Eighth International Workshop
Modelling and Reasoning in Context (MRC 2016)

Context plays an increasingly important role in modern IT applications. Context sensitivity and awareness is becoming essential, not only for mobile systems, ambient computing and the internet of things, but also for a wide range of other areas, such as learning and teaching solutions, collaborative software, web engineering and health care work-flow.

The Modelling and Reasoning in Context workshop series aims to bring together researchers and practitioners from different communities, both in industry and academia, to study, understand, and explore issues surrounding context. The workshop covers different understandings of what context is, different approaches to modelling context, mechanisms and techniques for structured storage of contextual information, effective ways to retrieve it, and methods for enabling integration of context and application knowledge.

The organizers would like to thank all the authors for submitting their papers and the members of the program committee for their valuable review contribution.

Workshop website
mrc.kriwi.de

Workshop chairs
Jörg Cassens, Rebekah Wegener, Anders Kofod-Petersen

Program Committee

Juan Augusto, Middlesex University, UK
Tarek Richard Besold, Free University of Bozen-Bolzano, Italy
Henning Christiansen, Roskilde University, Denmark
Adrian Clear, Newcastle University, UK
Bozidara Cvetkovic, Jožef Stefan Institute, Slovenia
Martin Christof Kindsmüller, University of Applied Sciences Brandenburg, Germany
David Leake, Indiana University, USA
Ana Gabriela Maguitman, Universidad Nacional del Sur, Argentina
Stella Neumann, RWTH Aachen University, Germany
Thomas Roth-Berghofer, University of West London, UK
Sven Schwarz, DFKI, Germany

List of Accepted Papers

Embedding a Creativity Support Tool within Computer Graphics Research

Yalemisew Abgaz[1], Diarmuid P. O'Donoghue[1], Donny Hurley[1], Horacio Saggion[2], Francesco Ronzano[2], Dmitry Smorodinnikov[1]

Abstract. We describe the Dr Inventor creativity support tool that aims to support and even enhance the creativity of active research scientists, by discovering un-noticed analogical similarities between publications. The tool combines text processing, lexical analysis and computational cognitive modeling to find comparisons with the greatest potential for a creative impact on the system users. A multi-year corpus of publications is used to drive the creativity of the system, with a central graph matching algorithm being adapted to identify the best analogy between any pair of papers. Dr Inventor has been developed for use by computer graphics researchers, with a particular focus on publications from the SIGGRAPH conference series and it uses this context in three main ways. Firstly, the pragmatic context of creativity support requires the identification of comparisons that are unlike pre-existing information. Secondly, the suggested inferences are assessed for quality within the context of a corpus of graphics publications. Finally, expert users from this discipline were asked to identify the qualities of greatest concern to them, which then guided the subsequent evaluation task.

1. INTRODUCTION

Creativity is a highly valued human ability, lying at the heart of many advances in scientific thinking and processes. Reasoning with the use of analogical comparisons [1] is a well-known explanation for many instances of scientific creativity and can also be a driver of scientific creativity [2]. Creativity support tools (CST) [3] aim to facilitate users in their efforts to produce some creative output. Dr Inventor [4] is a CST focused on creativity within scientific reasoning, helping in the creation of novel information that is useful to some scientific community.

We view the creative process as being composed of distinct sub-tasks, with Dr Inventor to perform some tasks while the user retains overall responsibility for the creative outcomes. Dr Inventor assumes responsibility for identifying high quality analogical comparisons between scientific publications (related to its application domain, computer graphics), based on a computational model [5] of the human ability of reasoning using analogies. Dr Inventor adopts a Big Data perspective towards creative inspiration, by exploiting the wide availability of academic documents for use as sources of inspiration for Dr Inventor's users. The user is then responsibility for ultimately evaluating and either using the presented analogy – or rejecting it as a false or fruitless comparison.

For example, many papers in computer graphics addressing the problem of cloth simulation use "thin plate equations" to simulate the look and behavior of clothes. But using these equations is based on an analogy between a piece of cloth and a thin metallic plate. The problem of modelling clothes is the *target/problem* while the metallic place is called the *source*. Even if such comparisons may seem obvious once they are presented, generating novel and useful analogies is a very difficult and challenging problem.

In this paper we present a novel combination of lexical and semantic processing with a computational analogy model, aimed at discovering novel and useful analogies between publications. Section two provides an overview of creativity and how it is supported by the process of thinking analogically. Section three describes the text processing pipeline and the subsequent generation of a semantic graph structure. Section four describes the core analogy model and its computational metrics. Section five then describes the document corpus and user studies that evaluated the effectiveness of the identified analogies.

2. ANALOGICAL COMPARISONS IN CREATIVE SCIENTIFIC REASONING

Creativity is a highly valued human ability and can be seen as a form of self-generated thought that produces new and useful knowledge, which makes subsequent reasoning more effective. We focus on creativity driven by bisociations [6] between disparate concepts, relying on the well-studied cognitive process of reasoning through the use of analogical comparisons.

Analogies pervade our understanding, particularly of complex or abstract concepts such as time [7]. Analogies involve comparisons between dissimilar objects, but the degree of semantic difference between the *source* and *target* analogs can vary greatly. A *target* from one area of computer graphics may be compared to a different area of computer graphics (often called "near analogies") or to politics or cooking ("far analogies"). Semantically far analogs have long been associated with more innovative and challenging comparisons. Notably, scientific revolutions [8] are strongly associated with these semantically distant comparisons.

[1] Department of Computer Science, Maynooth University, Co..Kildare, Ireland

[2] Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain.

