Cultural Scene Detection Using Reverse Louvain Optimization

Article *in* Science of Computer Programming · December 2014

DOI: 10.1016/j.scico.2014.01.006

4 authors, including:

Neil Lachapelle

1 PUBLICATION **3** CITATIONS

SEE PROFILE

Cultural Scene Detection Using Reverse Louvain Optimization

Mohammad Hamdaqa[a], Ladan Tahvildari[a], Neil LaChapelle[b], Brian Campbell[b]

[a] *Software Technologies Applied Research (STAR) Group, University of Waterloo, Waterloo, Ontario, Canada*
[b] *Sceneverse Inc., Kitchener, Ontario, Canada*

Abstract

This paper proposes a novel approach for discovering cultural scenes in social network data. "Cultural scenes" are aggregations of people with overlapping interests, whose loosely interacting activities form virtuous cycles that amplify cultural output (e.g., New York art scene, Silicon Valley startup scene, Seattle indie music scene). They are defined by time, place, topics, people and values. The positive socioeconomic impact of scenes draws public and private sector support to them. They could also become the focus for new digital services that fit their dynamics; but their loose, multidimensional nature makes it hard to determine their boundaries and community structure using standard social network analysis procedures. In this paper, we: (1) propose an ontology for representing cultural scenes, (2) map a dataset to the ontology, and (3) compare two methods for detecting scenes in the dataset. Method One takes a hard clustering approach. We derive three weighted, undirected graphs from three similarity analyses; linking people by topics, topics by people, and places by people. We partition each graph using Louvain optimization, overlap them, and let their inner join represent core scene elements. Method Two introduces a novel soft clustering approach. We create a "scene graph": a single, unweighted, directed graph including all three node classes (people, place, topic). We devise a new way to apply Louvain optimization to such a graph, and use filtering and fan-in/out analysis to identify the core. Both methods detect core clusters with precision, but the first method misses some peripherals. Method Two evinces better recall, advancing our knowledge about how to represent and analyse scenes. We use Louvain optimization recursively and in reverse to successfully find small clusters.

Keywords:
Scene Ontology, Scene Graph, Adventitious Network, Social Analytics, Community Detection, Cultural Web

1. Introduction

Cultural scenes[1] [1, 2, 3, 4] emerge whenever a critical mass of people interacts within some shared context (place and time) with overlapping interests on shared topics [5]. Examples include the New York art scene, the Silicon Valley startup scene, the Paris fashion scene, and myriad smaller and less sharply delineated local scenes all over the world.

Email addresses: mhamdaqa@uwaterloo.ca (Mohammad Hamdaqa), ladan.tahvildari@uwaterloo.ca (Ladan Tahvildari), neil@sceneverse.com (Neil LaChapelle), brian@sceneverse.com (Brian Campbell)
URL: http://stargroup.uwaterloo.ca (Mohammad Hamdaqa), http://www.sceneverse.com (Brian Campbell)

[1] Related social phenomena include: subcultures (Hebdige, 1979), neo-tribes (Maffesoli, 1996; Cova, 1997; Kozinets, 2001), and genres (Lena & Peterson, 2008).

1.1. Problem: The Challenge of Scene Analysis

People on a scene do not typically all know each other. Connections both within and between clusters can be weaker than in a friends-based network, as well as less direct. The only connection between two people may be two connected interests, participation in similar events, or patronage of a particular business that is a known scene place. This partial mutual anonymity is important for giving scenes the diffuse and pervasive character that lets them serve as a soft frame of reference for their diverse, differentially committed participants [6, 7].

The diffuse nature of scenes does not prevent them from being powerful drivers of economic and cultural value creation. The indie music contributed approximately $379.4 million to the Canadian national economy in 2011, and roughly half of that value was generated by smaller players operating at the local scene level [8]. Chicago assessed the impact of its own lo-

cal indie music scene by determining how much of the $80 million spent on live music tickets in 2004 went to large pop acts listed in the Billboard 100 versus niche and specialized artists listed in the Village Voice Pazz and Jop Critics Poll. Again, the split was close to 50% [9]. When Seattle assessed the economic impact of their own musical scene in 2004, they of course had to refer to one particular grassroots/indie scene repeatedly: the grunge scene, made world famous by bands like Nirvana, Pearl Jam, Soundgarden and Alice in Chains [10]. Small local scenes frequently blow up to become global phenomena, they can utterly transform local economies in the process.

The positive socioeconomic impact of scenes is strengthened, not weakened by the indirectness of scene networks. In this regard, scenes can be described as *adventitious networks*. The property of *adventitiousness* in this context means that many links are accidental and indirect, but fortuitous[2] [11]; producing positive feedback cycles of positive externalities[3] [7], like Adam Smith's invisible hand. People accidentally and unintentionally support and inspire people they will never meet to join the creative community and produce what it values, by virtue of these adventitious links [13].

In order to preserve the adventitious property of scene networks, *representations of the community structure and boundaries of scenes need to be inclusive*. High recall and larger cluster sizes are more desirable than narrower representations, given equal or near-equal precision. This is because *the scene periphery feeds the core*. The participation of less central people adventitiously supports the creativity of central people, so losing sight of scene participants seriously compromises a scene representation. However, achieving the necessary degree of recall with precision is difficult, because: (1) scenes are dynamic and evolve over time, (2) scenes are multifaceted, involving multiple interacting dimensions (topics, people, locations, times), and (3) scene interests can be hidden or implicit; in cold stars (i.e., not explicitly ranked or rated [14]), or sparse data.

1.2. Research Goal and Methods

The goal of this paper is to contrast two approaches to discovering scenes in cultural data with a special interest in assessing the power and precision of each method for retrieving scene people. The data for our comparison came from the location-based social networking service Meetup[4]. This online service helps people coordinate real-time, face-to-face gatherings ("meetups") on topics of shared interest, and so serves as an acceptable proxy or indicator of scene activity. Our dataset included all meetups within 25 miles of Waterloo, Ontario, Canada. We devised a scene ontology that we used for organizing and processing the dataset, and subjected it to the following procedures.

In Method One, we generated three weighted, undirected graphs based on similarity analysis. One graph represented scene people according to similar interests. The second graphed scene topic similarity based on people interested in those topics. The third graphed similar locations based on people at those locations. Each graph was then partitioned using Louvain modularity optimization [15] to reveal community structure, and then the three graphs were overlapped to reveal scenes. The inner join of the graphs was taken to represent central scene elements. This was a relatively hard clustering approach.

In Method Two, we devised a *scene graph*; an unweighted, directed graph which combines people, place and topic nodes in a single graph; with people as the source nodes and places and topics as target nodes. We also devised a way of applying the Louvain method to this graph, treating it as an undirected graph for modularization, then applying record reconciliation to restore node facet information to the partitioned graph for subsequent facet filtering. We could thus determine the community structure of scenes in the data; and identify their constituent topics and people. Then we exploited the still-available directional information in the graph using fan-in/fan-out analysis to determine centrality. This was a relatively soft clustering approach.

1.3. Findings and Limitations

A key finding of our research is that the limitations of Louvain optimization for identifying small clusters in large datasets can be overcome when the source domain is hierarchical. Large scenes generally contain sub-scenes. We therefore exploited the hierarchical operation of the Louvain method by applying it recursively "in reverse" to find the sub-scenes. That is, we first applied it to the whole network to discover the main scenes, and then to those resulting scenes to reveal subscenes. This enabled the successful detection of community structure at all scales.

[2] Adventitiousness produces serendipity, so adventitious networks would subsume and generate "serendipitous networks", defined as new connections between people who find themselves in the same immediate situation

[3] Shank (1994) [12] defines a scene as a runaway creative system: "an overproductive signifying community (in which) far more semiotic information is produced than can be rationally parsed"

[4] http://www.meetup.com

Results from the two different methods were evaluated with ground truth data, Jaccard similarity and our own metric of "scene theme" similarity. Both graphing techniques correctly identified the scene cores, but community size was larger with scene graph analysis (Method Two) than it was when similarity graphs were overlapped (Method One). This suggests that the softer scene graph analysis technique performed better at delineating the actual scene boundaries in the available data, and better preserved network adventitiousness.

Our scene graph is amenable to many more social network analysis techniques, and the extraction of more insights into scene structure and dynamics. However, such work lies beyond the scope of this paper.

1.4. Significance of the Study

Several original contributions to the literature emerge from our research.

(i) We introduce the unique features of cultural scenes, including the property of adventitiousness, and propose them as new objects for social network analysis.

(ii) We formalize our current conceptualization of scene elements in a semantic ontology.

(iii) We take Louvain optimization; a clustering and partitioning technique usually used on bipartite, undirected graphs; and apply it to a directed k-partite graph. This enables Louvain partitioning of a multidimensional directed network.

(iv) We exploit the hierarchical operation of Louvain optimization to circumvent its difficulty in detecting small clusters in large networks; applying it to the global network first, then recursively to emergent sub-graphs.

(v) We introduce a soft clustering strategy involving a novel "scene graph" and techniques for analyzing it; which together provide better scene recall than a harder clustering approach, with equal precision.

(vi) We create a "scene theme similarity" metric, in a manner which may turn out to be generalizable to other k-partite graphs. No use of Louvain optimization that we are aware of applies it in the manner described in our paper.

The reminder of this paper is organized as follows: Section 2 describes related work. Section 3 gives an overview of Sceneverse platform. Section 4 presents the scene ontology. Section 5 gives an overview of the proposed approach and motivates its main ideas. Section 6 presents our experimental design, the obtained results, and a discussion on the results, respectively. Finally, conclusions and directions for future work are presented in Section 7.

2. Related Work and Research Context

This section position our work within the existing related work and defines its context.

2.1. Related Work

This article presents an empirical study of scene discovery in online socio-cultural network data. This section puts our work in context within the substantial literature targeting similar problems.

2.1.1. Community Detection in Networks

A scene is a type of social community (i.e., people community) that shares topics of interest in designated locations during a period of time. In network and graph theory, a community is defined as a group of nodes that are densely connected to one another, but have relatively weak connections with other parts in the network [16]. Partitioning of nodes into groups and sub-groups is crucial to understand the meaning and behaviour of the network. Studying grouping patterns to detect communities has been the focus of many research studies for long time (e.g., Stuart Rice had manually clustered data to study political groups in the 1920s [17]). Communities have been studied in almost all domains (e.g., social sciences [18, 19, 20], bibliometrics [21], anthropology [22], telecommunication [15], biology (i.e., human brain connectome [23])). For comprehensive studies on the literature of community detection in networks the reader can refer to Porter et al [17], or Fortunato et al [16] respectively.

Recently, there has been increasing interest in applying community detection techniques to discover virtual communities in online social networks and the cultural web [24, 25, 26, 27, 28]. Several techniques and algorithms have been devised to automatically detect communities in networks. These techniques can be divided into two groups based on the type of the methods used to find the linkage between the network nodes; namely, community detection based statistical correlation and similarity analysis (e.g., hierarchical clustering, k-means), and community detection based graph partitioning [29] (e.g., Girvan Newman algorithm [20], network modularity [15], surprise maximization [30], k-clique percolation [31]).

Michelle Girvan and Mark Newman proposed using graph clustering for community detection in 2002 [20]. Since then, the field of community detection based

graph clustering techniques (i.e., Modularity Analysis) become vibrant. This is because, graph clustering and community detection share the same goal; to find clusters of vertices (i.e., modules) on a graph that are strongly connected to each other than to the rest of the network. However, while graph clustering techniques usually require the number of clusters as input to the algorithm, in community detection techniques the number of communities is one of the desired outcomes.

Several studies have compared the aforementioned community detection techniques with regard to modularity, performance, memory requirements, scalability and other measures. The work by Papadopoulos et al [26] is an example of an up-to-date comprehensive comparison between these techniques within the context of social media networks.

2.1.2. Social Graph Creation

Community detection is domain specific. This is because different domains expose different network structures. The underlying topology of a network is essential in order to utilize their data in applications. Social media networks comprise of multiple types of vertices and edges depending on the focal object the network is created around (e.g., people in Facebook , pictures in Flickr). The way the network is designed can significantly affect the community detection techniques that can successfully works on it. For example, a scene graph consists of three types of vertices. However, the majority of community detection techniques explained earlier work only with simple graphs (i.e., undirected with one or two types of vertices), and do not work with k-partite or hyper-graphs [26]. For example, the Louvain modularity optimization method, used in this paper, was originally designed to work with undirected graphs. Arenas et al [32] proposed a graph transformation method to enable applying modularity optimization on directed graphs. Similarly, Barber extended applying modularity optimization to bipartite graphs [33]. Conversely, our approach does not require applying any graph transformation to scene graphs; instead it utilizes Louvain modularity unawareness of the vertices-types or the edges-directionality to perform pure modularity clustering. Then, fan-in and fan-out analysis are applied to the graph to highlight the hidden information within each community.

To summarize, modularity and graph clustering techniques are usually applied to social networks after reducing the network into a simple form that consists of a maximum of two types of vertices. The price paid for this graph reduction is obviously a loss in information. Consequently, these approaches fail to detect communities in social networks that cohere in multifaceted ways (i.e., scenes).

2.1.3. Community Detection Applications

There have been several recent works that attempt to derive meaningful clustering using modularity techniques and graph partitioning. Most of these works are single facet graph clustering (i.e., topic, people, location, picture, etc.). In an attempt to create a folksonomy [34], Begelman et al. [35] first designed an inter-tag correlation graph, in which tags that describe the same resources are correlated. Then they partitioned this graph using spectral bisection and modularity function. Sharing the same goal of clustering similar tags to create folksonomies, Simpson [36], and Papadopoulos et al. [37] applied different variations of graph partitioning techniques on tag graphs to divide the graph into tags modules.

Fatemi et al [38] constructed a social network graph for the internet movies database (IMDb) based on the shared reviewers for these movies. Fatemi then used four community detection algorithms to discover the underlying community structure of these movies. The study of IMDb is interesting because users review diverse topics that are interesting to them, hence communities of movies linked by their reviewers can reveal diverse interests regardless of their genre tags.

In a study on friendship relationship between social network users, Mislove et al. [39] have crawled user public profiles from different social media providers (i.e., YouTube, Orkut, Flickr and LiveJournal). The authors then studied the structure of the created network by applying graph partitioning methods. Traud et al performed a similar study on data collected from Facebook, in which they examined the roles of universities in structuring the social networks of students [40]. In the same vein, several studies have been conducted to reveal user segmentation based on various similarity factors. This has extensively been explored lately in the field of content filtering and smart recommendation systems. For example, Tsatsou et al. [41] integrate the results of tag community detection in a personalized ad recommendation system. Moreover, Pham et al. [42] grouped users into clusters to identify the neighborhood and hence derive better recommendations than traditional content filtering algorithms. García-Crespo et al. used natural language processing and semantic categorizations of opinions to analyse customer emotional implications to assist in deriving marketing strategies and product development plans [13].

Perhaps the most relevant work to the approach proposed in this paper is the work done by Zhao et al. [43],

which is a multifaceted stepwise clustering approach. The goal of their work is to find events in social text streams (e.g., blogs). Zhao et al. defined an event as the information flow between a group of social actors on a specific topic over a certain time period [43]. There are two main differences between the scene as a concept and Zhao's definition of events. First a scene is a higher-level concept, in which an event serves as the temporal snapshot that captures a social occasion during a specific time interval on the scene timeline, and has a particular title that belongs to the scene topics theme (as explained in the scene ontology section). The second difference is that the scene concept also has a spatial dimension.

