A Case Study of NLG from Multimedia Data Sources: Generating Architectural Landmark Descriptions

Simon Mille[1] ✉, **Spyridon Symeonidis**[2] ✉, **Maria Rousi**[2], **Montserrat Marimon Felipe**[1], **Klearchos Stavrothanasopoulos** [2], **Petros Alvanitopoulos**[2], **Roberto Carlini**[1], **Jens Grivolla**[1], **Georgios Meditskos**[2], **Stefanos Vrochidis**[2], and **Leo Wanner**[1,3]

[1]Universitat Pompeu Fabra, Barcelona, Spain
✉ Corresponding author UPF: simon.mille@upf.edu
[2]Information Technologies Institute - CERTH, Thessaloniki, Greece
✉ Corresponding author ITI-CERTH: spyridons@iti.gr
[3]Catalan Institute for Research and Advanced Studies (ICREA)

Abstract

In this paper, we present a pipeline system that generates architectural landmark descriptions using textual, visual and structured data. The pipeline comprises five main components: (i) a textual analysis component, which extracts information from Wikipedia pages; (ii) a visual analysis component, which extracts information from copyright-free images; (iii) a retrieval component, which gathers relevant ⟨property, subject, object⟩ triples from DBpedia; (iv) a fusion component, which stores the contents from the different modalities in a Knowledge Base (KB) and resolves the conflicts that stem from using different sources of information; (v) an NLG component, which verbalises the resulting contents of the KB. We show that thanks to the addition of other modalities, we can make the verbalisation of DBpedia triples more relevant and/or inspirational.

1 Introduction

The bulk of the information reaches the reader nowadays across different media. Most of the videos uploaded to YouTube come accompanied by written natural language comments and so do most of the audio podcasts; even (online) newspaper articles can be hardly imagined without any visual illustrative material. This means that to generate a comprehensive but, at the same time, not partially repetitive, content summary of the provided information, the input of all media needs to be taken into account and merged.

The value of the merge of complementary information from multiple media for the generation of more informative texts has been pointed out already early in the field; cf., e.g., (Huang et al., 1999). However, since then only a few works tackled the problem; see, e.g., (Das et al., 2013; Xu et al., 2015). A few others merge multimedia content in the context of other tasks such as retrieval; cf. (Clinchant et al., 2011). In most of these works, the integration is done using similarity measures in a multimedia vector space. To the best of our knowledge, none aims for integration (or fusion) at, and subsequent generation from, the level of ontological <*subj predicate obj*> triples – which is crucial in order to be able to generalize, use inheritance or apply advanced reasoning mechanisms.

In this paper, we present a work that addresses this challenge. It fuses triples from DBpedia, textual information from Wikipedia, and visual information obtained from images as input to a pipeline-based generator. Our work is situated in the context of a larger research initiative, in which the objective is to automatically reconstruct architectural landmarks in 3D, such that they can be used by architects, game designers, journalists, etc. Each reconstructed landmark should be accompanied by automatically generated information that describes its main features. The goal is to convey information such as the landmark's architect, some of its facade elements, its date of construction and/or renovation, its architectural style, etc. in terms of a text like:

> *Petronas Towers, which César Pelli designed, are commercial offices and a tourist attraction in Kuala Lumpur. The building has 88 floors and 40 elevators and a floor area of 395,000 m^2. It was the highest building in the world between 1998 and 2004, and was restored in 2001.*

which is a verbalisation of the triples in Table 1.

In what follows, we describe how this is done, from content extraction to the use of a grammar-based text generator. We thus do not focus exclusively on text generation. Rather, we attempt to show how richer input structures can be created using different media for the benefit of more com-

3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), Dublin, Ireland (Virtual), 18 December 2020, pages 2–14,

Petronas Towers	location	Kuala Lumpur
Petronas Towers	restorationDate	2001
Petronas Towers	floorArea	395,000
Petronas Towers	floorCount	88
Petronas Towers	elevatorCount	40
Petronas Towers	buildingType	commercial offices
Petronas Towers	buildingType	tourist attraction
Petronas Towers	highestRegion	world
Petronas Towers	highestStart	1998
Petronas Towers	highestEnd	2004
Petronas Towers	architect	César Pelli

Table 1: A set of <*subj predicate obj*> input triples

prehensive and user-relevant texts.

2 Related work

As already pointed out above, to the best of our knowledge, only a few works deal with fusion of content from different media as input representation for downstream applications (in our case, text generation) and when they do, they use similarity measures in a multimedia vector space (Huang et al., 1999; Clinchant et al., 2011; Das et al., 2013; Xu et al., 2015) rather than mapping multimedia content onto a common ontology. This does not mean though that research related to text generation across different media would be neglected. For instance, generation of image captions (Hossain et al., 2019) and video descriptions (Aafag et al., 2019) has recently become a very popular research topic. All of the proposals in this area use sequence-to-sequence neural network models. In (Idaya Aspura and Azman, 2017), indexes of textual and visual features are integrated via a multi-modality ontology, which is further enriched by DBpedia triples for the purpose of semantics-driven image retrieval. On the other side, text generation from ontological structures is on the rise; cf., e.g., (Bouayad-Agha et al., 2014; Gatt and Krahmer, 2018) for overviews and the WebNLG challenge (Gardent et al., 2017a) for state-of-the-art works.

In general, there are three main approaches to generating texts from ontologies: (i) filling slot values in predefined sentence templates (McRoy et al., 2003), (ii) applying grammars that encode different types of linguistic knowledge (Varges and Mellish, 2001; Wanner et al., 2010; Bouayad-Agha et al., 2012; Androutsopoulos et al., 2013), and (iii) predicting the most appropriate output based on machine learning models (Gardent et al., 2017b; Belz et al., 2011). Template-based generators are very robust, but also limited in terms of portability since new templates need to be defined for every new domain, style, language, etc. Machine learning-based generators have the best coverage, but the relevance and the quality of the produced texts cannot be ensured. Furthermore, they are fully dependent on the available (still scarce and mostly monolingual) training data. The development of grammar-based generators is time-consuming and they usually have coverage issues. However, they do not require training material, allow for a greater control over the outputs (e.g., for mitigating errors or tuning the output to a desired style), and the linguistic knowledge used for one domain or language can be reused for other domains and languages. A number of systems also combine (i) and (iii), filling the slot values of pre-existing templates using neural network techniques (Nayak et al., 2017).

In what follows, we opt for a grammar-based generator. We show that information from visual, textual and structured (DBpedia) sources can be successfully fused in order to generate informative descriptions using a pipeline-based text generator.

3 System and dataset overview

Let us first introduce the architecture of our system and then outline the creation of the datasets used for development and testing.

3.1 General system architecture

The workflow of our system is illustrated in Figure 1. The initial input is the topic entity on which the text is to be generated. Based on this, the *data collection* module harvests relevant content from the Web. The resources of interest are images from the Flickr website and texts from Wikipedia, which are processed by the *visual* and *textual analysis* modules respectively. The two modules extract a rich set of features that describe the entity. The *knowledge integration and reasoning* module stores the extracted visual and textual features along with additional metadata retrieved from DBpedia in dedicated ontologies. Semantic reasoning and fusion operations are subsequently executed on top of the saved data to aggregate the information coming from the different media into a unified entity representation. *Text generation* starts from this representation in order to generate a textual description.