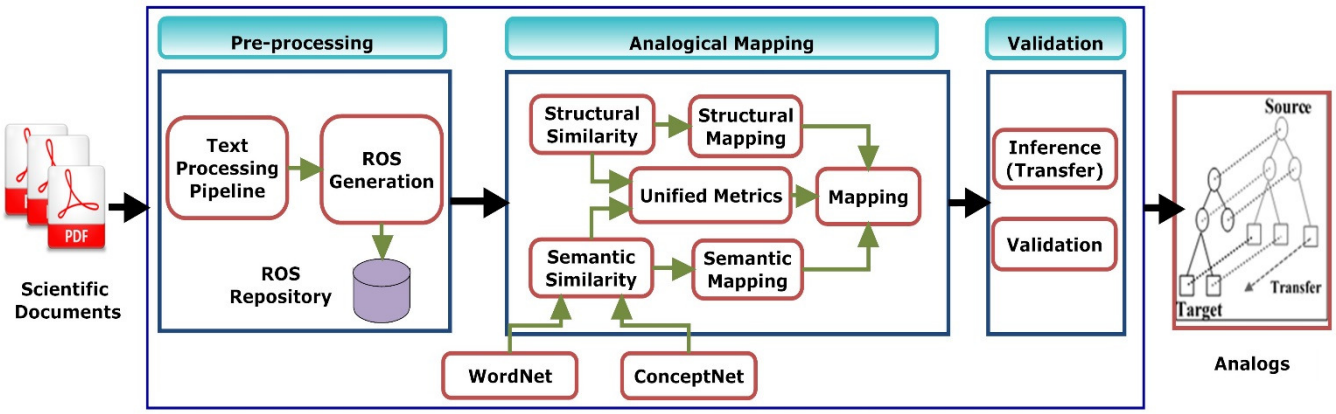

Figure 1. Dr. Inventor's Analogically Blended Creativity Framework

While Dr Inventor is not yet aiming at identifying creative analogies that might revolutionize some scientific discipline, it does hope to uncover latent analogies that might drive scientists' creativity. The role of analogies in scientific reasoning can be easily overlooked. A study of 16 one-hour meetings held across four different biological laboratories, identified the use of over 99 distinct analogies [9]. The majority of these analogies involved comparisons between semantically near items, such as comparisons between similar organisms or parts of organisms. This paper explores potentially creative "near" analogies between graphics publications.

[9] found that "far" analogies were often used to formulate a new hypothesis, using comparisons between an organism and (say) physics or even politics. Far analogies have also been shown to promote relational thinking [10], highlighting deep analogous similarity and overcoming any superficial similarities that may exist.

2.1. Computational Creativity

Computational creativity is a new discipline that aims to emulate human creativity, producing outputs that possess the central traits of creativity: *novelty* and *quality* (or *usefulness*) [11]. [12] demonstrated that a computational model of analogy is capable of generating many creative scientific analogies, but this work was limited by its reliance on hand-coded data. The approach adopted in this paper overcomes that limitation by sourcing all data directly from published documents, utilizing only machine-based processing of the original problem data. Dr Inventor forms and evaluates all of its analogical comparisons from the "raw" publications [13] using its novel combination of lexical and semantic processing.

2.2. Boosting Creativity with the Dr Inventor CST

We present the Dr Inventor CST (Figure 1) that aims to foster the creativity of practising scientists based on a cognitive computation model to simulate the generation of many analogies. From the results generated by our model, we choose the best analogical comparisons that offer a (potentially creative) interpretation of a given problem paper to ignite the scientist's creativity. Dr Inventor takes a descriptive computational model of the analogical reasoning process and uses it to predict those analogies that will have the greatest impact on its users' creativity.

The following factors are intended to help identify those analogies with the greatest creative potential:

- an extensive corpus with many candidate sources with which to re-interpret any given target problem
- metrics focused on identifying "good" analogies with creative potential
- persistence in exploring many analogies

Our CST addresses several of the challenges that are known to inhibit peoples' creativity:

- problem fixation and being entrenched in one view of a problem [14]
- memory limitations [15] and access to potentially useful information
- [16] showed that people do not notice analogies even when they are presented to them, but Dr Inventor can exhaustively explore all analogies [17] [18]

Additionally, our computational model enables us to quantify some metrics to help identify creative analogies by

- quantifying the level of pre-existing similarity between papers (using metrics based on the WordNet lexical database) and
- estimating the relative importance of pre-existing similarity and inferences for creative analogizing.

This paper explores the related challenges of developing and assessing the outputs of a CST within the specialized context of computer graphics research. We avail of experts in computer graphics to assist in this evaluation process. The major components of the tool are discussed in detail in Section 3 and 4.

2.3. Creativity in Computer Graphics Context

To ascertain the importance of creativity in the context of researchers in computer graphics, two surveys were undertaken. The first survey sought the opinions of practising researchers within this discipline as to the level of importance they placed on creativity when reviewing conference or journal papers. Respondents were asked for their opinion on the value they placed on creativity when reviewing papers. Three statements were rated by respondents:

1. Creativity is important when reviewing paper.
2. I can assess the level of creativity in a paper.
3. I can compare the levels of creativity between two papers

We believe the results shown in Figure 2 provide strong support for the importance of creativity in scientific research. Over 75% of respondents either "Strongly agreed" or "Agreed" that creativity is important when reviewing a paper. Additionally we infer that

creativity is important to the research underlying such publications. Around 80% of respondents said they are able to assess the level of creativity of a paper (presumably in part by detecting differences was previously read papers). Only the last question attracted a small level of disagreement, suggesting that comparing the level of creativity between two papers may sometimes be quite challenging.