Zhao et al. combined the three event dimensions; namely, the temporal, the social and the textual content of blog streams to discover events. Their approach consists of three phases. In the first phase they transform the social text streams into a graph, then partition the graph using the N-cut graph partitioning method [44] into topics so that each blog belongs to one topic. In the second phase, a social graph is constructed. The topic-based social graph is then partitioned into a sequence of graphs based on the intensity along the temporal dimension. Finally, in the third phase, the social-temporal-topical graphs are finally divided into finer grained events by applying the N-cut graph partitioning method for the second time. All the previous studies deal with undirected weighted graphs, in contrast to the people, topics and location graphs partitioning in our approach. Only the work by Zhao et al. [43] deals with multifaceted graphs. However, none of these works applied partitioning on a directed multifaceted graph in single step, and used the power of fan-in and fan-out analysis to further identify the facets types in each cluster as we did in our approach to scene graph partitioning.

2.2. Research Context: The Sceneverse Platform

This study is part of a collaborative research program supporting the development of the Sceneverse platform.

2.2.1. Sceneverse Mission

Sceneverse, a portmanteau of "scene" and "universe", aims to support all scenes on a platform optimized for representing scene dynamics and facilitating scene transactions. Though scenes are natural contexts for economic activity [45, 46], scene commerce can be contentious. For it to be successful, it must respect the complex interplay of values, politics, ideologies and attitudes that structure scenes [47, 48, 49, 50]. For this reason, accurate representations of scenes and scene values are crucial for providing them with digital services.

On the Sceneverse platform, scene data will be derived from two sources: user-contributed content, and the behaviour of people using Sceneverse enabled web and mobile applications. The provision of value-adding services will depend on the faithful representation and analysis of scenes in this data

2.2.2. Front-end Applications

Figure 1 presents an overview of the Sceneverse[5] platform. It provides complementary services at two different levels, front-end and back-end. Front-end services consist of web and mobile applications that serve participants' needs from the scene center to its margins.

There are different levels of participation in scenes. Active scenes have a small, dense core of avid participants; as well as "near-satellite" members who participate fairly frequently, and many peripheral "far-satellite" members who participate infrequently.

For example, the inner core of an art scene would consist of full-time professional artists, the dealers that represent them, the galleries that display their works, the art critics that review it, and the primary patrons who purchase works. It would also include avid amateurs who spend equal amounts of time in these activities, and attend many of the same events, but who do so largely on a voluntary, non-cash basis. Near-satellite members would be the friends and contacts of this inner core who participate as spectators or dabblers in art-scene-related activities on a consistent basis (their default choice is to participate unless they are busy), but whose main occupations and preoccupations lie elsewhere.

Far-satellite members many not have friends or relatives in the scene core, but maintain an interest in art. They attend exhibitions, buy books and prints, and take classes on an opportunistic and occasional basis, rarely committing to more than one such activity, and only doing so once in awhile.

Sceneverse-enabled applications help the inner-core find better ways to produce and consume the "cultural capital" of the scene. Satellite members enjoy high-recall services that give them easier access to the scene's core, enhancing their scene experience. Peripheral participants enjoy high-precision services that help them quickly and efficiently enjoy select scene activities as they fit their moods and inclinations.

Current Sceneverse frontend applications target avid/core and near-orbiting scene participants. One example Sceneverse currently offers is an event planning and ticket sales service for people booking concerts on

[5] www.sceneverse.com

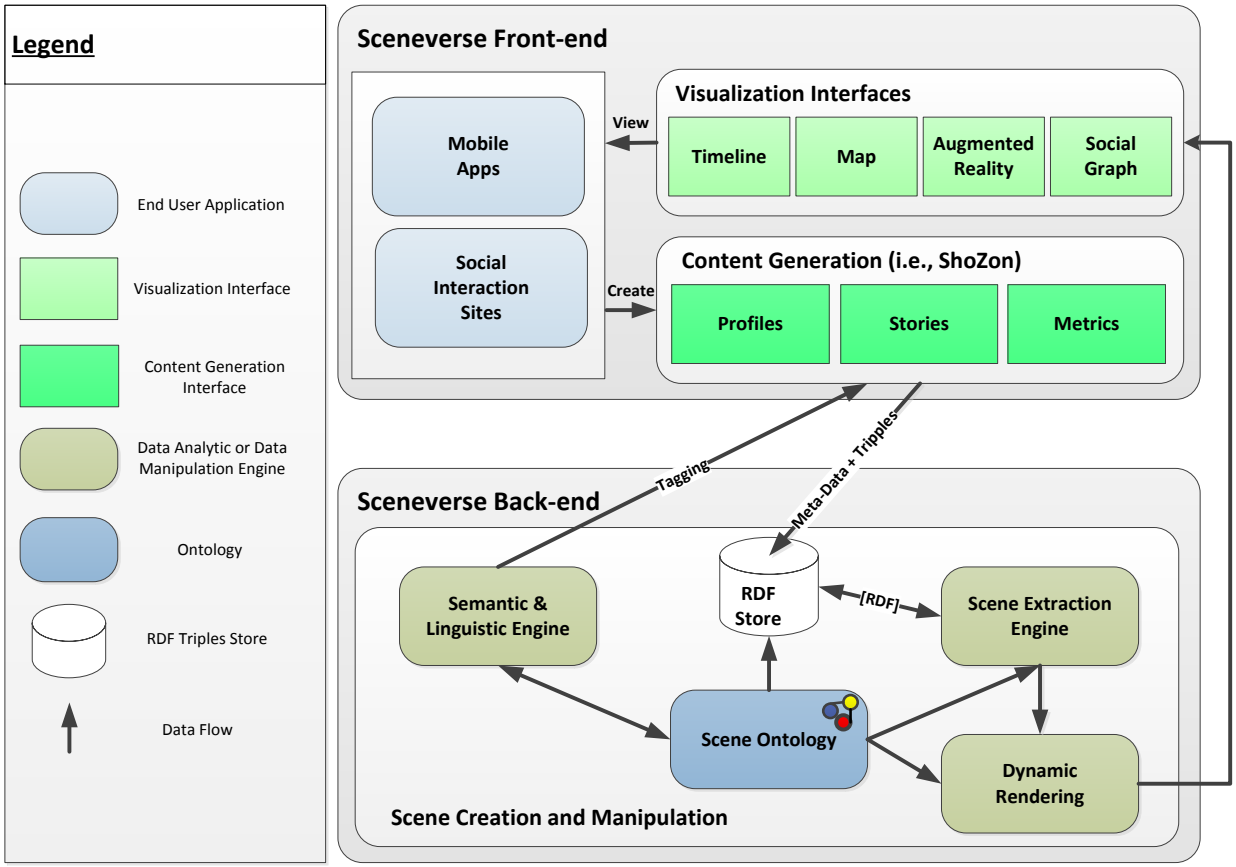


Figure 1: Sceneverse Platform Architecture

the indie music scene. Sceneverse is also producing a storytelling application that engages scene participants in social, mobile, augmented reality content creation and curation. The application lets participants compose stories describing personal scene experiences, chronicle the broader history of their scene, discover other peoples' stories, and indicate their sentiments towards stories. Stories include those based on Sceneverse Events. The user-contributed content and behaviour provides essential information to create what we call a user universe. A user universe can be seen as an ego network of a user's topics of interests, locations, and behaviour monitored in both time and space.

2.2.3. Back-end Processesing

On the back-end, the Sceneverse platform aims to offer a cross-service, cross-device, pervasive frame of reference for all the digitally mediated activities that might support a local scene.

The back-end architecture under consideration in this paper consists of three main components; the scene ontology, a semantic and linguistic engine, and the scene extraction engine. Both of the engines depend on the scene ontology, which provides the vocabulary for building semantic queries, rendering scene content, and reasoning about new and existing scenes. For example, the semantic and linguistic engine provides natural language understanding and semantic web links for processing and annotating user stories with context-appropriate tags. It also supports dynamic rendering of content based on the scene ontology. The extraction engine uses pre-existing web data as well as data gathered from frontend services to discover scenes. All the algorithms needed to identify and reason about the socio-spatio-topical boundaries of scenes are either implemented in the semantic/linguistic engine or the scene extraction engine.

These back-end processes support front-end tasks and facilitate the creation and manipulation of scene representations.

3. The Scene Ontology

The scene ontology developed for this paper furnishes a set of clear concepts with well-defined interrelationships for representing cultural scenes in web data.

Such a formal scene ontology is essential for (1) building semantic queries, (2) rendering scene content, and (3) reasoning about new and existing scenes. The ontology proposed in this section is mainly used to consolidate the data collected from different resources and check inconsistencies. It will also be used in our framework for query the data.

The rationale for scene participation is the scene itself. It is its own ultimate reason for gathering/clustering. No individual scene dimension alone can furnish the reason, since adventitious connections can come through all of them. This becomes clear when you ask a core scene participant why they care about the scene. The answer is unlikely to be only one thing, or one type of thing, but their cumulative scene experience in its totality. Because the scene is both psychologically and sociologically real, it has its own distinct representation in our ontology.

3.1. High-Level Overview

The Scene is constituted by several other ontological concepts, including Topics, People, Locations and Times. In our current ontology we split the Time dimension into two categories: Scene Active Period, and Events. The two time concepts are needed to define temporal boundaries of scenes. Scenes are constituted by many events, which in total sum up to the Scene Active Period. This is shown in the UML diagram illustrating the scene ontology in Figure 3.

While the UML diagram gives a useful schematic overview, the fully formalized ontology exploits RDF and OWL to explicitly represent the scene facets and their relationship in a way that allows easy discovery, dynamic access, and simple linkage to other resources on the web.

Ontologies can be either designed from scratch or as an extension of existing ontologies. Extending existing ontologies is the recommended best practice [51]. An ontology can be extended horizontally or vertically. In horizontal extension, the original ontology is imported and used in the same way (i.e. with the same semantics) as in the domain it was imported form. In contrast, with vertical extension, an ontology is imported and then updated to support the new domain. A good core ontology should be designed to support both horizontal and vertical extension, by maintaining the right balance between domain-independent and domain-specific concepts. Getting the balance wrong can restrict further vertical extensions in the future.

Our scene ontology was designed by horizontal extension [52] through importing existing ontologies, e.g.: TimeOntology, EventOntology, FOAF, and GeoOntology. It was also designed to be generic enough to support vertical extensibility [53] to other domains.

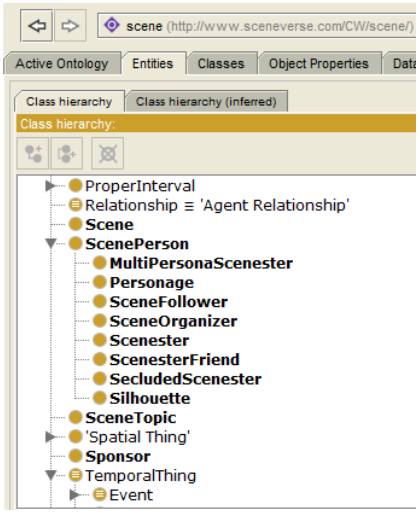


Figure 2: A Scene Ontology Excerpt

Figure 2 shows an excerpt of the Scene ontology. When transcribing an OWL ontology to RDF, every statement must be converted into triples. An RDF triple contains a subject, a predicate and an object. A set of such triples is a graph, where the subject is always a node, the predicate is always an arc and the object is always a node. The OWL scene ontology graph is fully laid out in Appendix A of this paper. Its corresponding RDF triples are given in Appendix B.

The Scene ontology has been constructed using the Protégé [54] ontology editor. Protégé utilizes various Description Logic reasoners (e.g., RACER [55], FaCT++ [56], Pellet[57]) to to perform different inferencing services (i.e., computing inferred superclasses, determining class consistency). In addition to reasoning, Protégé facilitates generating the RDFs needed to query the model using the SPARQL protocol and RDF query language [58]. In this paper, model reconciliation between the ontology and the dataset (i.e., Meetup data) was carried out by mapping the API schema elements to the ontology concept tree manually, and a script was used to populate the OWL ontology with individual elements based on the target social network site API. The design of the Sceneverse platform calls for a semantic and linguistic engine that automatically tags parsed data from users stories with concepts belonging to the scene ontology. The implementation of such an engine is out of scope for this paper.

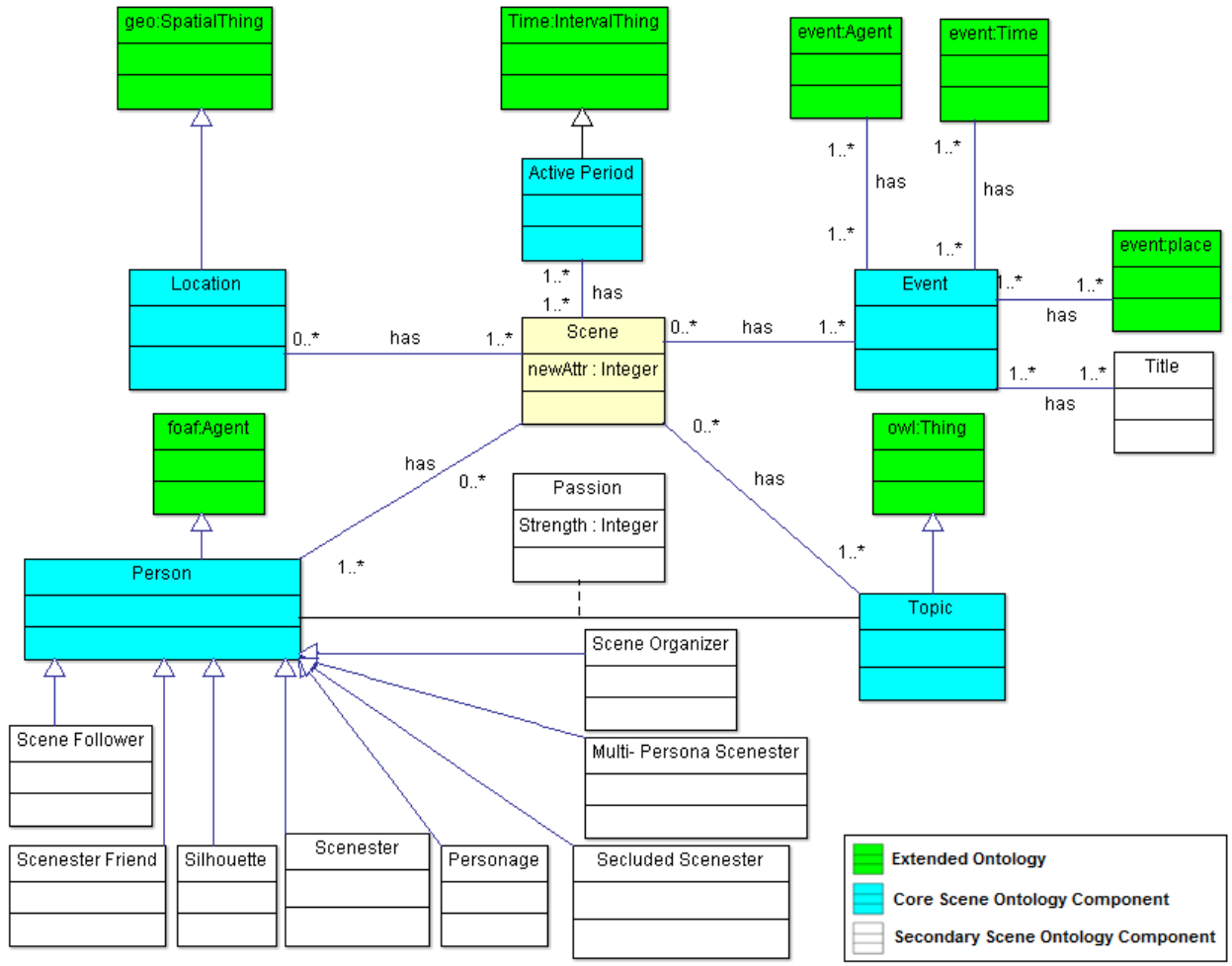


Figure 3: The Scene Ontology

3.2. Main Scene Concepts

The main scene concepts are Topic, People, Locations and Time. Each is expanded and explained in detail below.