3.2 Development and test datasets

The targeted entities are architectural landmarks such as buildings, statues or stadiums. The goal is to be able to generate a description which con-

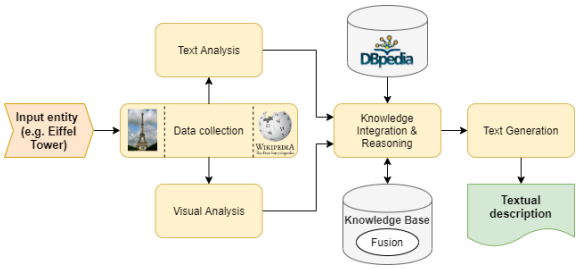

Figure 1: System architecture.

tains, e.g., the date of creation, the location, the architecture style or the popularity.

To create the sufficiently diverse datasets, we first manually compiled a list of 160 landmarks that vary in terms of the aforementioned characteristics. In the next step, we retrieved the available multimedia content on these landmarks: images (from Flickr), textual descriptions (from Wikipedia) and ontological properties (from DBpedia)[1] and selected then a subset of 120 landmarks that had either rich image or DBpedia contents, or both. We used 101 landmarks to develop and optimise our framework, whereas 19 randomly selected landmarks were left aside for the evaluation stage. The full list of landmarks is available in Appendix A.1.

4 Multimedia content acquisition

In this section, we describe the content that we extracted from the three sources (DBpedia, images and texts) used for generating the landmark descriptions, and how we extracted it.

4.1 DBpedia triple retrieval

DBpedia contains a lot of information that is potentially relevant to the description of architectural landmarks. We analysed manually the DBpedia entries of the 101 landmarks in the development set in order to see which properties are related to the landmark and its architectural features.[2] We identified 39 features of interest, most of which are consistently found across the landmarks in the list, among them, e.g., features related to the type of the landmark, its style, who built it, the dates of its construction, renovation, or extension, its location, its construction materials, its cost, its number of floors, elevators, or towers, etc. On average, about 6 features per landmark can be obtained through DBpedia (up to 13 for a single landmark). The information that corresponds to one feature can be encoded by a variety of property names (up to 10). For instance, the cost of a building can be expressed by *dbo:cost*, *dbp:cost*, or *dbp:constructionCost*.[3] The 39 features and the corresponding 98 properties are listed in Table 7, Appendix A.2.

For the retrieval of the corresponding DBpedia triples, we developed a component that applies SPARQL queries to the DBpedia SPARQL endpoint.[4] In some rare cases, the queries returned an error message at the time they were performed; in such a case, the property cannot be accessed and the information is not retrieved.

4.2 Visual content acquisition

The performed visual analysis for the purpose of visual content acquisition is twofold. First, an object detection module classifies indoor and outdoor scenes and detects landmark (in this case, building) elements, and objects. Second, an architectural style classification module assigns the related architectural style label to each outdoor scene of the selected dataset. Both classification modules are based on state-of-the-art deep learning techniques.

4.2.1 Visual scene classification and labeling

For visual scene classification, we draw upon the 145 relevant indoor and outdoor scene classes from the Places dataset (Zhou et al., 2018), which contains 1,803,460 images, annotated with a total of 365 classes. The classifier is a VGG16 deep neural network (Simonyan and Zisserman, 2014), pre-trained on the Places dataset for the first 14 layers (for all of its 365 classes) and fine-tuned on a subset of 145 selected classes for the last two layers.

For labeling the landmark elements and objects in the classified scenes, we use a Deeplab model, pre-trained on the PASCAL VOC (Everingham et al., 2009) dataset, where further training was applied using a combination of building façade segmentation datasets (Mapillary Vistas (Neuhold et al., 2017), CMP (Tylecek and Sára, 2013), ECP,[5] LabelMeFacade (Frohlich et al., 2010), eTRIMS

[1] While the resources on Wikipedia and DBpedia are free for use, we had to pay special attention to image collection from Flickr to ensure that we gather media whose license permits their reuse for our purposes.

[2] See for illustration the Petronas Towers page: http://dbpedia.org/page/Petronas_Towers.

[3] There are two main types of properties in DBpedia: "clean" properties that stem from the DBpedia ontology (*dbo:* prefix), and properties automatically extracted from raw Wikipedia infoboxes (*dbp:* prefix).

[4] http://dbpedia.org/sparql

[5] http://vision.mas.ecp.fr/Personnel/teboul/data.php

(Korč and Förstner, 2009)).[6] This not only resulted in a computationally efficient implementation for the detection of architectural landmark-related artefacts, but also increased the classification accuracy of the model. In order to further improve object detection, we added a third module based on the Mask RCNN model. We initialized this module using pre-trained weights on COCO dataset (Lin et al., 2014) and then performed fine-tuning on a customized set created by merging LVIS (Gupta et al., 2019) and ADEK20K (Zhou et al., 2016) datasets and removing all classes irrelevant to the scope of our task. Details about the training settings are provided in Appendix A.3.

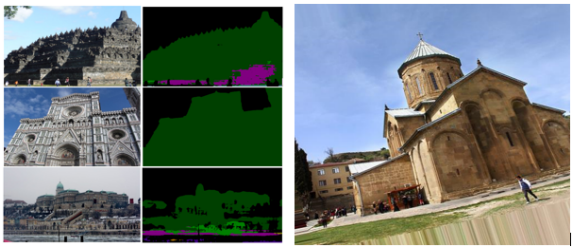

Figure 2: (L)Visual of the detection module's results. The algorithm detects the building along with the surroundings and generates the corresponding predicted tags. (R) Architecture style recognition, the model predicted label : Romanesque , True label : Romanesque.

4.2.2 Landmark style identification

In the context of this task, the visual analysis component aims to assign to a landmark one of the 18 architectural styles listed in Table 2 and to identify the following seven additional types of features that are later on used for text generation: (i) construction type (e.g., 'amphitheater', 'castle', 'hotel'); (ii) similarities with other construction types (same list as (i)); (iii) similarity of a part of the construction with another construction (e.g., 'bridge', 'arch'); (iv) facade elements (e.g., 'balcony', 'fire escape'); (v) interior components and objects (e.g., 'fireplace', 'elevator shaft', etc.); (vi) environment (e.g., 'downtown', 'village', 'park'); (vii) proximity to a natural landmark (e.g., 'river', 'park'). The full list of detected features is provided in Table 8, Appendix A.3. In total, eight different properties are extracted for generation, two of which (style, construction type) are fused with the properties obtained through the other modalities, and the other ones used for text generation as such.