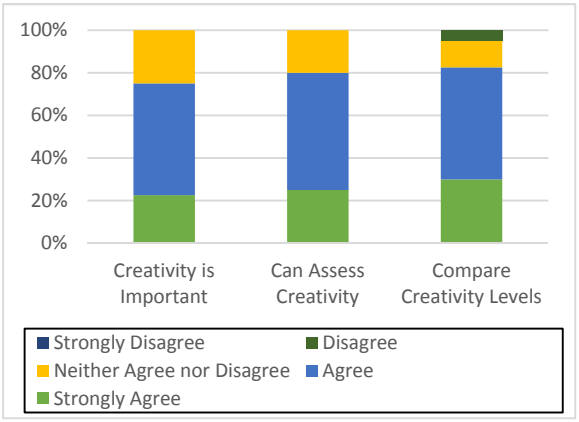

Figure 2: Do authors and reviewers of publications believe that creativity is important in a paper

Buoyed by this support for creativity within scientific research, we focused on specific metrics for use in evaluating the outputs of Dr Inventor. The SPECS standard [19] identified 14 independent components of general creativity, this encompassed creativity from diverse disciplines like the culinary arts, poetry, painting and architecture, with components like *emotion and self-expression* and *spontaneous and subconscious processing*. Thus, a survey was undertaken to identify the SPECS components of greatest relevance to scientific creativity and computer graphics researchers, with 34 researchers rating each quality on a 5-point Likert scale. The three qualities identified as most relevant to scientific creativity (by researchers in computer science) were as follows:

1 This is a novel or unexpected comparison (M=4.3, sd=0.73)

2 This comparison is potentially useful and recognizes gaps in current research (M=4.1, sd=0.83)

3 This comparison challenges the norms in this discipline. (M=3.8, sd=0.99)

Later, we shall see how these three qualities were used by respondents to evaluate the analogies developed by Dr Inventor.

3. SYNTACTIC AND SEMANTIC PROCESSING

3.1. Dr Inventor Text Mining Framework

The semantic analysis of the research articles and the extraction of *subject-verb-object* triples from the text of papers is supported by the Dr Inventor Framework [20] (DRI Framework), a pipeline of text-mining modules. The DRI Framework is distributed as a stand-alone Java library[2] that exposes an API to trigger the analysis of articles as well as to easily retrieve the results. In particular, the Framework defines a data model [21] of scientific publication properly structured to accommodate and conveniently expose the result of the analyses performed over a paper.

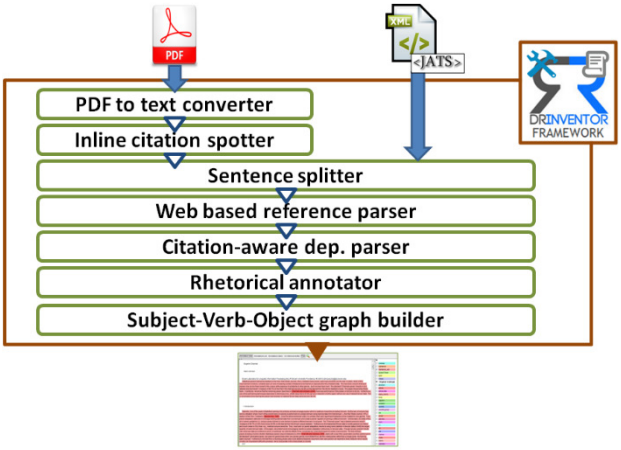

Figure 3: Architecture of the Dr Inventor Text Processing Framework

Figure 3 provides an overview of the core scientific text mining modules of the DRI Framework. Since most scientific publications are available in PDF format, the *PDF to text converter* processes PDF articles by invoking the PDFX online Web service[3] [22]: papers are converted into XML documents by identifying core structural elements including the title, the abstract, the hierarchy of sections and the bibliographic entries. This step can be by-passed if the article is available as *JATS*. Citations are identified by the *Inline citation spotter* relying on a set of high coverage regular expressions and heuristics. Sentence boundaries in the documents are identified by a *Sentence Splitter* specifically customized to the idiosyncrasies of scientific discourse. The bibliographic entries identified in the article are enriched by means of the *Web based reference parser* by accessing external Web services including Bibsonomy[4], CrossRef[5] and FreeCite[6]. In order to obtain syntactic dependencies between words in each sentence, a *Citation-aware dependency parser* builds the dependency tree of the sentences using [23] which we have customized so as to correctly deal with in-line citations. Since the rhetorical role of a sentence in a scientific document is important for information extraction and other scientific content analysis activities, a trainable logistic regression *Rhetorical classifier* was developed which assigns to each sentence of a paper a rhetorical category (i.e. Background, Approach, Challenge, Outcome and Future Work). The classifier is trained on the Dr Inventor Multi-layered Corpus[7] of Computer Graphics papers, manually annotated in the context of the Dr Inventor Project [18]. This corpus was used to train the classifier.

By relying on the output of the dependency parser, the *Subject-Verb-Object graph builder* extracts from the contents of a paper Subject-Verb-Object triples as shown in Figure 4. These triples constitute the core structure of the ROS graph that is mined in order to spot similar papers and analogies among the contents of publications.

Even if not explicitly shown in Figure 3, the Dr Inventor Framework also supports the generation of extractive summaries of

[2] The Dr. Inventor Text Mining Framework Java library can up the the IP be downloaded at: http://backingdata.org/dri/library/

[3] http://pdfx.cs.man.ac.uk/

[4] http://www.bibsonomy.org/help/doc/api.html

[5] http://search.crossref.org/help/api

[6] http://freecite.library.brown.edu/

[7] http://sempub.taln.upf.edu/dricorpus/

publications by implementing several approaches to select the most relevant sentences to be included in the summary [24] which can be used to select triples occurring in the most relevant parts of a document.

3.2. ROS-graph Generation

The analogy system does not work directly on the publications but instead uses a graph-centered representation based on the text extraction. These graphs are called Research Object Skeleton (ROS) graphs.