Topic: The subject of interest. It can be anything (e.g., person, place, event, topic, thing). A scene is normally described as a list of topics (e.g. the Blues/Jazz scene).

People (social graph): Scene participants share a reason for gathering, and thus form a membership group, albeit a diffuse one. The network centrality of People derives from their contribution to the Scene and their activity level (active or passive). Types of People include:

(a) Scenester: A Person whose Scene is clearly identified.

(b) Scenester Friend: A Person who communicates and collaborates with a Scenester, but does not participate directly in that Scenester's Scenes.

(c) Multi- Persona Scenester: A Person with multiple personas. A persona is how a Person is known on a particular Scene. A single-persona Scenester is just a Scenester.

(d) Personage: Some person named or mentioned in Scene stories, chronicles or news. May or may not also be a Scenester.

(e) Silhouette: An abstraction over Scenesters, personas and Personages. Silhouettes define various categories and types of People, what they value, how they are valued and how much prestige they have in the scene. The platform's engines generate Silhouettes for marketing purposes or making recommendations etc. This allows People to be typified while protecting their privacy.

(f) Scene Organizer: A Person who sets up Scenes and grants authorizations to new Scenesters.

(g) Scene Follower: A Person who follows Scenes, but is not a Scenester or Scenester Friend. A Scene Fol-

lower is a passive presence, whose existence amplifies the Scene's reputation. However, s/he does not otherwise participate in the Scene.

(h) Secluded Scenester: People who are not part of any Scene (e.g., new members in online networks who have insufficient profile or interest data).

Location: Holds the list of locations (e.g., Uptown Waterloo, University of Waterloo) where Scene events and happenings have previously occurred. These locations are centralized around the main scene location (e.g., Waterloo)

Time: Captures the temporal aspect of the scene. Processing Time is much harder than processing the other scene dimensions. Currently, we manage scene temporality using two main concepts:

(a) Scene Active Period: The period during which People were involved in activities related to the Scene, and Events were organized. Conventions for declaring a Scene inactive are needed (e.g., if no Events have occurred in the past two years).

(b) Event: Used to capture a Scene snapshot. An Event has the following properties:

- Title: Should be aligned with Scene themes, described by the list of Scene Topics.
- Location: Should fall within the perimeter defining the core Scene Location.
- Participants: Should be Scene People. Behavioural and social data indicate when someone new should be added to the list of Scene People.
- Time: This value percolates upwards to be used to inferences that determine the Scene Active Period. The timespan between the first Event Time and the Time of the last Event defines the Scene Active Period.

Capturing the temporal data is one of the main challenges. The elements used to capture the temporal data in our ontology (i.e., Scene Active Period and Event) are discrete and hence, by using the standard methods of publishing structured data (i.e., RDF) the ontology can be updated with the correct information. Currently, the Sceneverse framework depends on batch processing to update the data, and the update function runs periodically.

3.3. Passions as Associations

Scenes are fundamentally constituted by the collective set of relationships or associations that connect People with their Topics of interest. In Figure 4 these associations are represented as Passions connecting a Person to a Topic. Figure 4 shows that people can be part of the scene or peripherals. People who are part of the scene directly affect the scene reputation through their participation or contribution to the scene. While peripherals only follow the scene; hence, they affect the scene reputation by their collective engagement not direct contributions. For example in a soccer scene, a soccer player is part of the scene because s/he directly affects the scene, while the team fans are just followers. Similarly, in the social network scene people who con-

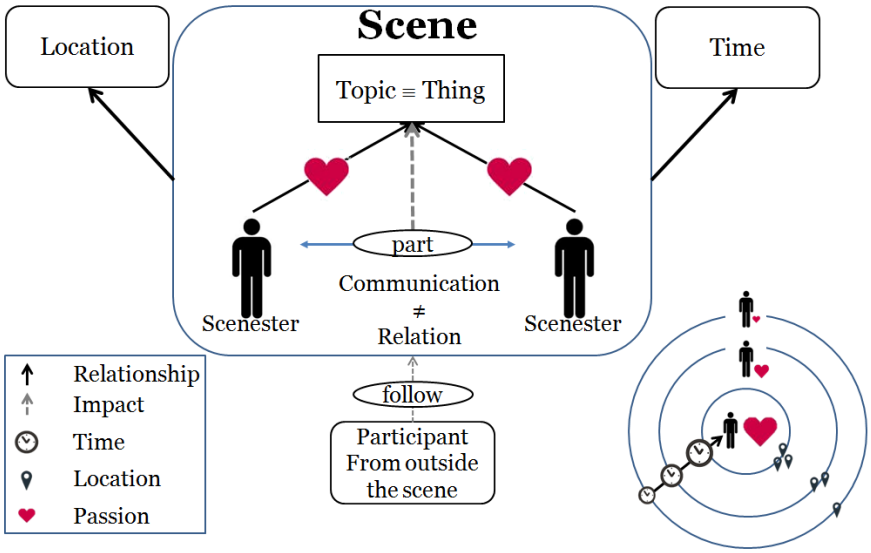


Figure 4: The Scene Conceptual Model

tribute to the topic by commenting can have direct impact on the scene and hence they are in the core of the scene; whereas, people who just like the topic or silently follow it are peripherals.

Recall that a Topic can be anything. There are topic-people who cannot be real people or Scene People (they may be fictional, like Harry Potter, or dead/historical, like Socrates). There can also be topic-people who happen to also be real people, and who may further be Scene People of some type. Similarly, there can be topic-places, topic-events and topic-periods.

"Passion for" can capture the strength of a Person's connection to any object of interest. These relationships can be explicit or implicit, with Time (real-time, not topic-time) and Location working as orthogonal factors (disjoint) that either weaken or strengthen these relationships. This is illustrated using concentric circles in the bottom right corner of Figure 4, where Passion decreases with temporal, spatial or social distance.

4. Graph Construction and Scene Identification Methods

Identifying the socio-spatio-topical boundaries of a scene is a non-trivial soft clustering problem. Clustering needs to exploit both the similarities among multiple concepts (i.e., people, topics and locations), and the relationships between these concepts, in order to identify community's boundaries. In this section, we describe how we collected and prepared our data, and enacted two methods for graphing that data and detecting scene structures and boundaries implicit in it.

4.1. Graph Construction Overview

Figure 5 illustrates the entire procedure we followed in our approach to detecting scenes, including both methods of graph generation and analysis that we evaluate in our study. Both techniques start with a data preprocessing stage. Method One (Blue) applies similarity analysis to create three undirected weighted graphs (i.e., a contingency matrix); one for topics associated by people who are interested in them, on for people associated by interest in similar topics, and one for places associated with similar people. Method Two (Red), by contrast, starts right away with the construction of a single scene graph; a simple directed graph that permits nodes of all three kinds: people, topics and locations.

Following graph construction, clustering is carried out on all of the graphs using network modularity analysis. In Method One, the three separate similarity graphs are clustered individually, and then combined by finding the intersection between the clusters. This produced

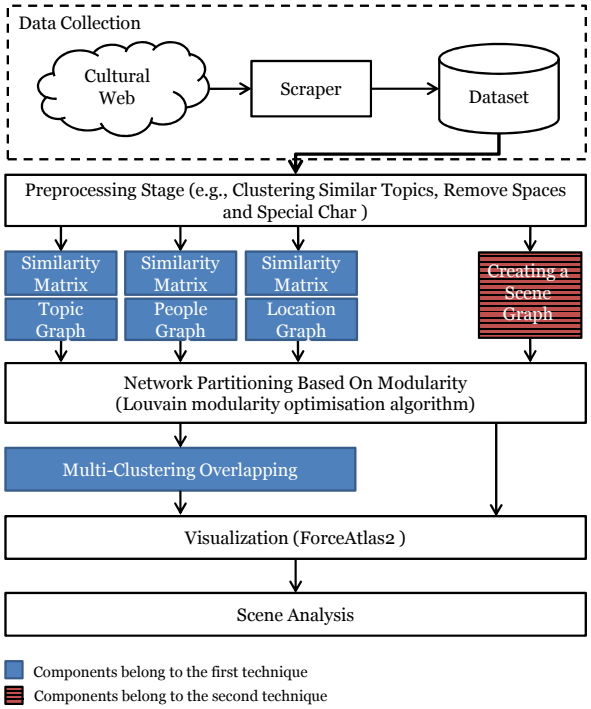


Figure 5: The Scene Discovery Approach

a single graph for comparison with the single graph already generated using Method Two.

After clustering, both resulting graphs are further analyzed and visualized using graph visualization and manipulation software (Gephi) [59]. Finally the scenes revealed by the procedures are analyzed.

Many social network analysis measures and techniques could have been used to bring out scene facets and rearrange them around graph centers. However, to keep things concise, this paper will focus mainly on scene discovery, and only mention centrality analysis techniques very briefly. A comprehensive paper will follow to illustrate in detail all the analysis techniques that can be applied to scene representations, and the socioeconomic research questions these analyses could illuminate.

4.2. Data Collection

Finding relevant cultural data for scene research is a challenging task. Most social networking sites present some, but not all of the needed data, restricting access to it for both business and privacy reasons. For our purposes, the best available data came from "Meetup.com"; an online social network that helps people organise gatherings at offline local venues to enjoy shared interests. The gatherings are called meetups, and their data

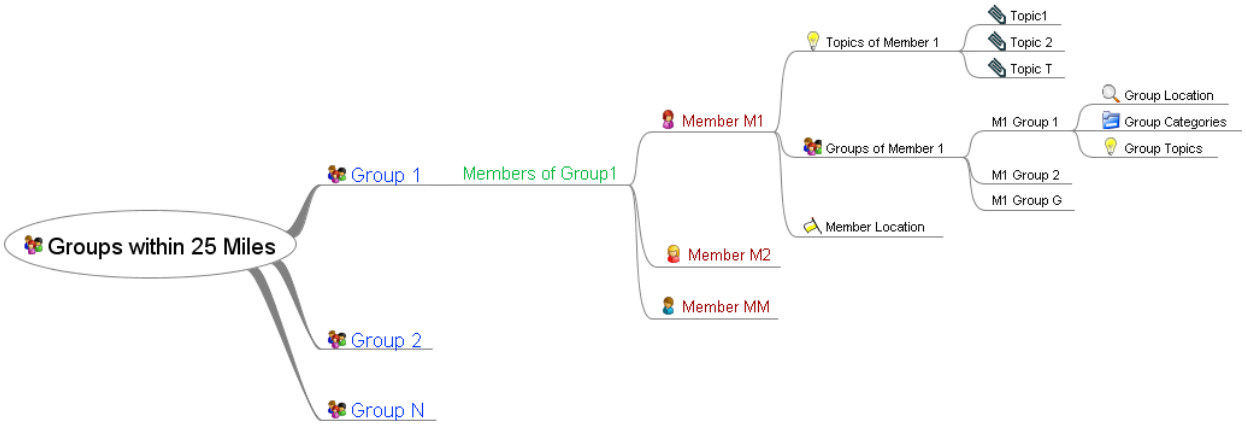


Figure 6: The Data Collection Process

points include social, topical, temporal and geographical information. Meetups are not clustered into scenes, but scenes are potentially detectable in this dataset.

Figure 6 shows the data collection process. A scraper was implemented using Meetup.com's APIs to collect the data required to build scenes. The scraper started with a specific geographical location, and returned all meetups around that location. For each Meetup group returned, the scraper then requested a membership list and the topics that describe the group. For each member in that group, the scraper then returned their topics of interest and profile information (i.e., age, residential location), as well as the groups and meetups they belong to. The time the user joined the group, as well as the user's last activity in that group were also retrieved.

4.3. Preprocessing

Several preprocessing steps were needed before the raw data was ready for clustering and analysis:

(a) **Topic Dimensionality Reduction:** Meetup lets people propose topics in their own terms at both the group purpose and personal interest level, producing a large population of topics with many similar terms that would compromise similarity-based clustering. We reduced this dimensionality by combining topics with high syntactic similarity using text-clustering techniques.

(b) **Outlier Removal:** Outliers would negatively impact scene discovery and analysis, so it was important to purge them from the dataset and correct skewed data. We used facet filtering to detect abnormalities in the data and remove outliers.

(c) **Formatting for Visualization and Analysis:** Further refinements and transformations were needed to make the data compatible with all the tools we used in our study (e.g., Gephi).

4.4. Graph Creation

As explained earlier, two methods were used to create graphs in preparation for the graph modularity partitioning step. Then the partitioning algorithm was applied in the same way to both graph types. We found that the way the graphs were created and weights assigned to their edges significantly affected the final partitioning results.

4.4.1. Method One: Similarity Matrix Graphs

In this technique, three similarity matrices were created; one for topics based on their being liked by similar people, one for people based on their liking of similar topics, and one for location based on persons who were there.

To find the topic similarity matrix, the topic-person table was first converted into a coincidence matrix (a.k.a. adjacency matrix). Each topic was represented as a vector of users who liked it. The algorithm then correlated topic similarities using cosine similarity as shown in Equation 1. Cosine similarity is defined as the cosine of the angle between two vectors (x and y) with the value being normalized between zero and one if both x and y are positive.

$$\text{CosSim}(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}} = \frac{x \cdot y}{\| x \| \| y \|} \quad (1)$$

Cosine similarity was selected in this case for two main reasons: First, cosine similarity is proven to be powerful, yet it is the simplest inner product correlation between two vectors [60]; hence, it will give better performance results. Second, our dataset does not contain any subjective values or ratings. It consists of vectors of only zeros and ones (e.g., zero represents the absence of a person on the topic list, whereas one presents their presence). This makes a higher performance correlation measure more valuable than a shift invariant one. Furthermore, calculating the cosine correlation can be easily map-reduced/parallelized. This has major implications for the applicability of our techniques on big data sets.