A list of the supported Architectural styles			
Art Deco	Art Nouveau	Baroque	Bauhaus
Biedermeier	Corinthian Order	Deconstructuvism	Doric Order
Early Roman	Gothic	Hellinistic	Ionic Order
Modernist	Neoclassical	Postmodernism	Renaisance
Rococo	Romanesque		

Table 2: Architectural Styles

For training of the model for landmark style and feature identification, images from Flickr, Europeana and Wiki were collected. Annotators with architectural expertise annotated the collected data. A set of 11,368 newly annotated images was used for training purposes, while a total of 1,276 newly annotated images were used for testing. VGG16 and ResNet50 models were enhanced with 3 layers (one GlobalAveragePooling2D and two Dense layers) and initialised with the pre-trained ImageNet weights. For better training, K-Fold Validation and Stratified Shuffle Split were applied.

4.3 Textual content acquisition

In addition to using visual features, we enriched the DBpedia information by entity-relation-entity triples extracted from the unstructured part of the Wikipedia articles. This is done using a pipeline that comprises concept detection, entity linking and WSD to identify the proper entities (linking to DBpedia URIs); then, PoS tagging, dependency parsing and coreference resolution to generate surface-syntactic structures and to link mentions of entities in the different parts of text; and, finally, semantic parsing to generate deep-syntactic structures from which we extract the triples.

In what follows, we outline the types of information we aim to extract from the textual data and how we do it.

4.3.1 Targeted information in textual data

Unlike the content extracted from visuals, which is not expected to be found in DBpedia since it is related to specific images and some "subjective" features, the information extracted from Wikipedia is supposed to be already captured in DBpedia. However, we observe that some of the relevant properties are often missing. The goal of the textual analysis component is thus to recover these missing properties, which concern, in particular, the type of the landmark, its date(s) of construction and renovation, its location, its architectural style and its architect, designer or creator. In order to reduce

[6] These datasets contain up to 25,000 high-resolution images annotated with a variety of semantic classes and possibly instance-specific labels.

the load on textual analysis, we analyse only the first paragraphs of the scraped Wikipedia articles.

As an example, consider the text *"Rouen Cathedral is a Roman Catholic church in Rouen, Normandy, France. It is the seat of the Archbishop of Rouen, Primate of Normandy. The cathedral is in the Gothic architectural tradition."*, for which the following triples are extracted:

Rouen Cathedral	Localisation	Roman Catholic church
Rouen Cathedral	Location	Rouen, Normandy, France
Rouen Cathedral	Style	Gothic architectural tradition

4.3.2 Triple extraction from texts

In order to extract the targeted triples, we apply a sequence of rule-based graph transducers on the output of an off-the-shelf syntactic parser. More specifically, we run the pipeline used for creating the deep input representations of the Surface Realisation shared tasks 2018 and 2019 (Mille et al., 2018), with one additional component responsible of identifying the configurations that correspond to the targeted information. Consider, for illustration, a sample rule in Figure 3, which extracts the 'Rouen Cathedral – Localisation – Roman Catholic church' triple from the predicate-argument (PredArg) structure as encoded by light verb *be*-constructions, where the first argument of the light verb *be* becomes the first argument of the predicate in the PredArg representation.

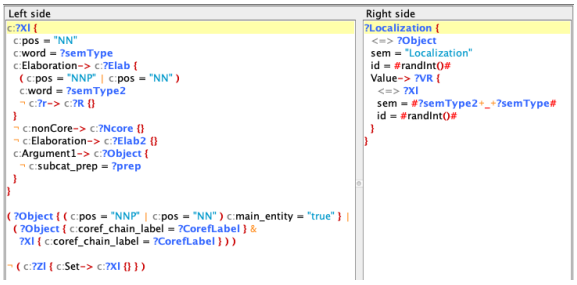

Figure 3: A sample rule to extract triples. The LeftSide matches a part of the input tree, the RightSide builds part of the output. Three types of objects are used: nodes (?N{}), relations between nodes (?r→) and attribute-value pairs associated to a node (?a = b), where question marks indicate variables and text in black indicates literal strings.

4.4 Quantitative analysis of the information acquisition modules.

We evaluated the text analysis component with a set of 16 texts and measured average values of 83% for precision and 40% for recall on the triple extraction for the targeted triples (see Section 4.3). In other words, the coverage of the module needs to be extended to get more information, and some incorrect values would need to be filtered. The main limitations here are the difficulty in covering the wide variety of surface syntactic structures, and the quality of syntactic parses.

For the evaluation of the architectural style classification model a set of building images were selected. The dataset for testing comprises 1,276 images and includes all the 18 architectural styles. The F1 score was taken into consideration and a significant 46.16% of correct classification was performed, similar to the SoA results (Z. et al., 2014); the confusion matrix for the architecture style classification is shown in Figure 7 in Appendix A.3. Even though the classification is state-of-the-art, in more than 50% of the cases the architectural style is wrong, which is one of the main comments from the evaluators in terms of incorrectness of contents (see Section 7.4).

5 Multimedia information fusion

The results of the visual and textual analyses and the retrieved DBpedia properties are mapped using the Web Annotation Data Model[7]. The model creates a body and a target for each annotation.

As main interconnection point between the content from different media, we use the name of the corresponding entity, which is mapped as the target of the annotation. The body of the annotation contains all the other information, which varies according to the nature of each input module. More specifically: (i) for the visual analysis content, the body contains information pertinent to scene, objects, façade, structure elements and architectural styles. A mapping example is shown in Figure 4; (ii) for the Wikipedia analysis outcome, the body contains creator and localisation information related to the entity; (iii) for the retrieved DBpedia triples, the information in the body is pertinent to the landmark, the architecture, the location and more general information about the landmark.

A reasoning mechanism applies the following property-based semantic rules at the time of the retrieval of DBpedia triples: (i) Extraction of class information about creators and locations: the mechanism detects whether a creator is a person, an organisation or a company, and whether a location is

[7] https://www.w3.org/TR/annotation-model/

```
@prefix examples: <https://v4design.eu/ontologies/examples#> .
@prefix oa: <http://www.w3.org/ns/oa#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix v4d: <https://v4design.eu/ontologies/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

examples:VisualAnnotation_1 a v4d:VisualAnnotation;
  oa:hasBody examples:VisualView_1;
  oa:hasTarget examples:VisualFeature_1.

examples:VisualView_1 a v4d:VisualView;
  v4d:containsImage v4d:Image_1 .

examples:VisualFeature_1 a v4d:VisualFeature;
  v4d:isRelatedWith "Alhambra" .

v4d:Image_1 a v4d:Image;
  v4d:hasArchitecturalStyle v4d:ArchitecturalStyle_1;
  v4d:hasFacadeElement v4d:FacadeElement_1;
  v4d:hasObject v4d:Object_1;
  v4d:hasScene v4d:Scene_1;
  v4d:hasStructureElement v4d:StructureElement_1;
  v4d:imageName "24.jpg" .

v4d:Scene_1 a v4d:Scene;
  rdfs:label "palace";
  v4d:hasGenericClass "http://www.semanticweb.org
                       /inlg-ontology#building";
  v4d:isOutdoor "true";
  v4d:probability "0.4205022" .

v4d:StructureElement_1 a v4d:StructureElement;
  rdfs:label "structure--building";
  v4d:hasGenericClass "http://www.semanticweb.org
                       /inlg-ontology#building";
  v4d:probability "0.4401882570316443" .

v4d:FacadeElement_1 a v4d:FacadeElement;
  rdfs:label "door";
  v4d:probability "0.5884110748255492" .

v4d:Object_1 a v4d:Object;
  rdfs:label "signboard";
  v4d:probability "0.9469741" .

v4d:ArchitecturalStyle_1 a v4d:ArchitecturalStyle;
  rdfs:label "Baroque";
```

Figure 4: Example of mapping Visual data

a region, a city or a country. (ii) Unit detection: in case that DBpedia information contains the concept of monetary cost, extracting the currency provides the corresponding information to Text Generation. The same rule is applied for literals. (iii) Filtering of undesired values: for instance 'buildingType' cannot contain values such as 'Cultural', 'style' cannot contain an affirmation of type "yes", etc. (iv) Retrieval of one or more values according to the property category: for example, for properties such as 'buildStartDate', if more than one results are found, only one is returned, while for properties like 'materials', if more than one results are found, all of them are returned.