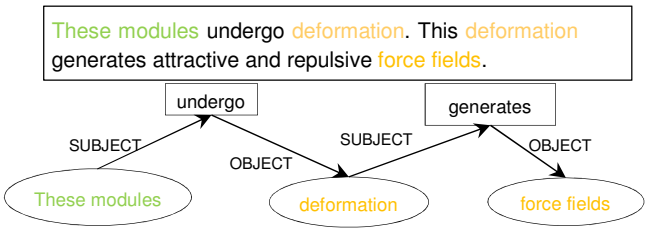

Figure 4: Subject-Verb-Object triples generated by the graph builder from two sentences

The ROS graphs have at the core of their structure the Noun-Verb-Noun type of relations (or Concept-Relation-Concept) enabling the application of Structure Mapping Theory [25] of analogy formation. While the core of the graph is the triple structure, the graph format chosen can have relationships between relations, i.e. second-order relations or causal relationships between nodes. These graphs are a form of *attributed relational graph* where nodes haves the attribute of "type" (i.e. noun, verb, causal). Among the additional attributes added to each node we consider the rhetorical category associated to the sentence in which the node occurs, extracted by means of the text processing pipeline and represented as an ontology-based semantic annotation [26]. This enables the creation of sub-graphs where analysis can be made on particular chosen categories of the publication. Dr Inventor relies on, for storage, the graph database Neo4j[8] which uses attributed relational graphs as its representation – making it highly suitable for our purposes.

The ROS is constructed by considering the dependency tree derived from each sentence in the publication. As in [27] a set of rules is applied to these trees, generating connected triples of nouns and verbs. One of the key properties of the ROS graphs is that multiple mentions of the same concept are uniquely represented. This is done either from the co-reference resolution of the text mining framework or by simply joining nodes that have the same word. Relation nodes, i.e. the verbs, can appear multiple times in the ROS. These constructed ROS graphs enable the steps of the analogy process and the mapping between different publications.

4. ANALOGY GENERATION AND ANALOGY METRICS

Analogy generation involves a mapping between the ROS of a selected target paper and the available source papers. The mapping pairs are then evaluated using a number of metrics and the best analogies are presented for evaluation by users.

[8] http://www.neo4j.com

4.1. ROS Mapping

Finding creative analogy requires exploration of many unsuccessful comparisons before discovering any useful analogy. Because of the high computational cost of performing retrieval, mapping and evaluation on a great many comparisons, computational efficiency was a primary concern – especially in the design of this central mapping phase.

Following Gentner's structure mapping theory [25], we generate the mapping between the source and target graphs. Our mapping involves structural mapping based on the graph structures and semantic mapping based on the semantics represented by the individual nodes and edges of the graphs. We also utilize mapping rules and constraints discussed in [28] distinctly incorporating both structural mapping and semantic aspects into the mapping process.

Generating the inter-ROS mapping is primarily driven by structure – that is, driven by any similarities between the topologies of the two ROS graphs. Thus, topology serves as a hard constraint on the space of possible mappings that is considered by Dr Inventor. However, when the structure of the two ROSs indicate multiple alternative solutions, we use semantic similarity to guide development of the preferred mapping. Thus, semantics are used as a soft constraint (or a preference constraint) on the mapping process, choosing between alternative mappings when different interpretations are available.

4.1.1. Structural Mapping

Our structural mapping is based on graph structure and conceptual structure. Graph structure focuses on identifying isomorphic graphs, while conceptual structure addresses the conceptual similarity between the nodes and edges that are to be paired by the mapping process [16, 29]. Specifically the objective of our structural mapping is to find the largest isomorphic subgraphs of a target paper in a source papers. For our specific purposes, let S be the set of all nodes in the source ROS graph $G_S=(S, E_S)$, let T be the set of all nodes in the target ROS graph $G_T= (T, E_T)$ and let $M = \{(S_i, T_i) | \ S_i \in S, T_i \in T, S_i$ is mapped to $T_i\}$ be the set of mappings between the source graph and the target graph. A mapping $M \subset S \times T$ is said to be an isomorphism iff M is a bijective function that preserves the branch structure of the graphs. And M is said to be the best analogical mapping if: 1) M is an isomorphism between a subgraph of G_T and subgraph of G_S, 2) M is the largest subgraph and, 3) M has the highest semantic similarity between its pairs.

We consider three constraints to guide structural mapping. The first constraint is defined on the types of nodes. A pair of nodes should have the same conceptual category to be a candidate of structural mapping. This means, "nouns" only map to "nouns" and "verbs" map only to "verbs". The second constraint is defined on the type of the edges. For two edges to be considered candidates, their corresponding nodes should satisfy the first constraint. We included the commutativity of relation (verb) nodes in a graph. If we consider a commutative relationship like (x adjacent y) and noting that this is equivalent to (y adjacent x), we allow such commutative relations to map more flexibly than non-commutative relations. The third constraint focuses on the degree of the mapping nodes. The degree of a candidate node of the source graph should be at least greater than the degree of the target node. This allows us to find isomorphic

subgraphs. In addition to these constraints, the traditional definition of structural mapping [25] holds true for this discussion.

Our structural mapping is implemented using a customized version of graph matching algorithm called VF2 [30]. The customization introduced the above constraints to preserve the properties of analogy mapping.

4.1.2. Semantic Mapping

Semantic mapping is an aspect of the mapping process that favours the generation of mappings that place a small cognitive workload on the Dr Inventor users – favouring semantically "simple" analogies whenever these are possible. This preference constraint is based on the similarity of the meaning of the words represented by each node in the ROS. Our semantic mapping utilizes the Lin similarity measure [31], which is based on WordNet [32], to calculate the similarity between source nodes and target nodes of similar type. These semantic similarity values are used during the computation and the selection phase of candidate pairs to be included in *M*. A pair with higher similarity score is selected and expanded first whenever we encounter two or more feasible candidate pairs. Thus, semantic mapping ensures a higher semantic similarity between the words represented by the mapping nodes of the isomorphic subgraph.