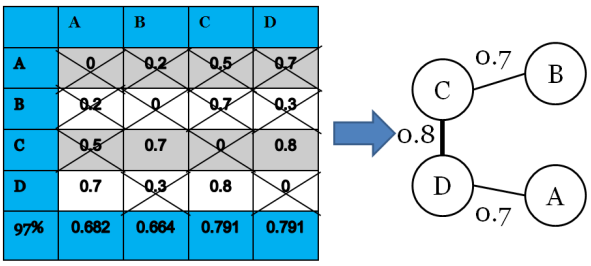


Figure 7: A Similarity Matrix and its Corresponding Graph

Cosine similarity was calculated between each pair of topics to generate a similarity matrix. The similarity matrix was then converted into a weighted undirected graph with links between nodes (edges) considered only when the weights (similarity measures) were within the upper 97th percentile. In other words, we considered the three nearest neighbors. Figure 7 shows an example of a similarity matrix and its corresponding graph.

The same steps were repeated to create the people and location similarity matrices. For example, in the case of people graph, each person was represented as a vector of the topics s/he liked. The graphs generated were then exported for partitioning and clustering based on the Louvain graph modularity algorithm.

4.4.2. Method Two: The Scene Graph

For Method Two: instead of separately clustering people, topics, and locations, we combined them on a single graph: a scene graph. This gave it the property of being bijective. We also created it as a simple (unweighted) directed graph, with people as source nodes and either topics or locations as target nodes. So a directed edge would be created from a person node to a topic node if that person expressed an interest in it. Similarly, a directed edge would be created between a person and a location if the person was involved in an activity there or a topic situated there.

We hypothesized that the precise relationships between nodes of different facet types might preserve important information, as would the directedness. Furthermore, generating a single graph would allow us to run a multifaceted similarity analysis in a single step. However, modularity maximization partitioning uses modularity strength, which depends on the graph structure to cluster the graph into communities. Hence, all nodes in the graph would be treated as equivalent regardless of type for purposes of clustering.

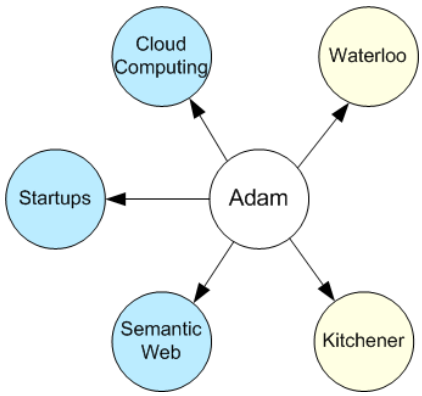


Figure 8: Scene Graph Example

Figure 8 illustrates the scene graph of one person who likes three topics, and participates in activities related to these topics in two locations (Kitchener and Waterloo). Note that a scene graph focuses on Scene Locations rather than the personal profile location, which reflects the place of residence.

The advantage of having a scene graph is twofold: (1) It enables simple one step multi-facet clustering when combined with modularity maximization graph partitioning. (2) Since scene graphs are bijective graphs, we can store more information about the relationship between scene concepts. In fact, a scene graph inherently models the core relationships of a scene (passions and places). Knowing the direction of the relationships, techniques like fan-in and fan-out analysis can be applied. This enriches knowledge discovery by providing insight about the types of clustered nodes. For example, scene graph partitioning can easily reveal who the most influential people in the scene are, where the scene is geographically centered, and what the most important topics are on a scene.

Our hypothesis was that this second approach to graph generation might produce better results than first approach, while tremendously simplifying scene dis-

covery. This would extremely important for cultural scene computing, since social network data volume can explode quickly on networks with heavy user participation.

4.5. New Louvain Graph Partitioning Techniques

This section shows how Louvain graph partitioning has been utilized and modified for scene detection.

4.5.1. Revealing Community Structure

The steps listed above generated graphs from "Meetup.com" data, but did not partition them into scenes. Our goal was to reveal scenes boundaries using network analysis and graph partitioning; discovering community structure and maximizing modularity by analyzing which nodes were most densely interconnected. Modularity (Q) maximization approaches partition graphs on the principle that a set of nodes are highly likely to be in the same module if they are densely interconnected as a cluster, relative to their sparser connection to other modules. Many natural networks, informal human networks, organizational networks and system networks in fact exhibit this kind of modular structure, also called community structure. Since scenes are informal human networks, it was reasonable to hypothesize that modularity maximization would reveal the community structure of scenes.

The Louvain method is a well accepted and widely used modularity maximization approach for discovering communities in large networks. The main advantage of the Louvain method is that it is very fast (e.g., in one experiment it was able to analyse 118 million nodes in 152 minutes) [15]). It also provides a generally acceptable degree of accuracy. This is extremely important for scene discovery due to the large size of social networks and that fact that data in such networks grows exponentially over time. However, what makes this approach appealing in our case is that scenes exhibit hierarchical structure; super-scenes may include several sub-scenes. This exactly matches how the Louvain method works.

4.5.2. "Reverse" Louvain

The Louvain method is an iterative algorithm with two phases. First, it searches small communities by optimizing modularity locally. This is done by assigning each node i in the network to a group (module) then calculating the gain in modularity ΔQ of merging the node i with each of its neighbor communities C, as shown in Equation 2. Where $\sum_{in}$ is the sum of the weights of the links inside C, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C, k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C, and m is the sum of the weights of all the links in the network. Based on the results, the node will be added to the module that maximizes network modularity.

$$\Delta Q = \left[\frac{\sum_{in}+k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \quad (2)$$

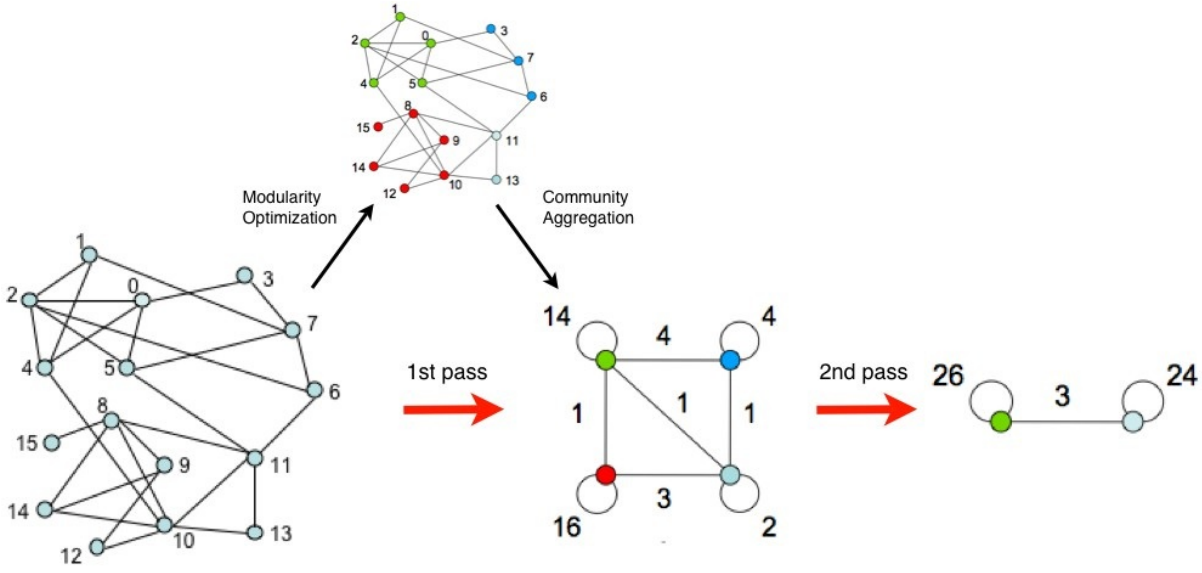


Figure 9: Louvain Modularity Algorithm [Adopted form [15]].

The second phase aggregates nodes within the same community, to build a new network whose nodes are communities that themselves are more densely interconnected than the relatively more sparse connections between nodes in the new super-community. These steps are repeated until a maximum modularity is achieved as shown in Figure 9.

In our effort to detect scenes and sub-scenes, we exploited this hierarchy-generation feature by reversing the process. We applied the method to the whole network to discover the main scenes within it. Then we recursively applied it to each of the identified main scenes to discover sub-scenes within them

4.5.3. Louvain on a Simple Directed Graph

The Louvain method takes an adjacency matrix as input; hence it can be applied to undirected graphs whether or not they are weighted. In Method One we applied the Louvain method quite conventionally to our three similarity graphs: weighted undirected graphs that grouped people, or topics, or locations into communities.

We also applied it quite unconventionally to our scene graph, which was an unweighted directed graph. We treated it as if it was an unweighted, undirected graphs for the purposes of group the scene graph nodes into communities, which immediately revealed scene structure. Then we applied fan-in and fan-out analysis using directional information to identify types of nodes within the discovered scenes.

These unconventional uses of the Louvain modularity maximization algorithm generated necessary results for us, and they constitute key contributions of our research.

4.6. Scene Analysis

Once the graph had been partitioned, and the boundaries of the scene were identified, depending on the type of the graph, the following techniques were applied to further reveal and analyse scene structure:

(a) **Facet Filtering:**
Facet filtering [61] is a multidimensional technique that uses different data properties to organize information into groups to facilitate its exploration and navigation [62]. In our work, after our two differently produced graphs were clustered as described above, we exported them and merged them with the original dataset as new labels (facets). Each record was thus described using four additional clustering classes; namely, people clusters, topic clusters, location clusters, and scene clusters (the clusters generated through scene graph partitioning). Facet filtering could then be used to filter the records based on commonalities and mutual exclusions across the different clusters. This cluster overlapping would help find people who conducted activities in the same places around the same topics of interest. In other words, it could help identify people, topics and locations that best represented the scene core or center as shown in Figure 10. Moreover, facet filtering could be used to evaluate the quality of clustering in both scene graph and similarity based clustering.

(b) **Fan-In Analysis:** In directed graphs, fan-in analysis is a measure of the number of links that are directed toward a node. In a scene graph, edges connect people to their topics of interest and locations. Accordingly, fan-in analysis can help identify the popularity of topics or locations within a scene. This can reveal the main topics that specify a scene, or its significant locations.

(c) **Fan-Out Analysis:** In directed graphs, fan-out analysis is a measure of the number of links that are directed out from a node. In scene graph, fan-out analysis could help identify the most active people in a scene.

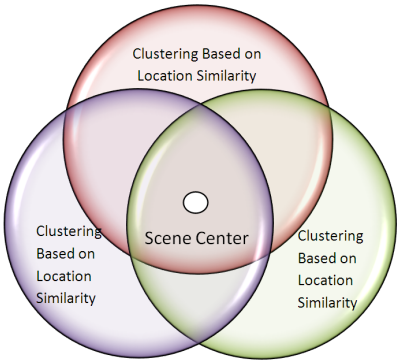


Figure 10: The Scene Center is the Inner Join of People, Topics and Locations Graphs

These simple techniques were essential for uncovering the main scene facets for each of the identified partitions. However, many additional network analysis techniques could be usefully applied to these graphs, generating insights that might help investors frame, find and answer questions about cultural scenes. Some of the relevant network measures are discussed Section 7.

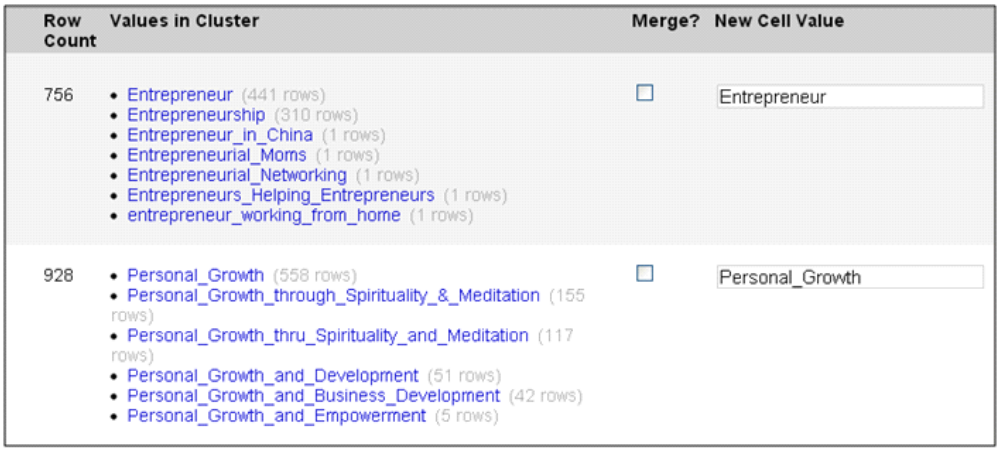


Figure 11: Clustering With Open Refine

5. Application of the Methods and Experimental Results

5.1. Case Study Description

As explained earlier, the dataset used in this study was crawled from Meetup.com using the process explained in Section 4.2. The data was collected and reconciled based on shared keys between the different datasets. In total, information about 150 groups were collected. The collected groups were all located within 25 miles of Waterloo, Ontario, and distributed between 14 urban communities. Out of those 150 groups, 132 groups were open, 1 was closed and 7 had only recently been approved. Out of the 132 open groups, only 123 groups were publically accessible. 813 topics within 28 categories were used to describe these groups with, an average of six topics per group. The total number of members including duplicates was 13,735 users; since a user can be a member of several groups.

Dataset	Location	Groups	Communities	Topics	Categories	Users
Dataset 1	Within 25 miles from Waterloo	150	14	813	28	13,735
Dataset 2	Kitchener and Waterloo	2 selected	2	364	23	100

Table 1: The Data Set Summary

In addition to the main dataset, a subset of the crawled data was created for validation. This subset was studied thoroughly to make it serviceable as a ground truth for qualitative evaluation of the clustering. Table 1 summarizes the parameters of the two datasets. Table 2 shows example of the crawled data.

5.2. Data preprocessing and refinement

Several decisions had to be made to prepare the data for analysis. First, groups and users who made their data private were filtered out. Then, topical dimensionality was reduced. After that, members with unusually many topics of interest were removed. Finally, data inconsistencies and special characters were treated.

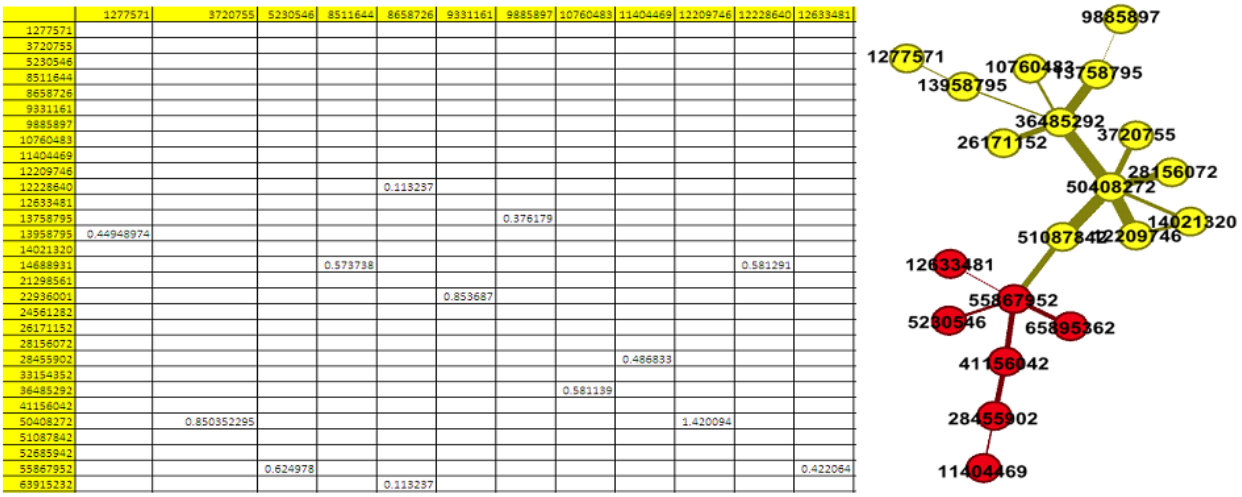
Data preprocessing and refinement was conducted using Open Refine (previously known as Google Refine). Open Refine is an open source tool for refining messy data, cleaning it up, and transforming it from one format into another [63]. It facilitates facet analysis and provides a set of clustering techniques out of the box. It also allows users to review clustering results before reflecting them back into the original dataset as shown in Figure 11.

