging the Semantic Gap in Image and Video Analysis*; Publisher: Springer, Berlin; ISBN: 978-3-319-73891-8
2. Clement J. **(2020)**, *Social media - Statistics & Facts*. Available online: https://www.statista.com/topics/1164/social-networks/ (accessed 23.08.2020)
3. Sony Electronics Inc. **(2021)**, *a7R IV 35 mm full-frame camera with 61.0 MP*. Available online: https://www.sony.com/electronics/interchangeable-lens-cameras/ilce-7rm4 (accessed 22.11.2021)
4. Xiaomi **(2021)**, *Redmi Note 10 Pro - The 108MP Voyager*. Available online: https://www.mi.com/global/product/redmi-note-10-pro/overview (accessed 22.11.2021)
5. The Washington Post **(2021)**, *Washington Post Archives*. Available online: https://www.washingtonpost.com (accessed 22.11.2021)
6. Domingue J. **(2011)**, *Introduction to the Semantic Web Technologies*. Available online: https://doi.org/10.1007/978-3-540-92913-0 (accessed 24.06.2021)

7. Wagenpfeil S. **(2021)**, *AI-Based Semantic Multimedia Indexing and Retrieval for Social Media on Smartphones*. Available online: https://www.mdpi.com/2078-2489/12/1/43 (accessed 01.03.2021)
8. Wagenpfeil S. **(2020)**, *GMAF Prototype*. Available online: http://diss.step2e.de:8080/GMAFWeb/ (accessed 23.08.2020)
9. Wagenpfeil S. **(2021)**, *Github Repository of GMAF and MMFVG*. Available online: https://github.com/stefanwagenpfeil/GMAF/ (accessed 28.11.2021)
10. Wagenpfeil S., Towards AI-bases Semantic Multimedia Indexing and Retrieval for Social Media on Smartphones
11. Beyerer J. **(2017)**, *Pattern Recognition - Introduction*; Publisher: Walter de Gruyter GmbH & Co KG, Berlin; ISBN: 978-3-110-53794-9
12. Kurland O. **(2018)**, *Fusion in Information Retrieval: SIGIR 2018 Half-Day Tutorial*. Available online: https://doi.org/10.1145/3209978.3210186 (accessed 23.06.2021)
13. Leveling J. **(2013)**, *Interpretation of Coordinations*. Available online: https://doi.org/10.1145/2484028.2484115 (accessed 28.09.2021)
14. Bhute A. **(2012)**, *Multimedia Indexing and Retrieval Techniques: A Review*; Journal: International Journal of Computer Applications, Volume 58, pp 35-42;
15. Lew M. **(2006)**, *Content-Based Multimedia Information Retrieval: State of the Art and Challenges*. Available online: https://doi.org/10.1145/1126004.1126005 (accessed 23.06.2021)
16. Hernandez-Gracidas, C. **(2010)**, *Data Fusion and Label Weighting for Image Retrieval Based on Spatio-Conceptual Information*; Journal: Adaptivity, Personalization and Fusion of Heterogeneous Information, pp 76-79;
17. Dufour R. **(2010)**, *Local and global models for spontaneous speech segment detection and characterization*; DOI: 10.1109/ASRU.2009.5372928;
18. Subrahmanian V. **(1998)**, *Principles of Multimedia Database Systems*; Publisher: Morgan Kaufmann Publishers, San Francisco; ISBN: 978-1-558-60466-7
19. Shih-Fu C. **(2001)**, *Overview of the MPEG-7 standard*; Journal: IEEE Transactions on Circuits and Systems for Video Technology, Volume 11, pp 688-695;
20. FFMpeg.org **(2020)**, *ffmpeg documentation*. Available online: http://ffmpeg.org (accessed 23.08.2020)
21. Mu X. **(2006)**, *Content-Based Video Retrieval: Does Video's Semantic Visual Feature Matter?*. Available online: https://doi.org/10.1145/1148170.1148314 (accessed 27.10.2021)
22. Wagenpfeil S. **(2021)**, *Graph Codes - 2D projections of Multimedia Feature Graphs for Fast and Effective Retrieval*; Journal: International Conference on Image and Video Retrieval;
23. yWorks GmbH **(2020)**, *yEd Graph Editor*. Available online: https://www.yworks.com/products/yed (accessed 23.08.2020)
24. Wagenpfeil S. **(2021)**, *Semantic Query Construction and Result Representation based on Graph Codes*. Available online: http://ceur-ws.org/Vol-2863/paper-06 (accessed 28.07.2021)