4.1.3. Lexico-Semantic Features

The text processing pipeline, ROS generation and analogy formation were largely developed as separate components, a number of features of each were aimed at maximizing the analogies that could be formed and their creative potential. The text processing pipeline and its dependency parser aimed to maximize the number of complete subject-verb-object triples, so that the rich and highly connected ROS graphs could be generated to form large rich mappings. The automated identification of the rhetorical category of sentences allows Dr Inventor to identify analogies between different parts of publications. This paper focuses on analogies formed between papers, each represented by its (lexical) "Abstract" and the rhetorical category of "Background".

We readily acknowledge that Dr Inventor does not have a deep understanding of the analogies it generates. Thus it could not be used to reliably create a new document from any of its discovered analogies for addition to its corpus. Therefore, it has not yet reached the level of being able to support the kind of *self-sustaining* computational creativity discussed in [33].

4.1.4. Inference and Validation

Inferences suggested by the analogy are modeled through the CWSG – Copy With Substitution and Generation [34] – a form of inference generation through of pattern completion. Dr Inventor ensures that all inferences are "grounded" in the mapping to ensure no spurious inferences are generated. While this paper explored analogies *only* between graphics publications and the resulting inference should (generally) be plausible combinations of source and target information, we report on some initial work aimed at validating inferences. Each inference is in the form of a triple (S V O), with each term arising in either the source or the target paper. A necessary step before evaluating Dr Inventor using publications outside the discipline of computer graphics, is to validate the inferences by detecting spurious combinations of S, V and O that may inadvertently arise.

Inference validation is one as the main mechanisms utilizing the graphics context and we explored several approaches to validating inferences. Firstly, inferences may be validated through comparison with existing triples in the Dr Inventor corpus by identifying a pre-existing instance in the Neo4j database. For less familiar triples an N-Gram model was developed to calculate the likelihood of combinations of S, V and O.

$$P(s,v,o) = P(s|<start>)\ P(v|s)\ P(o|v)\ P(<end>|o)$$

However, the N-Gram approach would be greatly hampered by zero probabilities arising from the novel (i.e. creative) combinations that Dr Inventor seeks. We explored additive smoothing [35], Good-Turing smoothing [36] and synonym substitution. Finding quality synonyms for the computer graphics context proved challenging an initial testing indicated that ConceptNet was not appropriate to validate graphics inferences. For this paper we focused on the WordsAPI provided by an online service[9].

4.2. Metrics

Once we generate the mappings between each source and target ROS, we further analyse the result to compute some metrics related to analogical similarity. This involves independent assessment of the semantic and structural factors involved in similarity. We then used a unified metric computed by multiplying structural similarity by semantic similarity. For measuring structural similarity we used Jaccard's coefficient [37]. The coefficient is used to measure the similarity between two finite sets, *A* and *B*. It is defined as:

$$J(A,B) = |A \cap B|/|A \cup B| = |A \cap B|/(|A| + |B| - |A \cap B|) \quad (1)$$

The Jaccard's coefficient gives a value of 1 if the A and B are structurally identical and yields 0 if there is no commonality between the two sets. Recall that $M = \{(S_i, T_i) |\ S_i \in S, T_i \in T, S_i$ is mapped to $T_i\}$. The Jaccard's coefficient for two graphs is then $J(S,T)$ where M is effectively $S \cap T$. Therefore, $J(S,T) = 0$, if there is no mapping between the two ROSs and $J(S,T) = 1$, if the two ROSs are structurally identical. Jaccard's coefficient gives a good estimation of *how much* of the two graphs have been mapped. For measuring semantic similarity between a pair of words, different approaches are suggested by research [38].

4.2.1. WordNet based metrics

The Lin metric returns value between 0 and 1 and has a readily accessible API. The overall semantic similarity of the mapping pairs is given by the average semantic similarity of the pairs in M, i.e.

$$\text{SemS}(M) = \frac{\sum_{i=1}^{m} \text{Lin}(S_i,T_i)}{m}, \quad (2)$$

where $m = |M|$ is the size of the mapping. Novel words not known within WordNet were not included in these calculations. A unified metric is computed as the product of the structural similarity and the semantic similarity. Unified Analogy similarity (AS) metrics is given as:

$$\text{AS}(S,T) = J(S,T) \times \text{SemS}(M) \quad (3)$$

To support the identification of analogous papers, we use the Lin metric to calculate independent levels of *relational similarity* – between mapped verbs and *conceptual similarity* between mapped

[9] http://www.mashape.com

nouns. This allows Dr Inventor to identify mappings with high relational similarity but low conceptual similarity, although there is no agreed definition of low and high.

An additional metric quantifies the number of inferences that are mandated by each analogical comparison, as modeled through a simple pattern-completion process based on the inter-ROS mapping. More inferences may indicate a comparison highlights something new about the target problem and we expect (at least) some of these inferences to be useful and meaningful if we adapt them from the source to the target paper.

5. EVALUATION OF GENERATED ANALOGIES BY EXPERTS

We present the setup of the experiment and evaluation results. To evaluate the performance of the system, we run our tool using a computer graphics collection of papers. Experts from computer graphics domain evaluated the results of the system. We ask the users to rate the analogs based on selected properties of creative systems identified by SPECS [19] and collect both quantitative and qualitative feedback. We present the results below.