Two clustering techniques were chosen based on their performance in finding phonetically and syntactically similar topics. The first was a key collision technique based on the Metaphone3 phonetic algorithm [64]. The second was based on a variation of the k nearest-neighbor algorithm that uses Levenshtein distance [65] to measure the similarity/difference between topics (i.e., strings). Combining both clustering techniques reduced the number of topics by an average of 11.5%.

Figure 11 shows the manual part of the process, in which users are given the choice to accept or reject merging the suggested similar raws under the same cluster. For example, based on Metaphone3, the algorithm suggests that both the topics “Entrepreneur” and “Entrepreneurship” should be clustered under the same topic “Entrepreneur”. By accepting this suggestion, 441 raws will be merged with 310 raws to have a bigger cluster of 751 raws.

Member Id	Topics ID	Topics Name	Member City	Member Groups	MGroups Cities	MGroups CategoryID	MGroup Category Name	MGroup Topics
12209746	1044	Event_Planning	Waterloo	Web Designers /Developers	Waterloo	34	tech	Web_Design
	10290	Spirituality						Graphic_Design
	4417	Consciousness						Web_Technology
	15478	Holistic_Health						Internet_AND_Technology
	15018	Music						Web_Development
	1322	Meditation		Gamers	Waterloo	11	games	Boardgames
	243	Alternative_Health						Dungeons_AND_Dragons
	17866	Meeting_New_People						Live_Action_Role_Playing
	10581	Social						Shadowrun
	2278	Drum_Circle						War_Games
	79103	Healing _Drum_Circle						Star_Wars_RPG
	17570	Drumming						D20_Gaming
	21309	Recreational_Drumming						White_Wolf
	16733	Hand_Drumming						Roleplaying_Games_(RPGs)
	496	Game_Development						Strategy_Games
								Gaming
								PC_Gaming
								Computer_Gaming
				Groovers	Kitchener	21	Music	Drummer
								Alternative_Health
								Meditation
								Drum_Circle
								Consciousness
								Live_Music
								Hand_Drumming
								Meeting_New_People
								African_drumming

Table 2: Example of The Crawled Data



	1277571	3720755	5230546	8511644	8658726	9331161	9885897	10760483	11404469	12209746	12228640	12633481
1277571												
3720755												
5230546												
8511644												
8658726												
9331161												
9885897												
10760483												
11404469												
12209746												
12228640					0.113237							
12633481												
13758795							0.376179					
13958795	0.44948974											
14021320												
14688931				0.573738							0.581291	
21298561												
22936001						0.853687						
24561282												
26171152												
28156072												
28455902									0.486833			
33154352												
36485292								0.581139				
41156042												
50408272		0.850352295								1.420094		
51087842												
52685942												
55867952			0.624978									0.422064
63915232					0.113237							

Figure 12: People Similarity Matrix and Its Corresponding People Graph

After applying facet filtering to the dataset, it became apparent that a few members have very large numbers of topics of interest (i.e., over 100 topics). A decision was made to exclude those members by removing the top 3% of members with highest topic counts. We surmised that these members did not have focal interests, but were rather Meetup trackers. Their presence in the dataset would have been deleterious to the clustering algorithms. Finally, all trailing spaces, inconsistencies, symbols, and special characters were removed or replaced (e.g., replacing & with AND) in order to avoid errors while transforming data from one format into another throughout our multi-step prodcedures.

5.3. Scene Discovery Results

After data refinement, the dataset was transformed into the four types of graphs our methods produce (people, topic, location, and scene graphs) based on the two techniques described in Section 4.4. The graphs were then exported to the Gephi visualization and analysis platform. Gephi is an open-source interactive visualization and exploration platform for networks, complex systems, dynamic and hierarchical graphs. It provides powerful tools that implement several statistical analysis, filtering and visualization algorithms that can be applied directly to graphs [59].

Figure 12 shows a sample of a *people similarity matrix* and its corresponding *people graph*. Each node in the graph represents a person; an edge between two people indicates a similarity between them, while the edge weight (thickness) corresponds to the strength of the relationship. Location and topic graphs are similar to the people graph; they consist of one type of nodes with weights on the edges. The Louvain algorithm was applied to all of the graphs that were based on similarity matrices (i.e., people, topics, locations). The result of clustering was exported and reconciled with each record in the dataset for evaluation.

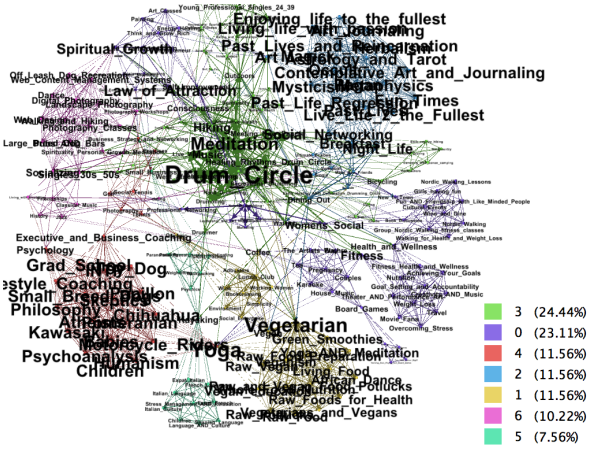
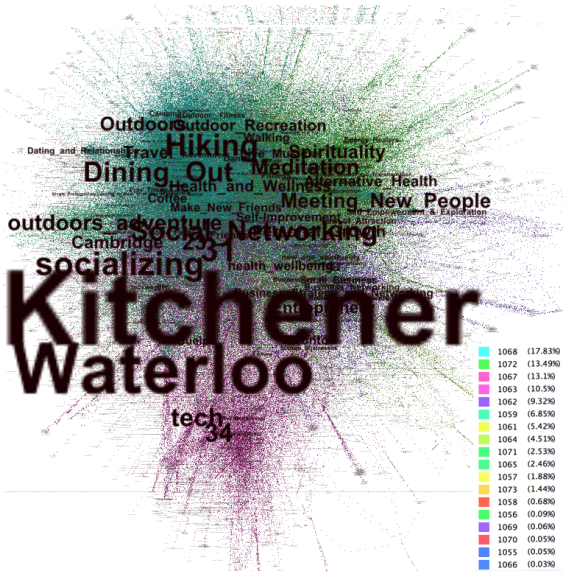
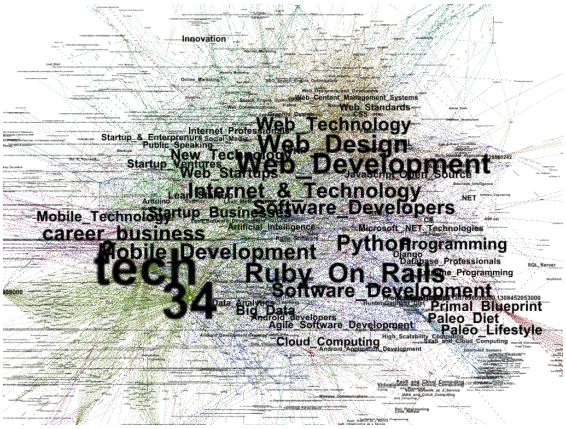


Figure 13: Topics Graph After Applying Louvainś Modularity

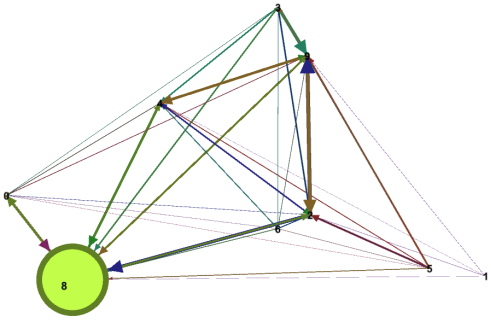
Figure 13 shows an example where modularity partitioning was applied to a *topic graph* using dataset 2 and the people similarity analysis. As shown in the figure, topic graphs express high modularity (i.e., Q = 0.713). Seven communities were identified; and within each community the topics were ranked using the degree of connectivity as a metric to show the importance and centrality of the topics. For example, in Figure 13, it is clear that the Drum Circle was a centralized topic within the main community in dataset 2.



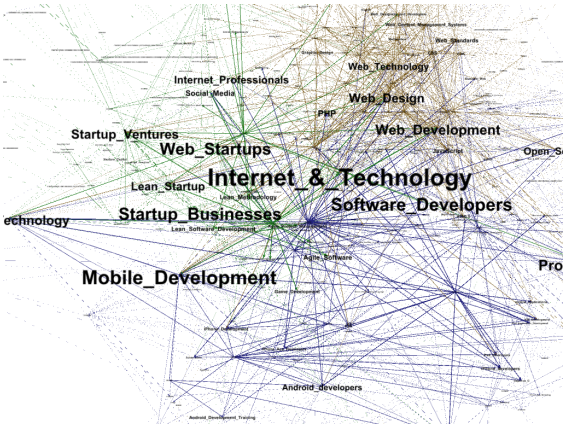
(a) The Scene Graph Generated From Dataset 1 After Partitioning



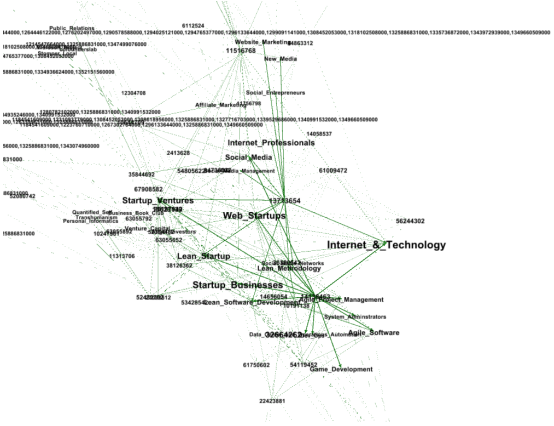
(b) Waterloo Technology Scene



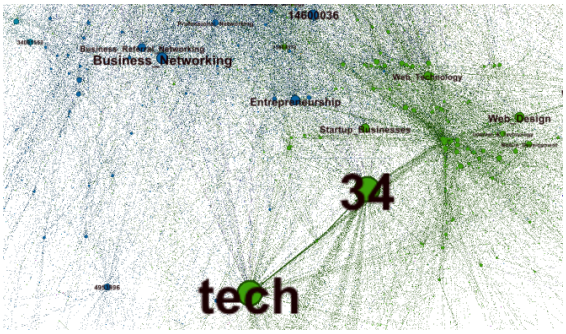
(c) The Relationships Between The Generated Communities Within The Technology Scene



(d) Mobile Development, Technology Start-Ups and Web Development Communities



(e) Example of Waterloo Technology Startup Scene That Shows How The Scene Graph Partitioning is Able To Capture All The Scene Concepts (Topics, People, and Location)



(f) Soft Clustering in Scene Graphs

Figure 14: Scene Detection and Analysis Process

Now, given the fact that this dataset was built using two groups; one of them is the Organic Groove Community Drummers, which was mainly a drumming circle meetup. It is apparent that the topic graphs can be effectively be used to discover communities in cultural web data. However, our goal was to discover scenes, not topical communities. The goal was to cluster people, location and topics all together. In order to create scenes, the results of clustering topics, locations, and people had to be merged. This was done by overlapping the partitioning results using facet filtering. More about this will follow in the evaluation section.

We proceeded as described earlier with applying the Louvain algorithm , unconventionally, to the *scene graphs* (directed unweighted graphs with different node types) we generated from the dataset. Figure 14a shows an example of applying the Louvain algorithm on the scene graph generated from dataset 1. 1070 communities were identified. The maximum modularity was (Q = 0.469) which can be considered adequate. Out of the 1070 communities identified, 14 communities represent 90% of the whole dataset. This is because in large social networks, modularity optimization often fails to detect clusters smaller than some scale [66]. For this reason, we applied modularity optimization in iterations. However, the preliminary results of this clustering strategy were already appealing. For example, when emphasizing fan-in analysis (by using it as a ranking factor to enlarge the node and label size), the main topics and topics groups (e.g, 34 which refers to the technology topic group) in each scene became visible. (Waterloo Region is known to be a center for the technology scene in Ontario). This was obvious in the result, which showed technology as the topic category with highest fan-in. The results of clustering via this method were also exported and reconciled with the records in the dataset for further analysis and evaluation.

At this juncture, it is important to highlight the concrete advantages of applying the Louvain algorithm in multiple iterations to discover sub-scenes. For example in Figure 14b, the graph in Figure 14a was filtered to show only the technology scene. The Louvain algorithm was then applied to the technology scene alone, which partitioned it into 9 technological communities with maximum modularity of (Q = 0.456). Figure 14c shows the relationships between the generated communities. In this figure, the bigger is the community, the bigger the size of the node. Figure 14d focuses on communities 3, 4, and 9, which refer to mobile development, technology start-ups and web development, respectively. Note that all these figures present fan-in analyses, revealing the topical dimension of the scene. Nevertheless, these scenes also include people and location nodes as shown in Figure 14e. In which, we zoomed in to show the labels of the different nodes that may not be apparent due to its low raking based on fan-in analysis. On the other hand Figure 14f shows the soft clustering characteristic in scene graphs. "Entrepreneurship" lies between the technology start-up scene and the business networking scene. Soft clustering is one of the main characteristics of scene graphs that facilitate discovering new scenes.

6. Evaluating the Scene Discovery Results

It is challenging to evaluate the results of scene discovery efforts without any ground truth data. Evaluating clustering approaches is known to be hard if no ground truth data is available. In fact, this is considered an open research problem. This section presents the techniques we used to evaluate our scene clustering results.

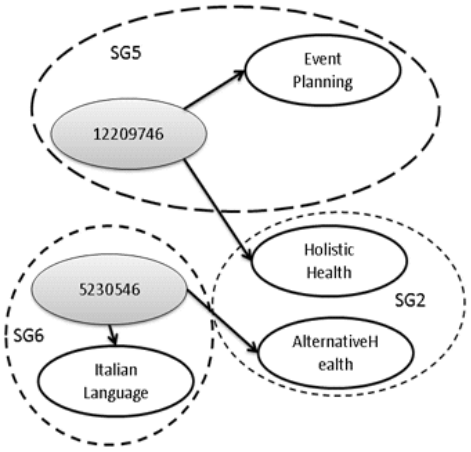


Figure 15: An Example Shows How One Record Is Distributed Into Different Partitions

After graph partitioning, each node belonged to one cluster. For example, as shown in Table 3, the person with member ID (12209746) belonged to cluster SG5 when the partitioning is done using the *scene graph* method. The same person belonged to partition T2 in the case of *people graph* partitioning, and P2 is the case of *topic graph* partitioning.