During the fusion procedure, the content obtained from Wikipedia, DBpedia and images are merged per entity. For visuals, since for each entity the results are analysed per image, we return the five values that have maximum occurrences in the images collection per category i.e., "scene", "object", 'façade'" and "structure elements". For the information that belongs to the same category and comes from different modules (e.g., type of building, creator, architectural style), we select the most frequent entities, or if there is none, we use the information from DBpedia or pick one randomly. The properties that are fused and the analysis module they come from are shown in Table 3. At the end of the fusion procedure, the results contain both the information from the individual modules and the fusion selection.

Property	Text	DBpedia	Visual
Building type	Localisation	hypernyms and buildingTypes	scene recognition features
Creator	creator	creator	-
Architectural style	-	style	architectural style

Table 3: Fused properties from different sources

6 Generation of landmark descriptions

Despite the advances in neural NLG, grammar-based generation is still a valuable option when training data are scarce and/or when a large coverage grammar-driven generator is already available.

6.1 Grammar-based generation

No annotated datasets of architectural landmark descriptions to train machine learning-based models or to extract sentence templates for template-based generation are available. Therefore, we tackle description generation from the fused ontological triples presented above using FORGe, a portable grammar-based generator that has been adapted to structured data inputs, in particular DBpedia triple sets (Mille et al., 2019). The input triples are individually mapped to minimal predicate-argument templates (see Figure 5), which are then sent to the generator. The generation consists of a sequence of graph-transduction grammars that map successively the PredArg templates to linguistic structures of different levels of abstraction, in particular syntax, topology, morphology, and finally texts. PredArg structures are very similar to the *Facts* in ILEX's Content potential structures (O'Donnell et al., 2001), or the *Message triples* in NaturalOWL (Androutsopoulos et al., 2013), with the difference that all predicates in the PredArg structures are intended to represent atomic meanings (e.g. *highest* + *building* as opposed to *highestBuilding*), allowing for more flexible aggregation and sentence structuring. The first part of the generation pipeline, which produces aggregated predicate-argument graphs, is also comparable to ILEX, while the surface realisation is largely inspired by MARQUIS (Wanner et al., 2010). Our generator shares not only its general architecture with these two systems, but also the use of lexical resources with subcategorisation information and of a multilingual core of rules. One

of the specificities of our pipeline is that two types of aggregation take place during generation, one at the predicate-argument level (in a NaturalOWL fashion), and one at the syntactic level (see below).

6.2 Extension of an existing generator

The base generator covers about 400 DBpedia properties, but only a few had to do with architectural landmarks, and generation of up to only 10 triples had been tested. In this work, the inputs can contain up to 19 triples, and most of the properties are new. We thus extended the coverage of the generator according to two main aspects: (i) addition of 38 manually crafted PredArg templates, (ii) addition of domain-specific "semantic" aggregation rules.

For (i), the 38 new properties[8] were each associated with a new PredArg template; see, for instance, the templates corresponding to the 'highestEnd' and 'interiorComponent' properties in Figure 5. Instantiating the template 5 (a) with the values of Table 1 ([name] = Petronas Towers, [highestEnd] = 2004) and generating it would result in the sentence *The Petronas Towers were the highest building in the world until 2004*. Template 5 (b) would be realised as *There is an elevator shaft in P. Towers*.

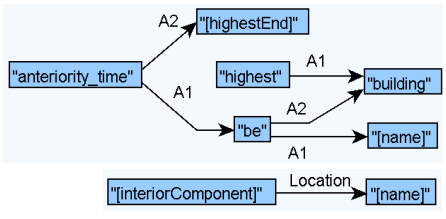

Figure 5: Sample predicate-argument templates (a) = DBpedia, (b) = visuals

For (ii), we designed new aggregation rules to complement the generic rules already in place, which are based on the identity between predicates and/or entities only. In particular, properties that involve dates need to be aggregated in a specific way when there are both a start and an end date. For instance, *highestStart* and *highestEnd* as seen in Table 1 trigger a rule that introduces a *between*: *the highest building in the world between 1998 and 2004*. In parallel, some other rules aggregate some properties in priority if found in the input: [style + date] $\succ$ [creator + style] $\succ$ [generic rules].

Specific rules (e.g., for linearisation) were also improved to cover the generation of texts that are not only larger due to the input size, but also more complex due to the more complex syntactic constructions (e.g., non-projective trees, as in *highest building in the world*). Finally, we crafted a new syntactic aggregation module, which aggregates coordinated and relative clauses based on identity of syntactic subjects/objects, locations and verbs.[9]

7 Evaluation

We evaluate the quality of the generated descriptions from fused representations, first of all, against monomodal descriptions generated solely from DBpedia triples. The goal is to assess to what extent architectural landmark descriptions benefit from additional content from other media. A comparison with Wikipedia texts is also carried out.

7.1 Evaluation method

Six journalists, architects and architectural landmark content providers were recruited for the evaluation.[10] They were asked to evaluate descriptions with respect to their correctness of form and content and their level of interest by rating the following statements on a 6-value Likert scale: [11]

Correctness of Form: *Independently of Correctness of content and Interestingness, (i) the surface form of the text is free of grammatical and spelling errors, (ii) the text is easy to read and understand, and (iii) it flows well.*

Correctness of Content: *Independently of Correctness of form and Interestingness, and using only my current knowledge on the topic, I do not identify information that looks obviously incorrect.*

Interestingness: *Independently Correctness of form and content, I find the information provided in the text interesting, relevant and inspirational.*

The evaluation test set consisted of the descriptions of 19 landmarks: 19 descriptions generated from fused multimedia representations, 19 generated from DBpedia triples and the first or second paragraph (whichever was the most informative in terms of architectural descriptions) from the Wikipedia articles on the 19 landmarks in question. For each building, all evaluators were presented with the three descriptions, in a random order, and

[8] 32 from DBpedia (7 out of the 39 were already covered; see Table 7, Appendix A.2) , and 6 coming from the visual content analysis (see Section 4.2)

[9] In the case of the 'interiorComponent' property, as seen in Figure 5, there is no verb at the semantic level; it is only introduced in the syntactic structure. The syntactic aggregation module covers such cases.