5.1. Experimental Conditions

5.1.1. Datasets – computer graphics corpus

A corpus of computer graphics publications formed the basis for this evaluation, consisting of publications from the ACM Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH) conference – the top-ranked conference on computer graphics according to Microsoft Academic Search. The corpus contained 957 papers from the proceedings of SIGGRAPH between 2002 and 2011. Papers ranged from 6 to 12 double column pages. Each paper of the corpus was processed by the DRI Framework, thus identifying sentences together with their rhetorical category (challenge, background, approach, outcome, etc.). A typical ROS graph contains an average of 997 nodes (median=1013, mode=1041 and SD=±265).

Ten *target* papers were selected using a simple random sampling technique, with their titles being listed in Table 1. For the experiments reported in this paper we considered only the triples generated from the abstract and from its sentences classified as background (rhetorical category) of each paper. This reduced the burden on evaluators by allowing them to focus on a subset of the paper (highlighted by a customized pdf viewer). Second, this reduced the size of the graphs, greatly expediting the computational process of finding the largest mapping.

Dr Inventor was then used to generate all possible analogies for each target, using all 957 papers in the corpus as potential sources. From the resulting 957 analogical comparisons, the best source paper was selected for each target using the metric described in section 4.2.

5.1.2. Overview of Respondents

The outputs of the system were evaluated by 14 active researchers working in different areas of computer graphics. Their professional level includes postgraduate students (9), postdoctoral researchers (2), senior lecturers (2) and professors (1). The gender distribution is female (4) and male (10). The evaluation task was preceded by users watching a training video and the entire evaluation task was completed over two days. Postgraduate evaluators were compensated for their participation in this evaluation task.

5.1.3. Evaluation procedure

Before the evaluation, the respondents were presented with a short introductory video outlining analogy and analogy based comparisons. Then they were introduced to the Dr Inventor system and their evaluation task. Each analog pair of papers was presented and evaluated in turn. Users had access to the pdf version of the papers, including a highlighting of the sentences from the rhetorical "background" category. Users also were able to see the terms that had been placed in correspondence by the analogical mapping process, to help them better understand the presented analogy.

The system also allowed the users to browse the ROS graph thanks to an interactive visualization. The system further allowed users to navigate to/from the source and the target papers to the ROS visualization to find the original text where the mappings occurred. After spending sufficient time studying the analogs, users then gave their feedback on each analogous pair of papers.

5.2. Expert Ratings for the 10 good Analogies

The 14 researchers rated the 10 analogs, found by the Dr Inventor system, (No 1 to 10) for the 3 qualities discussed in Section 2.3 using a 5 point Likert scale [1-5]. While the number of respondents may appear small, each evaluation required reading two graphics publications and interaction with Dr Inventor system to explore the similarities using the visualization tools. 14 users evaluated 10 analogies each (reading 20 papers) with each analogy evaluation taking around 45 minutes. Thus our detailed evaluation represented around 110 person hours of work (or almost 14 8-hour work days).

Table 1. List of SIGGRAPH paper titles that formed the best analogies

No	Target Paper	Creative Source Paper
1	*Linear Combination of Transformations*	*Gaussian KD-Trees for Fast High-Dimensional Filtering*
2	*Curve Skeleton Extraction from Incomplete Point Cloud*	*Fast Bilateral Filtering for the Display of High-Dynamic-Range Images*
3	*Deforming Meshes that Split and Merge*	*Near-Regular Texture Analysis and Manipulation*
4	*Rotational Symmetry Field Design on Surfaces*	*Subdivision shading*
5	*3D Modeling with Silhouettes*	*Invertible Motion Blur in Video*
6	*Converting 3D Furniture Models to Fabricatable Parts and Connectors*	*Multi-Aperture Photography*
7	*Physical Reproduction of Materials with Specified Subsurface Scattering*	*Enrichment Textures for Detailed Cutting of Shells*
8	*Unstructured Video-Based Rendering: Interactive Exploration of Casually Captured Videos*	*Popup: Automatic Paper Architectures from 3D Models*
9	*Robust Treatment of Collisions, Contact and Friction for Cloth Animation*	*Inverse Shade Trees for Non-Parametric Material Representation and Editing*

10	*Real-Time Hand-Tracking with a Color Glove*	*Direct-to-Indirect Transfer for Cinematic Relighting*

Table 1 lists the titles of the source and target papers involved in each of the 10 analogies generated by Dr Inventor. Table 2 lists the computational metrics derived from each of these 10 analogical comparisons, grouped under the "Metrics" heading. Additionally, the average ratings awarded to each of these analogies under the three categories (novel useful and challenge) is also listed, grouped under the "Ratings" heading. The analogies in table 1 and also in table 2 have been ordered on descending values user ratings.

Table 2. Metrics and expert evaluations for the 10 generated analogies

	Metrics					Ratings				
Number	Rel Sim	Con Sim	M Ratio	Number of Inferences	Analogical Similarity	Novel	Useful	Challenge	Avg Rating	LSA
1	0.79	0.37	0.72	16	0.24	4.5	3.7	4.0	**4.07**	0.4
2	0.80	0.37	0.58	12	0.25	3.9	3.2	3.4	**3.48**	0.5
3	0.67	0.56	0.65	1	0.30	3.8	3.3	3.3	**3.44**	0.6
4	0.62	0.48	0.50	5	0.10	3.8	3.4	3.2	**3.44**	0.4
5	0.62	0.48	0.70	2	0.24	3.9	3.1	3.3	**3.43**	0.7
6	0.75	0.38	0.54	9	0.22	3.8	2.8	3.3	**3.30**	0.2
7	0.66	0.37	0.60	5	0.04	3.5	3.5	2.8	**3.28**	0.7
8	0.71	0.41	0.71	6	0.24	3.5	2.9	2.8	**3.08**	0.6
9	0.66	0.53	0.59	6	0.11	3.8	2.5	2.6	**2.97**	0.8
10	0.65	0.51	0.66	3	0.26	3.8	2.5	2.5	**2.92**	0.7

The top ranked analogy pair (No 1 in Table 1) has average user ratings of 4.46, 3.73 and 4.00 for the three qualities respectively and has an overall average of 4.06. The second ranked analogy pair (no 2) has a rating of 3.88, 3.05, and 3.33 with average rating of 3.42. However, the overall correlation between the analogical similarity and the user ratings is not strong. This leads to a further investigation of the proposed analogy metrics.