The nodes with their corresponding partitions were then reconciled with the original records information (each record in the dataset was matched with its corresponding partition). As illustrated in Figure 15, one record could be distributed across different partitions depending on whether the reconciliation was done based on the topic, location, or member ID. Table 3 shows a sample of records after reconciliation. The

Member ID	Topic Name	...	People Clustering Topics Similarity	Topics Clustering People Similarity	Scene Graph People	Scene Graph Topics
12209746	Event_Planning	...	P4	T2	SG5	SG5
12209746	Spirituality	...	P4	T2	SG5	SG2
12209746	Consciousness	...	P4	T2	SG5	SG5
12209746	Holistic_Health	...	P4	T2	SG5	SG2
12209746	Music	...	P4	T2	SG5	SG5
. .	. .	...	. .	. .	. .	. .
5230546	Italian_Language	...	P5	T3	SG6	SG6
5230546	French_Language	...	P5	T3	SG6	SG6
5230546	Hiking	...	P5	T2	SG6	SG2
5230546	Walking	...	P5	T2	SG6	SG6
5230546	Spanish_Language	...	P5	T3	SG6	SG6
5230546	Dining_Out	...	P5	T4	SG6	SG1
5230546	Childfree	...	P5	T3	SG6	SG6
5230546	Alternative_Health	...	P5	T2	SG6	SG5
5230546	Meditation	...	P5	T2	SG6	SG5
5230546	Outdoor_Recreation	...	P5	T2	SG6	SG2

Table 3: A Sample of Records After Reconciliation.

evaluation case considered only topics and people; since the location in the dataset was the same for all the scenes (i.e., Kitchener/Waterloo).

The outcome of the scene discovery methods were evaluated based on three evaluation criteria; namely, (1) scene relevancy (i.e., precision, recall, F1-score) and community size (2) Jaccard similarity, and (3) modularity. The following subsections explains them in more detail.

6.1. Scene Relevance and Community Size

The ability to discover and retrieve relevant scenes depends on the quality of scene partitioning, which was evaluated based on calculating the scene topics (i) precision, (ii) recall, (iii) F1-Score. In addition ,we assessed (iv) the number of people within the scene (cluster size). The evaluation relied on two hypotheses:

(a) **Hypothesis 1**: When a scene discovery approach is applied to a dataset that consists of well-defined cultural groups (e.g., Meetup groups), then the minimum number of scenes should be at least equal the number of groups, with each scene centralized around the topics that describe each group.

(b) **Hypothesis 2**: A scene discovery approach that provides higher precision, recall and a larger cluster size is better. Precision and recall are calculated with respect to scene topics, while cluster size is based on the number of people within the scene.

Cluster size is important for Sceneverse because a new digital service aimed at enhancing scenes needs to successfully engage the more peripheral participants in any scene. Core/central participants are already well informed of scene events and opportunities through word of mouth, cultural news media, existing online social media and the like. To add value within the existing cultural media landscape, Sceneverse needs to detect marginal participants and increase the frequency of their participation in scene activities. That increased participation will funnel more support towards the efforts of scene activity organizers at the scene's core.

To find the overlap between topics and people clusters, and to analyse the results; facet analysis was performed using the different clustering results on the reconciled dataset. The dataset used in the evaluation process is a subset of dataset 2. It was generated around the "Organic Groove Community Drummers" group. Accordingly, and based on Hypothesis 1, the topics that describe that group will definitely represent at least one of the scenes within that group. Most of the time, it will be the largest scene within that group.

Table 4 shows the clustering results after (1) converting the dataset into a topic graph, a people graph and a scene graph, and (2) partitioning the graphs using the Louvain algorithm. The first column shows the similarity analysis results for the overlap of the topic and people graphs (method one), while the second column shows the results derived from the scene graph (method two).

	Similarity Analysis Graphs	Scene Graph
No. of Topic Clusters After Applying Partitioning	6	6
No. of People Clusters After Applying Partitioning	6	6
Total No. of Clusters	12	6
Total No. of Topics in the Graph	225	225
Total No. of People in the Graph	32	32
Total No. of Sample Records (sum of all fan-ins)	475 Records (a person has multiple topics)	475 Records
Largest Topics Cluster	274 Records (47 topic & 30 ppl)	172 Records (36 topic & 29 ppl)
Largest People Cluster	99 Records (49 topic & 6 ppl)	122 Records (47 topic & 11 ppl)
Inner Join of Topics & People (Topics ∩ People)	81 Records (33 topic & 6 ppl)	111 Records (36 topic & 11 ppl)

Table 4: Clustering Results After Applying The Different Graph Partitioning to the "Organic Groove Community Drummers" Dataset

To calculate precision, the topics (TP) with the highest fan-in within the inner join of the largest topic (T) and people (P) clusters were compared to the original group topics ($TP_{Original}$) (i.e., the Organic Groove Community Drummers). Equation 3 shows how scene precision has been calculated using Similarity Analysis (SA). Similarly, the scene graph topics ($\Pi_{TP}\{SG\}$) of the largest scene detected were compared with the original group topics. Equation 4 shows how scene precision has been calculated for the Scene Graph (SG) results. In the equations, the Π symbol represents the topic projection, while $\bowtie$ represents the join of the topic and people graphs, i.e., the scene graph as derived by this method

$$\text{Precision}_{SA} = \frac{|\Pi_{TP}\{T_G \bowtie P_G\} \cap \{TP_{Original}\}|}{|\{TP_{Original}\}|} \quad (3)$$

$$\text{Precision}_{SG} = \frac{|\Pi_{TP}\{SG\} \cap \{TP_{Original}\}|}{|\{TP_{Original}\}|} \quad (4)$$

Equation 5 shows the formalism for assessing scene recall. Scene recall was calculated by comparing the number of records in the scene ($R_{\text{Scene-Retrieved}}$), where a user indicated interest in any topics used to describe the ground truth scene, to the number of records in the dataset that refer to the same topics ($R_{\text{Scene-Relevant}}$).

$$\text{Recall} = \frac{R_{\text{Scene-Retrieved}}}{R_{\text{Scene-Relevant}}} \quad (5)$$

$R_{\text{Scene-Retrieved}}$ can be calculated by summing all the fan-in values for all topics that constitute the scene, as shown in Equation 6.

$$R_{\text{Scene-Retrieved}} = \sum_{i=1}^{n} TP_i(\text{FanIn}) \quad (6)$$

On the other hand, $R_{\text{Scene-Relevant}}$ can be found by searching the records for scene-specific topics as shown in the following pseudo-code.

```
Input: Dataset
Output: Scene-Relevant
forall the Records r ∈ Dataset do
    if RecordTopic rtp ∈ SceneTopic then
        Scene-Relevant++;
    end
end
```

Table 5 shows the main topics that described the "Organic Groove Community Drummers". The table also shows the fan-in analysis of these topics in both the scene graph, as well as the inner join of both topic and people similarity graphs.

As shown in Table 6. Both techniques provide 100% precision with respect to the topics that describe the "Organic Groove Community Drummers" group. However, the scene graph technique provides higher recall with respect to the total number of records returned in which a user indicated an interest in any of the topics used to describe the Organic Groove Community Drummers group. However, the scene graph technique recalled more of the total number of records where users indicated interest in topics describing the Organic Groove Community Drummers group. Moreover, the size of the scene community in terms of participants identified is much higher using the scene graph technique; almost 92% higher.

Finally, to show the accuracy of the scene graph approach over the similarity analysis graph approach, the harmonic mean of precision and recall, or F1 score, has been calculated. The F1 score takes values between zero and one; the closer the value to 1 the higher the accuracy

Original	Similarity Analysis Graphs	Scene Graph
Alternative Health	3	4
Meditation	4	6
Drum Circle	6	11
Consciousness	3	3
Live Music	3	7
Social	3	3
Music	4	7
Hand Drumming	5	9
Drumming	4	6
Meeting New People	4	4
African drumming	4	7
Recreational Drumming	5	5
West African Drumming	4	9
Healing Rhythms Drum Circle	5	5
Social	3	3
Total Number of Records With Exact Topics	60	89

Table 5: Topics of The Organic Groove Community Drummers

of the information retrieval approach. Equation 7 shows how the F1 score is calculated. The results are shown in Table 6,where the F1 score confirms higher accuracy for the scene graph over the similarity analysis graph technique.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{7}$$

6.2. *Jaccard Similarity for Scenes with No Ground Truth*

A similar process that does not require ground truth data was applied to all other clusters (other than the main one which was used in the previous analysis). The process began by finding all the inner joins of the different combinations of people and topics graphs partitions. Then, for each partition in the scene graph, similarity was calculated to find the distance between each of the inner join sets and the partition. As shown in Equation 3, the Jaccard Similarity Index was calculated by finding the size of the intersection between the scene topics (A) and each of the inner join combinations (B) divided by the size of the union of the two sets. After calculating the Jaccard Similarity Index, it was apparent that more information was needed in order to reason scientifically about the results. For this reason, another similarity metric that also uses Jaccard Similarity was used to refine and confirm the results of the first metric.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

We call this metric scene theme similarity. As indicated by the name, this metric focused on the main scene topics; those shared by many Scene People. Scene theme similarity was calculated by creating a list

	Similarity Analysis Graphs	Scene Graph
Total Number of Records With Exact Topics Returned	60	89
Total Number of Records With Exact Topics In the Dataset	122	122
Precision (with respect to the original group topics list)	100%	100 %
Recall	49%	73%
F1 Measure	65.77%	84.39%
Number of people in the Scene	6 ppl	11 ppl

Table 6: Precision, Recall and Scene Size Results

Scene Graph Partition	Equivalent Topic ∩ People Graphs Partition	Jaccard Similarity Index	Theme Similarity Index	Total No. of Topics SG	Total No. of Topics (T ∩ P)	Total No. of People SG	Total No. of People (T ∩ P)
SG1	T4∩P2	0.547	1.0	40	26	4	2
SG2	T2∩P5	0.271	0.6	52	37	8	7
SG3	T2∩P1	0.442	0.2	22	40	2	6
SG4	T1∩T3	1.0	1.0	26	26	2	2
SG5	T2∩P4	0.605	1.0	36	33	11	6
SG6	T5∩P1	0.308	0.2	31	32	4	5

Table 7: Evaluation Results Using Jaccord and Theme Similarity

that consisted of the top five topics (highest in fan-in) for each scene and topic people inner join cluster. Then, the themes were compared using Jacquard Similarity.

As shown in Table 7, by combining both Jaccard and theme similarity, better insight into clustering results was attained. For example, if both Jaccard similarity and theme similarity were high (i.e., above 0.5), this was a good indication that the scene discovered had a well-defined boundary. Examples of such scenes were SG1, SG5 and SG4. In this case, the larger the size of the scene, the better the scene. For example, even though both similarity indices of SG 4 were high; it was considered to be a weak scene, because it was so small (2 people).

On the other hand, when Jaccard and theme similarity were both low, or in cases when theme similarity was higher than Jaccard similarity, manual data inspection was performed. We adjudged that scene graph partitioning provided more rational results. In the case where theme similarity was higher than Jaccard, the scenes discovered using the inner join method were composite mixed scenes (i.e, contained more than one scene). On the other hand, when both Jaccard and theme similarity were low, the scene was not clearly identified in the case of the inner join.

For instance, the SG6 theme contained the following topics: Walking, Camping and Kayaking, Backpacking, and Board Games. The (T5 $\bigcap$ P1) theme contained: Board Games, Executive and Business Coaching, Psychology, Yoga, Atheists. Clearly the topical theme of SG6 is more coherent and makes more sense for clustering as a scene than the one generated using the inner join method (T5 $\bigcap$ P1).

It is worth mentioning that the results obtained using these metrics is aligned with the results obtained when ground truth data were available. Both results favor scene graph over the people-topic inner join for scene discovery. Moreover, using either quantitative or qualitative analysis, a good scene still evinced the same properties; a cohesive community, strong central topics and significant numbers of people participating.

6.3. Modularity Metric Q

Modularity is a widely used metric to show how well a network is partitioned. Consequently, modularity was calculated for the three graphs generated here. The results of applying the Louvain method to the people, topic, and scene graphs are 0.647, 0.713, and 0.469 respectively. Overall modularity was adequate, indicating cohesive communities. The modularity of both people and topic graphs separately was higher than that for the scene graph. This result is expected since the more dimensions you add to the graph the less modularity you have. However, it is this characteristic that gives the scene graph its special property as a graph uniquely able to reveal the cultural contours of the scene.

7. Discussion

Graphs with a single node type that are created using similarity analysis play important roles in recommendation systems. Such graphs can achieve high modularity when applying graph partitioning, and hence they delineate coherent partitions or communities. Unfortunately, communities identified in these graphs fail at representing cultural scenes. This is because a scene is a multidimensional concept, in which time, location, topics and people all contribute to define the scene boundary. A workaround can be devised by first creating a graph for each of the dimensions, then partitioning (clustering) these graphs, and finally finding the inner join of the different partitions. Unsurprisingly, the results of this workaround are disappointing because of the size of the communities generated. Overlapping the communities is a type of hard clustering. It assists in identifying the scene center, but fails to capture the whole scene. In order to be able to identify the boundaries of the scene

efficiently, there is a need for either (1) efficient techniques that can combine different similarity measures, or (2) similarity measures that work the same way with different types of objects.

The scene graph approach was created to address this particular problem. It creates a graph that combines different types of nodes. Then it uses graph modularity to partition the graph. From the graph partitioning algorithm perspective, all nodes are the same, despite their scene-dimension type. In order to preserve as much information about the node types in the graph as possible, the scene graph was constructed as a directed graph. Moreover, a record reconciliation process, followed by facet filtering, was used to merge the partitioning results with the original records in order to further analyse the clustering results.

7.1. The Main Findings

Scene graph partitioning is a soft clustering technique. That is designed specifically to discover scenes in cultural data. It is easier faster and more suitable for discovering cultural scenes than single facet graphs and their overlaps. This is shown in the evaluation experiment on the small Meetup dataset, in which the scene graph method outperformed the method based on graph similarity and overlap by almost 53% in terms of execution time. The 53% has been obtained by comparing the time needed for scene detection using the scene graph, and the sum of the time needed to partition and overlap the people, location, and topics graph. Moreover, the quality of the scenes generated using the scene graph method demonstrated much more complete and representative scene communities, with almost a 92% larger community size and 18.62% higher accuracy, based on the F1 measure.

7.2. Dependency on Louvain Graph Partitioning

Both methods proposed in this paper depend on Louvain graph partitioning. In fact, the performance bottleneck for the scene graph approach stems from its dependency on Louvain graph partitioning for community detection. The scene graph calls the partitioning algorithm recursively; and this has obvious implications for the efficiency and scalability of the partitioning algorithm in large datasets. In a recent study, Papadopoulos et al. [26] have compared the performance of existing community detection techniques, and they favoured the Louvain method over other methods for large scale graphs such as social networks. This study supports Papadopoulos findings. We compared six different community detection techniques(i.e., Louvain [15], Fast Greedy [67], Leading Eigen Vector [68], Walktrap [71], Label Propagation [69], and Infomap [70]) in terms of complexity, performance, modularity and the number, the size and structure of the generated communities. We applied the different techniques on the first Meetup dataset used for scene detection (i.e, 10781 vertices and 61151 edges).