25. Kapetanios E. **(2002)**, *Query Construction through Meaningful Suggestions of Terms*; Book: Flexible Query Answering Systems, Publisher: Springer-Verlag, Berlin Heidelberg New York, ISBN: 978-3-540-36109-1, pp 226-239;
26. Kapetanios E. **(2002)**, *The design and implementation of a meaning driven data query language*; DOI: 10.1109/SSDM.2002.1029702, pp 20-23;
27. Hussain S. **(2016)**, *Transforming Natural Language Query to SPARQL for Semantic Information Retrieval*; Journal: International Journal of Engineering Trends and Technology, Volume 41, pp 347-350;
28. Jung H. **(2020)**, *Automated conversion from natural language query to SPARQL query*. Available online: https://doi.org/10.1007/s10844-019-00589-2 (accessed 24.11.2021)
29. W3C.org **(2020)**, *W3C Semantic Web Activity*. Available online: http://w3.org/2001/sw (accessed 23.08.2020)
30. Wong R. **(2008)**, *Automatic Semantic Annotation of Real-World Web Images*; Journal: IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 30, pp 1933-1944;
31. Ni J. **(2017)**, *Research on Semantic Annotation Based Image Fusion Algorithm*; Journal: Research on Semantic Annotation Based Image Fusion Algorithm, pp 945-948;
32. Asim M. **(2019)**, *The Use of Ontology in Retrieval: A Study on Textual*; Journal: IEEE Access, Volume 7, pp 21662-21686;
33. Beierle C. **(2019)**, *Methoden wissensbasierter Systeme - Grundlagen*; Publisher: Springer-Verlag, Berlin Heidelberg New York; ISBN: 978-3-658-27084-1
34. Aho A., *Compilerbau*; Publisher: Oldenbourg Wissenschaftsverlag, ISBN: 9783486252941
35. Evans J. *Thinking and Reasoning: A Very Short Introduction*; ISBN: 0198787251
36. Hauser R. *Principles of Computer Linguistics*; Publisher: Springer Publishing, Berlin Heidelberg; ISBN: 3-540-67187-0.
37. W3C **(2021)**, *SKOS Simple Knowledge Organisation System*. Available online: https://www.w3.org/2004/02/skos/ (accessed 28.05.2021)
38. Bochman A. **(2007)**, *Nonmonotonic Reasoning*. Available online: https://www.sciencedirect.com/science/article/pii/S1874585707800124 (accessed 24.10.2021)
39. Das A. **(2003)**, *Knowledge Representation*. Available online: https://www.sciencedirect.com/science/article/pii/B0122272404001027 (accessed 28.10.2021)
40. Reiter R. **(1980)**, *A logic for default reasoning*. Available online: https://www.sciencedirect.com/science/article/pii/0004370280900144 (accessed 20.09.2021)
41. Poole D. **(1988)**, *A logical framework for default reasoning*. Available online: https://www.sciencedirect.com/science/article/pii/000437028890077X (accessed 14.07.2021)

42. W3C.org **(2013)**, *SPARQL Query Language for RDF*. Available online: https://www.w3.org/TR/sparql11-overview/ (accessed 23.08.2020)
43. Wagenpfeil S. **(2021)**, *Fast and Effective Retrieval for Large Multimedia Collections*. Available online: https://www.mdpi.com/2504-2289/5/3/33 (accessed 23.07.2021)
44. Gamma E. **(1994)**, *Design Patterns - Elements of Reusable Object Oriented Software*; Publisher: Addison Wesley, ISBN: 0-201-63361-2
45. Wikidata.com **(2021)**, *Wikidata - the free knowledgebase*. Available online: https://www.wikidata.org/wiki/Wikidata:Main_Page (accessed 24.10.2021)
46. Vrandecic D. **(2014)**, *Wikidata: A Free Collaborative Knowledge Base*. Available online: http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext (accessed 01.09.2021)
47. Text Retrieval Conference **(2021)**, *Datasets*. Available online: https://trec.nist.gov/data.html (accessed 14.09.2021)
48. Young P. **(2016)**, *From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions*. Available online: http://shannon.cs.illinois.edu/DenotationGraph/ (accessed 23.04.2021)
49. LSTM Model **(2021)**, *Long short-term memory*. Available online: https://en.wikipedia.org/wiki/Long_short-term_memory (accessed 28.11.2021)