We do not expect all analogies generated by Dr Inventor to be rated highly for novelty, usefulness and challenging the norms. Figure 5 compares the ratings given to the best analogy with the average ratings awarded to all these analogies. The best analogy received higher than average ratings on each of the three qualities.

Looking particularly at the (computational) metrics for the top two analogies, an interesting pattern emerges. Firstly, these two analogies have the highest relational similarity (*RelSim* in Table 2) and the lowest conceptual similarity (*ConSim* in Table 2). These two qualities are the essential hallmarks of good analogical comparisons [1]. The larger *ConSim* scores indicate a difference in the nominals being discussed and are a strong indication that the anaology involves information arising from different research contexts – suggesting the source is document likely to be overlooked by a researcher. Additionally, these two analogies generated the largest number of inferences. A Pearson product-moment correlation of 0.608 was found between the number of inferences and the user ratings of each analogy, supporting importance of inferences to quality of analogies. Interestingly, the metrics for the two best comparisons displayed the classical hallmarks of good analogical comparisons is seen as strong support for both our approach and our computational model.

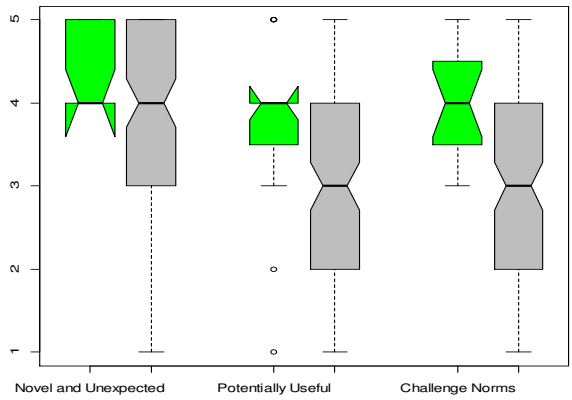

Figure 5. The ratings for the best and average analogies for each of the three qualities of creativity

We also highlight that Dr Inventor's finds similarities that are different to other techniques by comparison to Latent Semantic Analysis (LSA), which has been used in previous work on analogy identification [39]. The LSA model was set to make its comparisons in document-to-document mode, using the first 300 factors of the "General Reading up to 1st year college" training set, which was used as a loose reflection of the linguistic exposure of the respondents (the majority of whom were postgraduate students).

The final column in Table 2 illustrates the (LSA) [40] score between analogous papers, using the lexical Abstract with rhetorical Background of each paper. The Pearson product-moment correlation between the *analogy score* and the LSA score was 0.1948 indicating that Dr Inventor is identifying documents that are quite dissimilar to those identified by LSA (noting that the corpus used for these results concerned *only* publications from SIGGRAPH). Similarly, the Pearson product-moment correlation between the user ratings and the LSA score was -0.523 indicating that Dr Inventor's users and LSA are identifying very different types of similarity between documents.

5.3. Qualitative Feedback

As well as quantitative feedback, two senior professors further identified their favorite analogs from the 10 generated pairs. The first user favored analogy number 1 (Table 2). This comparison suggested interesting relations. The subtopics of the two papers (interaction versus image, photography animation and collision), their year of publication (2002 and 2009 respectively) and the problems the two papers tried to solve were surprisingly different. The technique adopted by the target paper could be used in the context of the source paper, suggesting that "manipulations applied to *filters* can be applied to *matrices* and vice versa "*leading to a few possible research questions*".

The second user favored analogy number 2 (Table 1). The target paper covers topics such as modeling and point cloud whereas the source focuses on topics such as image processing and photography. Here the target paper is published in 2009 whereas the source was published in 2002. The first paper addresses the problem of incomplete data during 3D laser scan, where the point cloud data

representing the object contains large *holes* where the laser did not scan. The second paper addresses the problem of poor management of light for under/over exposed *areas* in a photographs. The respondent found that the suggested mappings are useful to recognize the technique used in one could be used in the other regardless of the different problem areas the two papers tackle. One evaluator was particularly interested in the mappings between "*hole*" and "*area*" and also between "*region*" and "*window*" (see Table 3). This professor noted that these two terms are generally used very differently and that thinking of one as being like the other was highly unusual and thought-provoking - despite the fact that the WordNet metrics did not show them to be particularly different. This analogy suggested that techniques described in the source paper could be used to effectively solve the problem of the target paper. Based on this analogy, the user suggested new ideas such as the use of the technique in the source paper to reconstruct hidden information for missing video data, facial expression, motion capture, recovery of 3D scan, X-ray etc.

Table 3. Excerpts from the mapping of analogy 2.

Source Word	Target Word	Sim Score	Source Word	Target Word	Sim Score
use	Utilize	1	outlier	source	0
function	information	0.350	**area**	**hole**	**0.419**
domain	Key	0.342	relate	to_compute	0.505
use	Be	0.774	weight	mesh	0.458
do_address	to_handle	1	**window**	**region**	**0.390**

One unexpected result of the evaluation is that some users found inspirations from the target to the source - while we only expected users to gain inspirations from the source to the target. This positive, though unexpected, finding may be attributed to a number of causal factors. It may have arisen for users who are more familiar with the topic of the source paper, where the presented comparisons serves to overcome their problem fixation. It may be attributed to the (symmetric) visualizations that presented the source-to-target mapping or may be attributed to a number of other factors. Even if this specific situation triggers the need for further investigation, our system has a potential to identify such inspirations which could not be identified by human otherwise.