Table 8: Comparison of community detection algorithms in terms of complexity

Method	Actual Complexity	Sparse Graphs Complexity
Louvain [15]	$O(n^2)$	$O(n)$
Fast Greedy [67]	$O(n^2 d \log n)$	$O(n \log^2 n)$
Leading Eigen Vector [68]	$O(n^3 d)$	$O(n^2 \log n)$
Walktrap	$O(n^4)$	$O(n^2 \log n)$
Label Propagation [69]	$O(n^2)$	$O(n)$
Infomap [70]	$O(n^2 \log n)$	$O(n \log n)$

Table 8 compares the complexity of each of the aforementioned methods. The first column in the table represents the complexity obtained without any assumptions about the underlying graph, while the second assumes a sparse graph, in which the number of vertices is approximately equal the number of edges. The complexity of community detection algorithms is usually expressed in terms of the number of vertices (n), number of edges (m), as well as the depth of the tree (d) when hierarchical methods are used. Complexity gives a good indication of how well the algorithms will perform as the dataset size reaches infinity. The results of comparing complexity shows us that the Louvain method is one of the best candidates in terms of complexity. Other possible candidates include Label Propagation and Infomaps.

Complexity analysis helps us understand the scalability of the algorithm. However, the experimental results better illuminate the performance of the algorithm with respect to the different dataset types, sizes, and other factors that can impact the algorithm performance (e.g., memory consumption).

The radar chart in Figure 16 shows a multivariate comparison between the aforementioned community detection algorithms, in terms of modularity and execution time. Figure 16 shows the results of applying the different algorithms on the Meetup dataset. The results show that the Louvain method has the highest modularity and the lowest execution time over all other techniques for this specific data-set. The Label Propagation

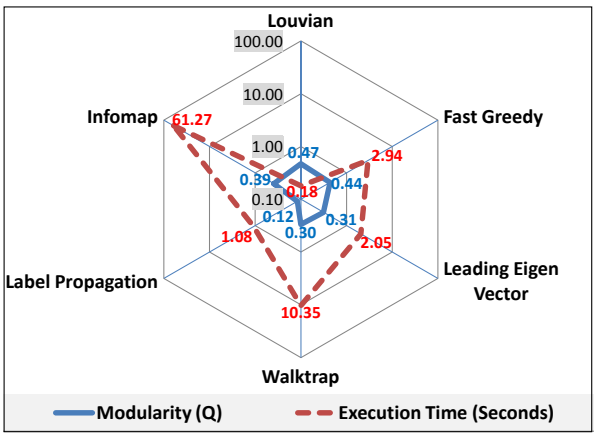


Figure 16: A radar chart displaying a comparison between six different community detection algorithms in terms of modularity and execution time

method, which shares the same complexity metrics as the Louvian method, performs more poorly by comparison, and hence it has been eleminated.

The suitability of the Louvian method for this task has been further confirmed by comparing the methods in terms of communities size and structure as shown in Figure 17.

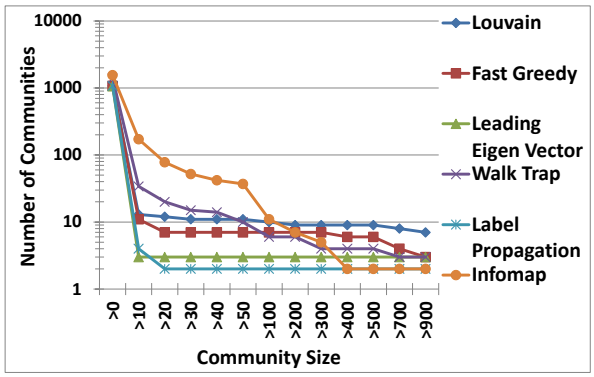


Figure 17: Number of Communities Detected Based on Community Size

The Figure shows the number of communities detected by each method with respect to the size of the community. It is clear from the figure that the Louvian method tends to generate more structured communities of significant size. Conversely, Infomap generates large number of very small communities. These Infomap-derived communities are not very representative of scenes, as explained in the evaluation section.

By applying the Louvain method recursively on the detected communities, more sub-communities can be revealed in a meaningful hierarchical structure. For example, when applying the Louvain method in two steps on the Meetup dataset, 128 sub communities were detected. The whole process took around 1.78 seconds an average modularity of 0.62. This result is comparable to the one achieved by Infomap. However, the communities here are more meaningful and follow a hierarchical order that preserves the consistency of topic information.

Moreover the recursive Louvain method outperforms Infomap. It is 3442 times faster on this dataset. The number itself rapidly scales in a linear relationship with the scale of the graph.

7.3. Analysis Techniques

Once scenes have been discovered, and their topics and key people have been identified, a second round of analysis is needed to rank people within each scene based on centrality. In this paper fan-in and fan-out analysis was used. However, more factors and many additional analytical techniques could be applied to reveal these scene relationships.

For example, factors such as the distance between a person's place of residence and the scene's central location, or the duration in time of a person's connection to a topic, could significantly affect the assessed strength of the connection between the person and the scene. Several other analytical techniques could also be used; for example:

(i) **Centrality:** Roughly indicates the social power of a node based on how much the connectivity of a network depends on it (i.e. how disconnected the graph would be if that node were removed). Betweenness, Closeness, and Degree are all centrality measures.

(ii) **Betweenness:** The extent to which a node lies between other nodes in the network. This takes into account the connectivity of the node's neighbours, giving a higher value for nodes which bridge clusters. The measure reflects the number of people a person is connected to indirectly through their direct links.

(iii) **Closeness:** The degree an individual is near all other individuals in a network (directly or indirectly). It indicates the node's ability to access information through the grapevine of network members.

(iv) **Bridging:** A relationship is a bridge if deleting it would cause its endpoints to lie in separate components of a graph.

(v) **Density:** The degree a respondent's ties know each other, i.e. the proportion of ties among an

individual's group of contacts. Network-wide or global density is the proportion of ties in a network relative to the total number possible. There can be sparse versus dense networks.

(vi) **Ego-effect:** The degree an individual's network reaches out into the network and provides it with novel information and influence.

(vii)) **Structural cohesion:** The minimum number of members who, if removed from a group, would disconnect the group.

In the context of scene discovery, the output from these social network analysis techniques would have to be combined and mapped to scene concepts. For example, a combination of fan-out, bridge and centrality analysis might reveal different ontological roles for people within a scene.

The application of these additional social network analysis techniques is out of scope for this paper. A later paper will be dedicated to discussing their use in the context of scene discovery.

8. Conclusions and Future Directions

In this paper, we argued the need for a platform dedicated to facilitate engaging its users in socio-cultural activities, and the need for tools and techniques that can reveal socio-cultural communities in existing datasets. We introduced Sceneverse, a proposed platform for supporting the creation and analysis of an online sociocultural universe. The proposed Sceneverse platform consists of several components. This paper focused on the concepts, techniques and methods used to implement the scene extraction engine component. Accordingly, we first created a scene ontology to provide a crisp understanding of the scene concept, and to enable building scene representations from the cultural and social data available on the web. We then devised an approach for automatic scene discovery in that data.

Scene discovery depends on the ability to cluster similar people, who have similar interests, expressed around similar events and venues, in certain locations, within a general span of time. This is challenging, since most clustering techniques work on single facet, and the scene is a multifaceted concept. To deal with this challenge, two techniques were examined. In the first technique, social and cultural data were first converted into three types of single faceted socio-cultural graphs; one for people, one for topics and one for locations. Each of these graphs were then partitioned into groups based on Louvain modularity optimization, and the resultant communities were overlapped to create scenes. In the second technique, a scene graph, which is a multifaceted directed graph, was created. Then it was partitioned directly into scenes using Louvain modularity optimization.

The two proposed methods were empirically evaluated using data crawled from the cultural and social network Meetup.com. Preliminary results demonstrate the superiority of the scene graph technique over the overlapping of single-faceted graphs in identifying the scene boundaries. While both techniques were able to detect the scene center in terms of the key people, main topics and central locations for a scene, the size of the community in terms of number of people identified was larger with scene graph partitioning. This is because a scene graph partitioning is a softer clustering technique than community overlapping.

The scene graph technique proposed in this paper overcomes two of the main drawbacks associated with graph partitioning. The first drawback is the information lost when converting a multifaceted dataset into graphs. We overcame this problem by preserving relational information in directed graphs, then using fan-in and fan-out analysis to highlight the different nodes in each partition. The second drawback is specifically related to modularity optimization techniques, in which modularity at a large scale fails to reveal small communities. This problem has been addressed by applying the Louvain algorithms in iterations. In fact, taking this approach proved ideal for scene discovery, since it organizes scenes in hierarchical order (scene, sub-scene), which fits the natural social topology of scene. Scene graphs are designed specifically to discover scenes in cultural data. Scene graph analysis using reverse Louvain optimization is easier, faster and more suitable for discovering cultural scenes than overlapping single facet graphs.

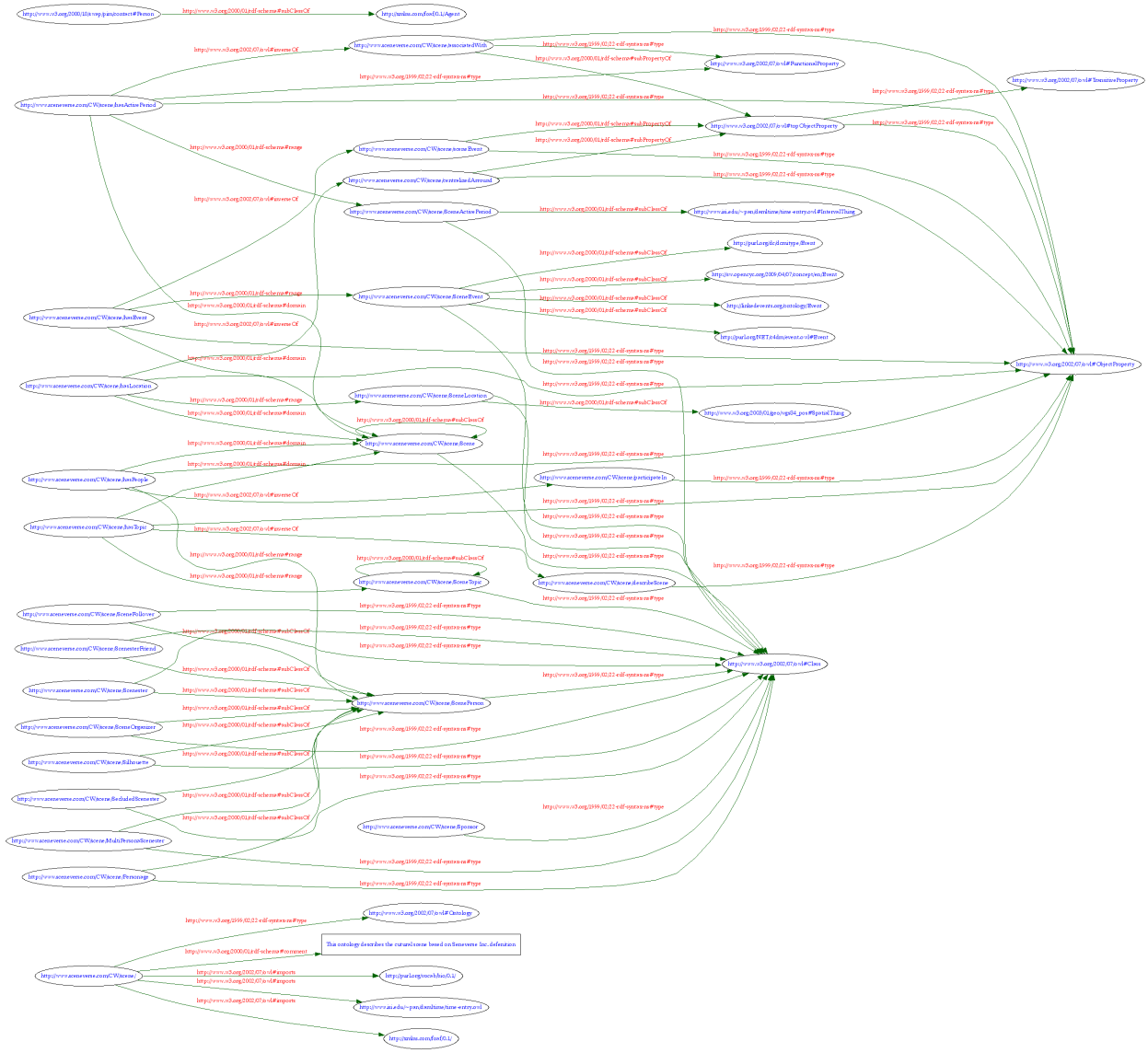
In addition to our scene discovery procedure, this paper also briefly discussed scene analysis techniques. We used fan-in, fan-out and facet filtering to discover important scene graph nodes. Future efforts will focus on investigating other analysis techniques to rank people, topics and locations within each scene (e.g., based on centrality or other social network analysis measures). The results will be used in prototypes of the Sceneverse platform to enhance scene experience and to provide scene participants with better choices. Moreover, the scene discovery approach proposed in this paper is potentially generalizable to other domains. Hence, we are planning to apply this technique to other domains where multifaceted recommenders are needed.