[10] All evaluators are fluent in English and familiar with the described landmarks.

[11] Answers from 1: strongly disagree to 6: strongly agree.

scored each description. That is, each system received 114 ratings for each of the 3 criteria, which we believe makes the evaluation trustworthy.

7.2 Evaluation against DBpedia-based descriptions

The comparison between the descriptions generated from fused multimedia representations and the descriptions generated from DBpedia triples (see Table 4) is shown in Figure 6. A 2-tailed Mann-Whitney U test indicates that only for Form the difference is not statistically significant at $p < 0.05$.

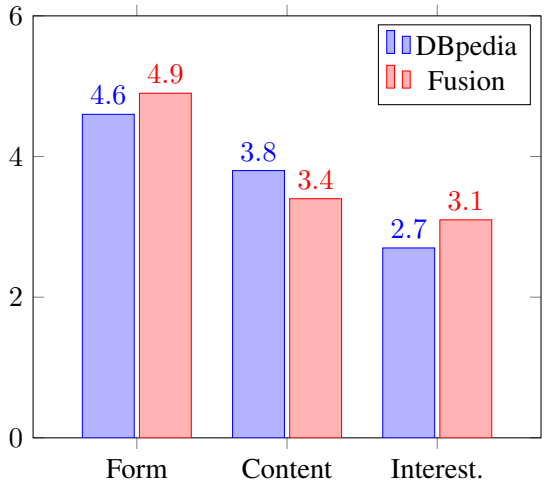

Figure 6: Results of the human evaluation

7.3 Evaluation against Wikipedia

The 6 evaluators were also asked to rate the Wikipedia articles. In most cases, one single Wikipedia paragraph was longer and richer than either of our generated descriptions, so the texts are not fully comparable, but our objective has been to define some upper bound scores for a short text. Wikipedia paragraphs scored 5.5, 5.3 and 4.4 for the correctness of form, of contents and interestingness respectively, that is, 0.6, 1.9 and 1.3 points higher than our fused descriptions. We also asked the evaluators to pick which text they preferred among the 3 candidates, and interestingly, Wikipedia articles were not always chosen: in 15 cases out of 114, an automatically generated text was picked (DBpedia: 4, Fusion: 11).

7.4 Discussion

Table 4 shows texts from the different sources. Figure 6 shows that while the scores for the correctness of form are rather high for both the fused and DBpedia descriptions (close to 5), the scores for the other

Wikipedia (human)
The Sydney Opera House is a multi-venue performing arts centre at Sydney Harbour in Sydney, New South Wales, Australia. It is one of the 20th century's most famous and distinctive buildings.
DBpedia
Sydney Opera House, which Jørn Utzon designed, is a Performing arts center in Sydney. Sydney Opera House, the architectural style of which is Expressionist architecture, was built between 1 March 1959 and 1973. Its structure is made of Concrete frame & precast concrete ribbed roof.
Fused
Sydney Opera House, which Jørn Utzon designed, is a **centre** in Sydney. Its structure is made of Concrete frame & precast concrete ribbed roof. **An element of the structure is like a bridge. Sydney Opera House has similarities with a beach house**, **an amusement park and a museum**. Sydney Opera House, the architectural style of which is **Deconstructuvism**, was built between 1 March 1959 and 1973.

Table 4: Sydney Opera house descriptions (eval set)

two criteria are low, in particular for interestingness (2.7 and 3.1). However, adding information from different sources does increase slightly the interestingness of the texts, and even their correctness in terms of form, but at the expense of the correctness of the contents.

The lower scores obtained for interestingness (when compared to the other two criteria) even for the human-written Wikipedia texts highlight the difficulty for short texts to be considered interesting and inspirational. But the fact that the automatically generated texts score more than one point lower than Wikipedia shows that the content representation of the 44 features is still not sufficient; more content needs to be provided. Other DBpedia properties such as 'owner', 'tenant', or information related to other landmarks or persons such as 'architecturalStyle (of)', 'birthPlace (of)', 'influencedBy', 'location (of)', etc. could augment the interestingness of the descriptions.

The low scores of our generator for the correctness of the contents, in particular for the fused descriptions (3.4) are due to several causes. First of all, as shown by the scores of the DBpedia-only descriptions (3.8), not all the information in DBpedia is factually correct, in particular the information extracted automatically from Infoboxes: buildings can be assigned types such as "Series", "Nickname"or "Mixed-use" (see Tables 9 and 11, A.4); construction dates can be irrelevant ("between 2015 and 532"); locations sometimes refer to a relative location (e.g., "right"), etc. Second, the information extracted from texts and visuals, tasks which are traditionally difficult to solve, is also not perfect (a detailed error analysis is provided in Section

4.4); incorrect architectural styles (e.g., Tables 10 and 12, A.4) and comparisons between supposedly similar buildings (see Table 10, A.4) were found particularly disconcerting by the evaluators. Finally, the performance of the fusion component is currently heavily dependent on the cases seen in the development dataset. In the development set, in most cases, the selected entities were valuable and supported the Text Generation as expected, but in the evaluation set, many cases had not been seen, such that ill-informed decisions were taken, sometimes triggering the replacement of a correct value from DBpedia by an incorrect value from visual or textual analysis (see Table 4 and Tables 9 and 12, A.4). A larger development set would be needed in order to identify more erroneous configurations. Another solution may be more generic strategies to foresee the possible mistakes in the inputs.

8 Conclusions

We presented the case of the generation of architectural landmark descriptions from ontological structures that contain fused content from visual, textual and ontological sources. The evaluation showed that when compared to descriptions generated from the DBpedia RDF-triples obtained from textual material only (i.e., Wikipedia), descriptions that communicate fused content are considered more interesting and better in terms of textual quality. However, also due to the limited content features that were considered in the experiments, these descriptions cannot compete, in general, with more comprehensive well-written descriptions as encountered in Wikipedia. Still, it needs to be taken account that by far not all architectural landmarks that are of interest from the professional or cultural viewpoint are covered by Wikipedia. Fused content descriptions are then a welcomed solution.

Acknowledgements

This work was supported by the European Commission in the context of its H2020 Program under the grant numbers 870930-RIA, 779962-RIA, 825079-RIA, 786731-RIA at Universitat Pompeu Fabra and Information Technologies Institute - CERTH.

References

Nayyer Aafag, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. 2019. Video description: A survey of methods, datasets and evaluation metrics. *ACM Computing Surveys*, 52(6).

Ion Androutsopoulos, Gerasimos Lampouras, and Dimitrios Galanis. 2013. Generating natural language descriptions from owl ontologies: the naturalowl system. *Journal of Artificial Intelligence Research*, 48:671–715.

Anja Belz, Mike White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first Surface Realisation Shared Task: Overview and evaluation results. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation (ENLG)*, pages 217–226, Nancy, France.

Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, Marco Rospocher, Horacio Saggion, Luciano Serafini, and Leo Wanner. 2012. From ontology to nl: Generation of multilingual user-oriented environmental reports. In *International Conference on Application of Natural Language to Information Systems*, pages 216–221. Springer.

Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. 2014. Natural language generation in the context of the semantic web. *Semantic Web*, 5(6):493–513.

Stéphane Clinchant, Julien Ah-Pine, and Gabriela Csurka. 2011. Semantic combination of textual and visual information in multimedia retrieval. In *Proceedings of the 1st ACM international conference on multimedia retrieval*, pages 1–8.

Pradipto Das, Rohini Kesavan Srihari, , and Jason J. Corso. 2013. Translating related words to videos and back through latent topics. In *Proc. of WSDM*.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2009. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–308. Printed version publication date: June 2010.

Bjorn Frohlich, Erik Rodner, and Joachim Denzler. 2010. A fast approach for pixelwise labeling of facade images. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, page 3029–3032, USA. IEEE Computer Society.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating training corpora for micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Agrim Gupta, Piotr Dollár, and Ross Girshick. 2019. Lvis: A dataset for large vocabulary instance segmentation.

Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys*, 51(6).

Qian Huang, Zhu Liu, Aaron Rosenberg, David Gibbon, and Behzad Shahraray. 1999. Automated generation of news content hierarchy by integrating audio, video, and text information. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 6, pages 3025–3028. IEEE.

Yanti Idaya Aspura and Shahrul Azman. 2017. Semantic text-based image retrieval with multi-modality ontology and dbpedia. *The Electronic Library*.

Filip Korč and Wolfgang Förstner. 2009. eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Susan W McRoy, Songsak Channarukul, and Syed S Ali. 2003. An augmented template-based approach to text realization. *Natural Language Engineering*, 9(4):381.

Simon Mille, Anja Belz, Bernd Bohnet, and Leo Wanner. 2018. Underspecified Universal Dependency Structures as Inputs for Multilingual Surface Realisation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 199–209, Tilburg, The Netherlands.

Simon Mille, Stamatia Dasiopoulou, Beatriz Fisas, and Leo Wanner. 2019. Teaching FORGe to verbalize DBpedia properties in Spanish. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 473–483, Tokyo, Japan. Association for Computational Linguistics.

Neha Nayak, Dilek Hakkani-Tür, Marilyn A Walker, and Larry P Heck. 2017. To plan or not to plan? discourse planning in slot-value informed sequence to sequence models for language generation. In *Proceedings of INTERSPEECH*, pages 3339–3343, Stockholm, Sweden.

Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. 2017. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Mick O'Donnell, Chris Mellish, Jon Oberlander, and Alistair Knott. 2001. Ilex: an architecture for a dynamic hypertext generation system. *Natural Language Engineering*, 7(3):225.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition.

Radim Tylecek and Radim Sára. 2013. Spatial pattern templates for recognition of objects with regular structure. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 364–374. Springer Berlin Heidelberg.

Sebastian Varges and Chris Mellish. 2001. Instance-based natural language generation. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Leo Wanner, Bernd Bohnet, Nadjet Bouayad-Agha, Francois Lareau, and Daniel Nicklaß. 2010. MARQUIS: Generation of user-tailored multilingual air quality bulletins. *Applied Artificial Intelligence*, 24(10):914–952.

Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. 2015. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Xu Z., Tao D., Zhang Y., Wu J., and Tsoi A.C. 2014. Architectural style classification using multinomial latent logistic regression. In *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, vol 8689*, Cham. Springer International Publishing.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2018. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2016. Semantic understanding of scenes through the ade20k dataset.

A Appendices

A.1 List of buildings in the datasets

Tables 5 and 6 show the buildings used for development and evaluation respectively.

Alhambra	Arc de Triomphe
Belém Tower	Blue Church Bratislava
Borobudur Temple	Bran Castle
Brandenburg Gate	Bratislava Castle
Buckingham Palace	Buda Castle
Burj Al Arab	CN Tower
Canton Tower	Casa Batlló
Casa Milà	Castel Sant'Angelo
Catherine Palace	Chichen Itza
Chrysler Building	Château Frontenac
Château de Chenonceau	Cologne Cathedral
Colosseum	Dome of the Rock
Dresden Frauenkirche	Edinburgh Castle
Eiffel Tower	Elbphilharmonie
Empire State Building	Faisal Mosque
Fallingwater	Fisherman's Bastion
Florence Cathedral	Forbidden City
Gatchina Palace	Giza Pyramids
Grand Place Brussels	Harpa Concert Hall
Helsinki Cathedral	Heydar Aliyev Center
Himeji Castle	Jin Mao Tower
Kiev Pechersk Lavra	Knossos Palace
Konark Sun Temple	Kronborg Castle
Lincoln Center	Lincoln Memorial
Lloyd's Building	London Eye
Madrid Palace	Marina Bay Sands
Metropolitan Cathedral of Brasília	Milan Cathedral
Mosque of Córdoba	Musée d'Orsay
Niterói Contemporary Art Museum	Notre Dame
Odeon of Herodes Atticus	One World Trade Center
Oriental Pearl Tower	Palace of Versailles
Peles Castle	Pena Palace
Petra Jordan	Porta Nigra
Potala Palace	Prague Castle
Rouen Cathedral	Royal Liver Building Liverpool
Royal Observatory (Greenwich)	Sacré-Coeur
Sagrada Familia Cathedral	Space Needle
St. Basil's Cathedral	Statue of Liberty
Stonehenge	Sultan Ahmed Mosque
Taipei 101	Taj Mahal
Tech Tower	The Atomium
The Cristo Rei	The Flatiron Building
The Gherkin	The Guggenheim New York
The Lotus Temple	The Pantheon Rome
The Shard	The Sistine Chapel
The Temple of Olympian Zeus (Athens)	The White House
Tokyo Skytree	Tokyo Tower
Tower of Pisa	Villa Savoye
Wembley Stadium	Westminster Abbey
Wilanów Palace	Windsor Castle
Wuppertal Schwebebahn	

Table 5: The 101 buildings used for development

Angkor Wat	Big Ben
Burj Khalifa	Camp Nou
Christ the Redeemer	Dancing House Prague
Guggenheim Museum Bilbao	Hagia Sophia
Hungarian Parliament Building	Kremlin
Louvre	Machu Picchu
Neuschwanstein Castle	Parthenon
Peterhof Palace	Petronas Towers
Sydney Opera House	Walt Disney Concert Hall
White Tower Thessaloniki	

Table 6: The 19 buildings used for the evaluation

A.2 List of retrieved DBpedia properties

Table 7 lists the 39 features used for generation, roughly grouped by topic, and their correspondence