References

1. Hebdige D. Subculture: The meaning of style. *Critical Quarterly* (2007);**37**(2):120–4.
2. Maffesoli M. *The time of the tribes: the decline of individualism in mass society*; vol. 41. Sage Publications Limited; 1996.
3. Cova B. Community and consumption: Towards a definition of the linking value of product or services. *European Journal of Marketing* (1997);**31**(3/4):297–316.
4. Kozinets R. Utopian enterprise: Articulating the meanings of Star Treks culture of consumption. *Journal of Consumer Research* (2001);**28**(1):67–88.
5. Silver D, Clark T, Yanez C. Scenes: Social context in an age of contingency. *Social Forces* (2010);**88**(5):2293–324.
6. Hume C. Art of the city: Exhibit shows how relationship with cultural creativity, or lack thereof, can make or break a metropolis. *Toronto Star* (2001);Online ed, March 10.
7. Straw W. Scenes and sensibilities. *Public* (2002);**22**(23):245–57.
8. Nordicity . Sound Analysis: An examination of the Canadian Independent Music Industry. Tech. Rep. NGL13-02-15; Canadian Independent Music Association (CIMA); 2013. Retrieved from: http://www2.cimamusic.comwp-content/uploads/2013/07/Sound_AnalysisĊIMA_ḞINAL_2013p̀df [last accessed: October 2013].
9. Rothfield L, Coursey D, Lee S, Silver D, Norris W, Hotze T, et al. Chicago Music City: A report on the music industry in Chicago. Tech. Rep. ISBN 0-9747047-3-3; The Chicago Music Commission; 2006. Retrieved from: http://74.220.219.62/ñatkinne/chicago-music.org/wp-content/ uploads/2008/12/ chicagomusiccity_fullreport1.pdf [last accessed: October 2013].
10. Beyers WB, Bonds A, Wenzl A, Sommers P. The Economic Impact Of Seattle's Music Industry: A Report for the City of Seattles Office of Economic Development. Tech. Rep.; University of Washington and City of Seattle's Office of Economic Development; 2004. Retrieved from: http://web.williams.edu/ Economics /ArtsEcon /Documents /Seattle _Music _StudyFinal.pdf [last accessed: October 2013].
11. Jang H, Choe S, Song J. Exploring serendipitous social networks: sharing immediate situations among unacquainted individuals. In: *The International Conference on Human Computer Interaction with Mobile Devices and Services*. 2011, p. 513–6.
12. Shank B. *Dissonant identities: The rock'n'roll scene in Austin, Texas*. Wesleyan; 2011.
13. García-Crespo Á, Colomo-Palacios R, Gómez-Berbís JM, Ruiz-Mezcua B. Semo: A framework for customer social networks analysis based on semantics. *Journal of Information Technology* (2010);**25**(2):178–88.
14. Schein A, Popescul A. Popescul L, Pennock D. Methods and metrics for cold-start recommendations. In: *The International ACM SIGIR Conference on Research and Development on Information Retrieval*. 2002, p. 253–60.
15. Blondel V, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* (2008);**1**(10):1742–54.
16. Fortunato S. Community detection in graphs. *Physics Reports* (2010);**486**(3):75–174.
17. Porter M, Onnela J, Mucha P. Communities in networks. *Notices of the AMS* (2009);**56**(9):1082–97.
18. Monge P, Contractor N. *Theories of Communication Networks*. Oxford University Press, USA; 2003.
19. Freeman L. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press; 2004.
20. Girvan M, Newman M. Community structure in social and biological networks. *National Academy of Sciences* (2002);**99**(12):7821–6.
21. Liu X, Bollen J, Nelson M, Van de Sompel H. Co-authorship networks in the digital library research community. *Information Processing and Management* (2005);**41**(6):1462–80.
22. Kottak C. *Cultural Anthropology: Appreciating Cultural Diversity*. McGraw-Hill; 2011.
23. Bullmore E, Bassett D. Brain graphs: Graphical models of the human brain connectome. *Annual Review of Clinical Psychology* (2011);**7**:113–40.
24. Du N, Wu B, Pei X, Wang B, Xu L. Community detection in large-scale social networks. In: *The International Workshop on Knowledge Discovery on the Web and International Workshop on Social Networks Analysis*. 2007, p. 16–25.
25. Tang L, Liu H. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery* (2010);**2**(1):1–137.
26. Papadopoulos S, Kompatsiaris Y, Vakali A, Spyridonos P. Community detection in social media. *Data Mining and Knowledge Discovery* (2012);**24**(3):515–54.
27. Traud A, Mucha P, Porter M. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications* (2011);**391**(16):4165–80.
28. Ferrara E. Community structure discovery in facebook. *International Journal of Social Network Mining* (2012);**1**(1):67–90.
29. Schaeffer S. Graph clustering. *Computer Science Review* (2007);**1**(1):27–64.
30. Aldecoa R, Marín I. Deciphering network community structure by surprise. *PLOS ONE* (2011);**6**(9):24–95.
31. Evans T. Clique graphs and overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment* (2010);**12**(1):1–39.
32. Arenas A, Duch J, Fernández A, Gómez S. Size reduction of complex networks preserving modularity. *New Journal of Physics* (2007);**9**(6):176–90.
33. Barber M. Modularity and community detection in bipartite networks. *Physical Review E* (2007);**76**(6):1–11.
34. Dellschaft K, Staab S. An epistemic dynamic model for tagging systems. In: *The ACM Conference on Hypertext and Hypermedia*. 2008, p. 71–80.
35. Begelman G, Keller P, Smadja F, et al. Automated tag clustering: Improving search and exploration in the tag space. In: *The Collaborative Web Tagging Workshop*. 2006, p. 15–33.
36. Simpson E. Clustering tags in enterprise and web folksonomies. In: *The International AAAI Conference on Weblogs and Social Media*. 2008, p. 222–3.
37. Papadopoulos S, Kompatsiaris Y, Vakali A. A graph-based clustering scheme for identifying related tags in folksonomies. *Data Warehousing and Knowledge Discovery* (2010);**6263**:65–76.
38. Fatemi M, Tokarchuk L. An empirical study on imdb and its communities based on the network of co-reviewers. In: *The Workshop on Measurement, Privacy, and Mobility*. 2012, p. 1–7.
39. Mislove A, Marcon M, Gummadi K, Druschel P, Bhattacharjee B. Measurement and analysis of online social networks. In: *The ACM SIGCOMM Conference on Internet Measurement*. 2007, p. 29–42.
40. Traud A, Kelsic E, Mucha P, Porter M. Community structure in online collegiate social networks. *American Physical Society* (2009);**88**:1–38.
41. Tsatsou D, Papadopoulos S, Kompatsiaris I, Davis P. *Online Multimedia Advertising: Techniques and Technologies*; chap. Distributed Technologies for Personalized Advertisement Delivery. Information Science Publishing; 2010, p. 233–261.
42. Pham M, Cao Y, Klamma R, Jarke M. A clustering approach

for collaborative filtering recommendation using social network analysis. *Journal of Universal Computer Science* (2011); **17**(4):583–604.
43. Zhao Q, Mitra P, Chen B. Temporal and information flow based event detection from social text streams. In: *The National Conference On Artificial Intelligence*; vol. 2. 2007, p. 1501–6.
44. Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Transactions on Communities in networks, Pattern Analysis and Machine Intelligence* (2000);**22**(8):888–905.
45. Silver D, Miller D. Contextualizing the artistic dividend. *Journal of Urban Affairs* (2012);.
46. Silver D. The american scenescape: amenities, scenes and the qualities of local life. *Cambridge Journal of Regions, Economy and Society* (2012);**5**(1):97–114.
47. Císař O, Koubek M. Include em all?: Culture, politics and a local hardcore/punk scene in the Czech Republic. *Poetics* (2012); **40**(1):1–21.
48. Silver D, Clark T, Rothfield L. A theory of scenes. *Manuscript, University of Chicago* (2007);Retrieved from http://tnc. research. googlepages. com/atheoryofscenes (accessed January 27, 2013).
49. Kozinets R. Technology/ideology: How ideological fields influence consumers technology narratives. *Journal of Consumer Research* (2008);**34**(6):865–81.
50. Kozinets R, Handelman J, Lee M. Dont read this; or, who cares what the hell anti-consumption is, anyways? *Consumption, Markets and Culture* (2010);**13**(3):225–33.
51. Samwald M, Cheung K. Experiences with the conversion of senselab databases to rdf/owl. *World Wide Web Consortium (W3C) Interest Group Note* 2008;Retrieved from http://www.w3.org/TR/hcls-senselab/ [Accessed October 2013].
52. Siorpaes K, Hepp M. myOntology: The marriage of ontology engineering and collective intelligence. *Bridging the Gap between Semantic Web and Web* (2007);**2**:127–38.
53. Villegas N, Müller H. *The SmarterContext Ontology and its Application to the Smart Internet: A Smarter Commerce Case Study*; vol. 7855; chap. The Personal Web. Lecture Notes in Computer Science; 2013, .
54. Knublauch H, Fergerson RW, Noy NF, Musen MA. *The Protégé OWL plugin: An open development environment for semantic web applications*; chap. The Semantic Web. Springer; 2004, p. 229–43.
55. Haarslev V, Müller R. Racer system description. In: *Automated Reasoning*. 2001, p. 701–5.
56. Tsarkov D, Horrocks I. FaCT++ description logic reasoner: System description. In: *Automated reasoning*. 2006, p. 292–7.
57. Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web* (2007);**5**(2):51–3.
58. Quilitz B, Leser U. *Querying distributed RDF data sources with SPARQL*; chap. The Semantic Web: Research and Applications. Springer; 2008, p. 524–38.
59. Bastian M, Heymann S, Jacomy M. Gephi: An open source software for exploring and manipulating networks. In: *The International AAAI Conference on Weblogs and Social Media*. 2009, p. 361–2.
60. Friedman J, Hastie T, Tibshirani R. *The Elements of. Statistical Learning: Data Mining, Inference, and Prediction*; vol. 1. Springer Series in Statistics; 2001.
61. Herring S. A faceted classification scheme for computer-mediated discourse. *Language at Internet* (2007);**4**(1):1–37.
62. Tunkelang D. Faceted search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* (2009);**1**(1):1–80.
63. Ma H. Google refine. *Technical Services Quarterly* (2012); **29**(3):242–3. http://code.google.com/p/google-refine/.
64. Philips L. Hanging on the metaphone. *Computer Language* (1990);**7**:9–43.
65. Lu S, Fu K. A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Transactions on Systems, Man and Cybernetics* (1978);**8**(5):381–9.
66. Fortunato S, Barthelemy M. Resolution limit in community detection. *National Academy of Sciences* (2007);**104**(1):36–41.
67. Clauset A, Newman ME, Moore C. Finding community structure in very large networks. *Physical review E* 2004;**70**(6):66–72.
68. Newman ME. Finding community structure in networks using the eigenvectors of matrices. *Physical review E* 2006;**74**(3):36–46.
69. Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 2007;**76**(3):036–46.
70. Rosvall M, Bergstrom CT. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 2008;**105**(4):1118–23.
71. Pons P, Latapy M. Computing communities in large networks using random walks. In: *Computer and Information Sciences-ISCIS 2005*. 2005, p. 284–93.

Appendix A. OWL Scene Ontology Graph



Appendix B. Scene Ontology Triples

Subject	Predicate	Object
http://www.sceneverse.com/CW/scene/	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Ontology
http://www.sceneverse.com/CW/scene/	http://www.w3.org/2000/01/rdf-schema#comment	"This ontology describes the cutural scene based on Seneverse Inc. defenition"
http://www.sceneverse.com/CW/scene/	http://www.w3.org/2002/07/owl#imports	http://purl.org/vocab/bio/0.1/
http://www.sceneverse.com/CW/scene/	http://www.w3.org/2002/07/owl#imports	http://www.isi.edu/~pan/damltime/time-entry.owl
http://www.sceneverse.com/CW/scene/	http://www.w3.org/2002/07/owl#imports	http://xmlns.com/foaf/0.1/
http://www.sceneverse.com/CW/scene/associatedWith	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/associatedWith	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#FunctionalProperty
http://www.sceneverse.com/CW/scene/associatedWith	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	http://www.w3.org/2002/07/owl#topObjectProperty
http://www.sceneverse.com/CW/scene/centralizedArround	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/centralizedArround	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	http://www.w3.org/2002/07/owl#topObjectProperty
http://www.sceneverse.com/CW/scene/describeScene	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/hasActivePeriod	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/hasActivePeriod	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#FunctionalProperty
http://www.sceneverse.com/CW/scene/hasActivePeriod	http://www.w3.org/2000/01/rdf-schema#domain	http://www.sceneverse.com/CW/scene/Scene
http://www.sceneverse.com/CW/scene/hasActivePeriod	http://www.w3.org/2000/01/rdf-schema#range	http://www.sceneverse.com/CW/scene/SceneActivePeriod
http://www.sceneverse.com/CW/scene/hasActivePeriod	http://www.w3.org/2002/07/owl#inverseOf	http://www.sceneverse.com/CW/scene/associatedWith
http://www.sceneverse.com/CW/scene/hasEvent	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/hasEvent	http://www.w3.org/2000/01/rdf-schema#domain	http://www.sceneverse.com/CW/scene/Scene
http://www.sceneverse.com/CW/scene/hasEvent	http://www.w3.org/2000/01/rdf-schema#range	http://www.sceneverse.com/CW/scene/SceneEvent
http://www.sceneverse.com/CW/scene/hasEvent	http://www.w3.org/2002/07/owl#inverseOf	http://www.sceneverse.com/CW/scene/sceneEvent
http://www.sceneverse.com/CW/scene/hasLocation	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/hasLocation	http://www.w3.org/2000/01/rdf-schema#domain	http://www.sceneverse.com/CW/scene/Scene
http://www.sceneverse.com/CW/scene/hasLocation	http://www.w3.org/2000/01/rdf-schema#range	http://www.sceneverse.com/CW/scene/SceneLocation
http://www.sceneverse.com/CW/scene/hasLocation	http://www.w3.org/2002/07/owl#inverseOf	http://www.sceneverse.com/CW/scene/centralizedArround
http://www.sceneverse.com/CW/scene/hasPeople	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/hasPeople	http://www.w3.org/2000/01/rdf-schema#domain	http://www.sceneverse.com/CW/scene/Scene
http://www.sceneverse.com/CW/scene/hasPeople	http://www.w3.org/2000/01/rdf-schema#range	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/hasPeople	http://www.w3.org/2002/07/owl#inverseOf	http://www.sceneverse.com/CW/scene/participateIn
http://www.sceneverse.com/CW/scene/hasTopic	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/hasTopic	http://www.w3.org/2000/01/rdf-schema#domain	http://www.sceneverse.com/CW/scene/Scene
http://www.sceneverse.com/CW/scene/hasTopic	http://www.w3.org/2000/01/rdf-schema#range	http://www.sceneverse.com/CW/scene/SceneTopic
http://www.sceneverse.com/CW/scene/hasTopic	http://www.w3.org/2002/07/owl#inverseOf	http://www.sceneverse.com/CW/scene/describeScene
http://www.sceneverse.com/CW/scene/participateIn	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/sceneEvent	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.sceneverse.com/CW/scene/sceneEvent	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	http://www.w3.org/2002/07/owl#topObjectProperty
http://www.w3.org/2002/07/owl#topObjectProperty	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
http://www.w3.org/2002/07/owl#topObjectProperty	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#TransitiveProperty
http://www.sceneverse.com/CW/scene/MultiPersonaScenester	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/MultiPersonaScenester	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/Personage	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/Personage	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/Scene	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/Scene	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/Scene
http://www.sceneverse.com/CW/scene/SceneActivePeriod	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/SceneActivePeriod	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.isi.edu/~pan/damltime/time-entry.owl#IntervalThing
http://www.sceneverse.com/CW/scene/SceneEvent	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/SceneEvent	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://linkedevents.org/ontology/Event
http://www.sceneverse.com/CW/scene/SceneEvent	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://purl.org/NET/c4dm/event.owl#Event
http://www.sceneverse.com/CW/scene/SceneEvent	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://purl.org/dc/dcmitype/Event
http://www.sceneverse.com/CW/scene/SceneEvent	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://sw.opencyc.org/2009/04/07/concept/en/Event
http://www.sceneverse.com/CW/scene/SceneFollower	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/SceneFollower	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/SceneLocation	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/SceneLocation	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing
http://www.sceneverse.com/CW/scene/SceneOrganizer	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/SceneOrganizer	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/ScenePerson	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/SceneTopic	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/SceneTopic	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/SceneTopic
http://www.sceneverse.com/CW/scene/Scenester	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/Scenester	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/ScenesterFriend	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/ScenesterFriend	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/SecludedScenester	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/SecludedScenester	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/Silhouette	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.sceneverse.com/CW/scene/Silhouette	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.sceneverse.com/CW/scene/ScenePerson
http://www.sceneverse.com/CW/scene/Sponsor	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
http://www.w3.org/2000/10/swap/pim/contact#Person	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://xmlns.com/foaf/0.1/Agent