Features (count)	**Properties**
building type (54)	dbo:buildingType, dbo:type, dbp:buildingType, dbp:type, dbp:architecturalType, dbp:architectureType, dbp:category,
hypernym (89)	http://purl.org/linguistics/gold/hypernym
architectural style (49)	dbo:architecturalStyle, dbp:architecturalStyle, dbp:architectureStyle, dbp:style, dbp:architecture
architect (61)	dbo:architect, dbo:builder, dbp:architect, dbp:author, dbp:builder, dbp:engineer, dbp:foundedBy, dbp:renArchitect, dbp:renOthDesigners
architecture firm (2)	dbp:architectureFirm
sculptor (1)	dbo:sculptor, dbp:sculptor
other name (17)	dbo:synonym, dbp:alternateName, dbp:alternateNames, dbp:designation1Offname, dbp:designation2Offname, dbp:nativeName, dbp:otherName
former name (2)	dbo:formerName
completion date (45)	dbo:buildingEndDate, dbp:built, dbp:completionDate, dbp:completedDate, dbp:dateComplete, dbp:dateConstructionEnds, dbp:established, dbp:founded, dbp:used, dbp:yearCompleted
construction start date (37)	dbo:buildingStartDate, dbo:yearOfConstruction, dbp:beginningDate, dbp:brokeGround, dbp:date, dbp:dateConstructionBegins, dbp:groundbreaking, dbp:startDate, dbp:yearsBuilt
demolition date (1)	dbp:demolished
extension date (1)	dbp:extension
restoration date (4)	dbp:restored, dbp:dateRenovated, dbp:renovationDate
UNESCO designation date (12)	dbp:year, dbp:whsYear, dbp:designation1Date
location (86)	dbo:location, dbp:location
country (27)	dbo:country, dbp:country, dbp:locationCountry, dbp:state, dbp:stateParty
culture (2)	dbp:cultures
bell count (4)	dbp:bells
dome count (2)	dbp:domeQuantity
elevator count (16)	dbp:elevatorCount
floor count (18)	dbo:floorCount
minaret count (3)	dbp:minaretQuantity
room count (2)	dbp:roomCount, dbp:rooms
spire count (6)	dbp:spireQuantity
step count (1)	dbp:stepCount
suite count (1)	dbp:suites
tower count (3)	dbp:towerQuantity
cost (17)	dbo:cost, dbp:cost, dbp:constructionCost
elevation (1)	dbo:elevation
floor area (11)	dbo:floorArea
height (9)	dbo:height, dbp:height
seating capacity (10)	dbp:capacity, dbp:seatingCapacity, dbp:garrison
building confused with (2)	owl:differentFrom
facade direction (3)	dbp:facadeDirection
highest building start date (7)	dbp:highestStart
highest building end date (7)	dbp:highestEnd
highest building region (1)	dbp:highestRegion
construction material (8)	dbo:material, dbp:material, dbp:materials
structural system (2)	dbp:structuralSystem

Table 7: List of retrieved features from DBpedia, and number of occurrences in the development set (in grey, properties already covered by the base generator)

with the 98 properties from DBpedia. In parentheses, the number of times each property had one or more value(s) for a building. There can be two reasons why there are more than one value for a feature: (i) one property is given more than one value, or/and (ii) multiple properties have one value.

Alley	alcove	amphitheater
amusement_park	apartment_building_outdoor	aqueduct
Arch	archaeological_excavation	atrium_public
Attic	auditorium	balcony_exterior
balcony_interior	ball	Bar
barn	barrier–curb	barrier–fence
barrier–guard-rail	barrier–wall	bathroom
bazaar_indoor	bazaar_outdoor	beach_house
bedroom	berth	bow_window_indoor
box	building_facade	cafeteria
campus	car	case
castle	catacomb	cemetery
chalet	chest of drawers	children_room
church_indoor	church_outdoor	classroom
cloister	computerroom	concert_hall
cornice	corridor	cottage
courthouse	courtyard	crosswalk
deco	department_store	dining_hall
dining_room	discotheque	doorway_outdoor
downtown	driveway	eiffel-tower
elevator	embassy	engine_room
entrance_hall	escalator_indoor	excavation
fabric_store	façade/wall	farm
fire_escape	fire_station	fireplace
flat–bike-lane	flat–crosswalk-plain	flat–curb-cut
flat–parking	flat–pedestrian-area	flat–rail-track
flat–road	flat–sidewalk	food_court
formal_garden	gameroom	garage_indoor
garage_outdoor	gazebo_exterior	general_store_indoor
general_store_outdoor	golden-gate-bridge	greenhouse_indoor
greenhouse_outdoor	gymnasium_indoor	home_office
home_theater	hospital	hotel_outdoor
hotel_room	house	hunting_lodge_outdoor
igloo	indoor	industrial_area
inn_outdoor	kasbah	kindergarden_classroom
kitchen	library_indoor	library_outdoor
living_room	lighthouse	mansion
lobby	mausoleum	manufactured_home
market_indoor	market_outdoor	meeting_room
mirror	mosque_outdoor	motel
movie_theater_indoor	oast_house	museum_indoor
museum_outdoor	nursery	office
office_building	palace	pagoda
painting	pantry	park
person	parking_lot	patio
pavilion	pier	playground
pool/inside	pub_indoor	pyramid
restaurant	restaurant_kitchen	restaurant_patio
River	rock_arch	rope_bridge
Ruin	Schoolhouse	sculpture
Shed	Shopfront	shopping_mall_indoor
sill	ski_resort	Skyscraper
smokestack	Stable	stained-glass
staircase	structure–bridge	structure–building
structure–tunnel	swimming_pool_indoor	swimming_pool_outdoor
synagogue_outdoor	temple_asia	throne_room
Tower	tower-pisa	train_station_platform
tree_house	Village	water_tower
waterfall	wind_farm	windmill
window	youth_hostel	zen_garden

Table 8: List of classes supported by the object detection module

A.3 Details on the visuals analysis

List of extracted visual features. Table 8 shows the list of all features extracted from images.

Training of models. The training settings of each component's model involve a batch size of value 2, learning rate of 0.0001, momentum value equal to 0.9, weight decay of 0.0005 and weights initialisation as described on the above section. For the architectural style recognition task (see the confusion matrix in Table 7), the experiments involved Stochastic Gradient Descent and Adam as optimisers. Different epochs, batch size and learning rates were tested. Finally a VGG19 model was trained for 130 epochs. The training includes 3-fold cross validation, and SGD optimiser of learning rate equal to 0.001. All trainings and evaluations were conducted on a 1080Ti GPU.

A.4 Sample output texts

Tables 9, 10, 11 and 12 show sample texts for a few buildings; the parts of the text that come from the textual and visual analysis are shown **in bold**, and incorrect content is shown in red.

Wikipedia (human)
The Burj Khalifa, known as the Burj Dubai prior to its inauguration in 2010, is a skyscraper in Dubai, United Arab Emirates. With a total height of 829.8 m (2,722 ft, just over half a mile) and a roof height (excluding antenna, but including a 244 m spire) of 828 m (2,717 ft), the Burj Khalifa has been the tallest structure and building in the world since its topping out in 2009 (preceded by Taipei 101).
DBpedia
Burj Khalifa, which Adrian Smith (architect) designed, is a Mixed-use in Dubai. It costed 1,500,000,000$. It has 2 floors and 57 elevators and a floor area of 309,473m2. It was the highest building in the world. Burj Khalifa, the architectural style of which is Neo-futurism, was built between 6 January 2004 and 31 December 2009. It was built of glass, steel, aluminium, reinforced concrete. It was formerly called Burj Dubai.
Fused
Burj Khalifa, which Adrian Smith (architect) designed, is **a skyscraper in a downtown environment** in Dubai. It costed 1,500,000,000$. It has 2 floors and 57 elevators and a floor area of 309,473m2. **It has similarities with a tower and a train station.** It was the highest building in the world. Burj Khalifa, the architectural style of which is **Deconstructuvism**, was built between 6 January 2004 and 31 December 2009. It was built of glass, steel, aluminium, reinforced concrete. It was formerly called Burj Dubai.

Table 9: Burj Khalifa (eval set)

Wikipedia (human)
Christ the Redeemer is an Art Deco statue of Jesus Christ in Rio de Janeiro, Brazil, created by French sculptor Paul Landowski and built by Brazilian engineer Heitor da Silva Costa, in collaboration with French engineer Albert Caquot. Romanian sculptor Gheorghe Leonida fashioned the face. Constructed between 1922 and 1931, the statue is 30 metres (98 ft) high, excluding its 8-metre (26 ft) pedestal. The arms stretch 28 metres (92 ft) wide.
DBpedia
Christ the Redeemer (statue), which was built of Soapstone, is a Statue in Brazil.
Fused
Christ the Redeemer (statue), which was built of Soapstone, is **a statue in a zen garden environment** in Brazil. **Its architectural style is Hellinistic. Christ the Redeemer (statue) has similarities with a windmill and a beach house. There is an elevator shaft in it.**

Table 10: Christ the Redeemer (eval set)

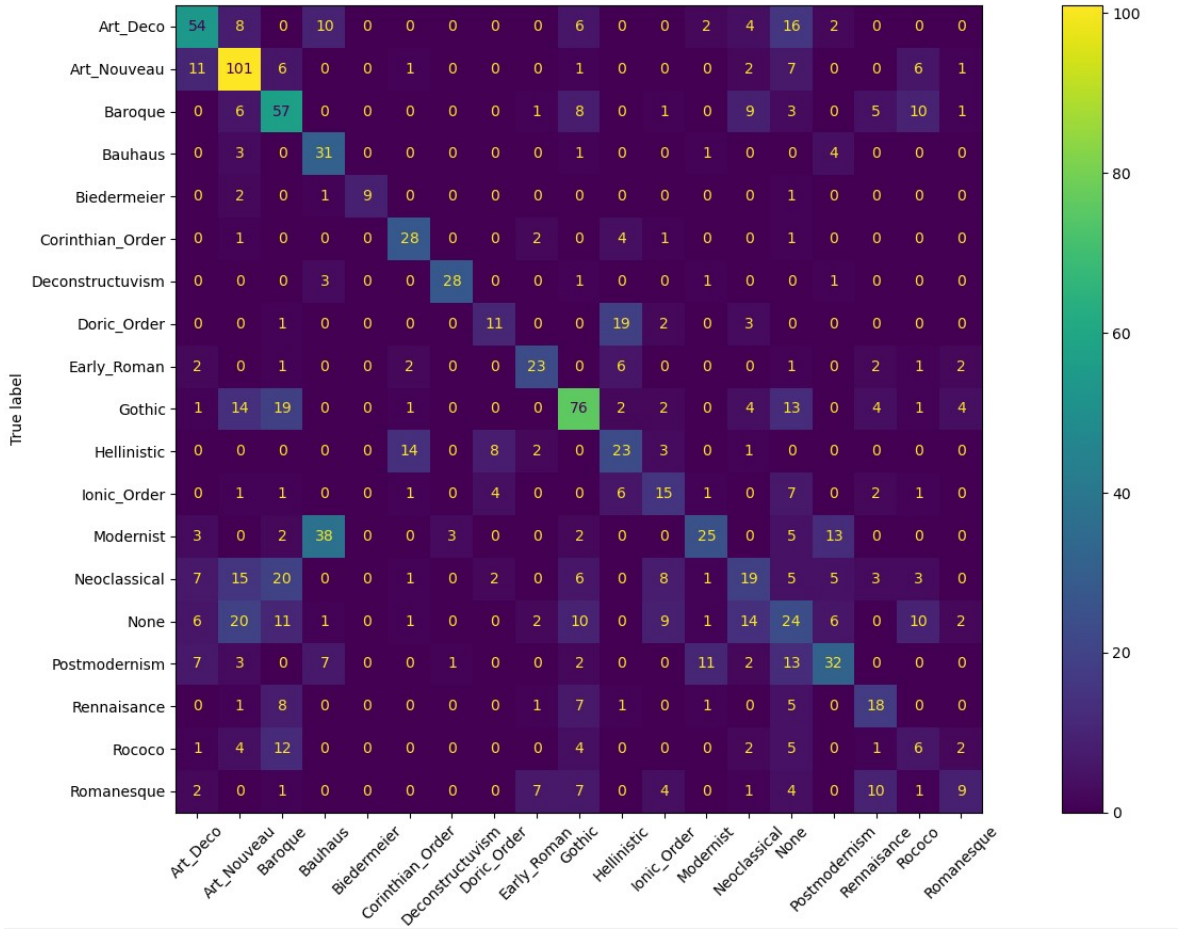

Figure 7: Confusion matrix of the produced architectural styles model

Wikipedia (human)
The Dancing House, or Fred and Ginger, is the nickname given to the Nationale-Nederlanden building on the Rašínovo nábřeží (Rašín Embankment) in Prague, Czech Republic. It was designed by the Croatian-Czech architect Vlado Milunić in cooperation with Canadian-American architect Frank Gehry on a vacant riverfront plot. The building was designed in 1992 and was completed four years later in 1996.
DBpedia
Dancing House, which Frank Gehry designed, is a Nickname in CzechRepublic (Prague). It was built between 1992 and 1996. It was formerly called Fred and Ginger.
Fused
Dancing House, which Frank Gehry designed, is a nickname in CzechRepublic (Prague). **It has similarities with an embassy, a palace and a parking garage. A fire escape can be seen on its facade.** Dancing House, the architectural style of which is Art Nouveau, was built between 1992 and 1996. It was formerly called Fred and Ginger.

Table 11: Dancing house (eval set)

Wikipedia (human)
The Sydney Opera House is a multi-venue performing arts centre at Sydney Harbour in Sydney, New South Wales, Australia. It is one of the 20th century's most famous and distinctive buildings.
DBpedia
Sydney Opera House, which Jørn Utzon designed, is a Performing arts center in Sydney. Sydney Opera House, the architectural style of which is Expressionist architecture, was built between 1 March 1959 and 1973. Its structure is made of Concrete frame & precast concrete ribbed roof.
Fused
Sydney Opera House, which Jørn Utzon designed, is a **centre** in Sydney. Its structure is made of Concrete frame & precast concrete ribbed roof. **An element of the structure is like a bridge. Sydney Opera House has similarities with a beach house**, **an amusement park and a museum**. Sydney Opera House, the architectural style of which is **Deconstructuvism**, was built between 1 March 1959 and 1973.

Table 12: Sydney Opera house (eval set)