5.4. Inference Quality Evaluation

1000 inferences were generated and scored by the Additive Smoothing and Good-Turing methods. These scores were then used to categorize inference as High, Medium and Low, with the High category representing the best 20% of inferences, Low represents the bottom 20% and Medium are the remainder.

The top 20 inferences as scored by both techniques were collected, as were the weakest 20 inferences from both. Human ratings were then obtained for these inferences from 10 independent human raters, on a 5-point Likert scale (5 = Very good, 1 = very bad). Both methods showed a good ability to distinguish between good and bad inferences. The average score awarded to the High Category was Additive Smoothing (M=4.5) and Good-Turing (M=4.1), while for the Low category ratings were Additive Smoothing (M=2.2) and Good-Turing (M=2.0). As can be seen these techniques are more reliable at identifying good inferences than bad ones. Overall, additive smoothing seems to offer the best potential at helping Dr Inventor at managing inference quality.

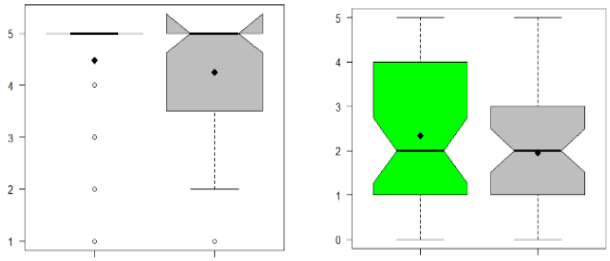

Figure 6. Scores awarded by Additive Smoothing (green) and Good-Turing (grey) to the inferences that people rated as good (left) and bad (right).

6. CONCLUSION AND FUTURE WORK

This paper described the Dr Inventor creativity support tool (CST) that aims to support scientific creativity by presenting novel analogical comparisons between publications. Firstly we presented the case for a CST based on the cognitive process of analogical thinking, describing how it might have a positive impact on the creativity of its scientist users.

We then described the major components of the Dr Inventor system. Dr Inventor is the first system to ever use "real" and automatically generated data from publications to simulate creative analogical thinking. It processes raw texts of scientific publications, generates graphs and analogically compares such graphs to identify analogies between documents. Based on the identified analogical similarity, Dr Inventor suggests inferences that can be transferred from the source for possible use in the target problem.

Thirdly, we presented an evaluation of the system to determine the level of creative support it provides to its users. We used the creative qualities of *novelty*, *usefulness* and *challenging the norms* to evaluate the level of inspiration and creativity support the system provides. The results indicated that Dr Inventor has a potential to identify novel and useful analogs. User ratings, of the analogies between pairs of papers identified by Dr Inventor, were provided by active researchers from computer graphics, using a 5 point Likert scale, with this feedback showing that the two highest rated comparison had many of the hallmarks of a good analogical comparison: high relational similarity, low conceptual similarity and a large number of inferences. The qualitative analysis indicates that Dr Inventor is capable of producing quality analogies and that these comparisons have a very beneficial impact on the creativity of the expert evaluators from the discipline of computer graphics.

Our future work will include co-references and causality to enhance the text analysis and in effect to improve the analogy mapping process. Another area of future work will focus on the metrics. Even if it is difficult to measure cognitive process, some preliminary results (relational and conceptual similarity) show that the correlation between users rating and the systems ranking could be improved by further enhancement of the metrics. Another future work that emerges from this research is the potential of creating a conceptual blend by merging analogical mappings of various papers.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme ([FP7/2007-2013]) under grant agreement no 611383.

REFERENCES

[1] D. Gentner and L. Smith, "Analogical reasoning. In V. S. Ramachandran (Ed.)," in *Encyclopedia of Human Behavior (2nd Ed.).* , Oxford, UK, Elsevier., 2012, pp. 130-136.

[2] T. Brown, Making Truth: Metaphor in Science, University of Illinois Press, 2003.

[3] B. Shneiderman, "Creativity support tools," *Communications of the ACM,* vol. 45, no. 10, 2002.

[4] D. O'Donoghue, Y. Abgaz, D. Hurley, F. Ronzano and H. Saggion, "Stimulating and Simulating Creativity with Dr Inventor," in *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*, Park City, Utah, 2015.

[5] D. Gentner and K. D. Forbus, "Computational Models of Analogy," *Wiley Interdisciplinary Reviews: Cognitive Science,* pp. 2(3):266-276, 2011.

[6] A. Koestler, The act of creation, 1964.

[7] O. Fuhrman and L. Boroditsky, "Cross-Cultural Differences in Mental Representations of Time," *Cognitive Science,* vol. 34, no. 8, pp. 1430-1451, 2010.

[8] T. S. Kuhn, The structure of scientific revolutions, 1962.

[9] K. Dunbar and I. Blanchette, "The in vivo/in vitro approach to cognition: The case of analogy," *Trends in cognitive sciences,* vol. 5, no. 8, pp. 334-339, 2001.

[10] M. Vendetti, A. Wu and K. Holyoak, "Far-out thinking generating solutions to distant analogies promotes relational thinking," *Psychological science,* vol. 25, no. 4, pp. 928-933., 2014.

[11] M. Boden, The Creative Mind, 2004.

[12] D. O'Donoghue and M. Keane, "A Creative Analogy Machine: Results and Challenges," in *4th International Conference on Computational Creativity*, Dublin, Ireland, 2012.

[13] F. Ronzano and H. Saggion, "Dr. Inventor Framework: Extracting Structured Information from Scientific Publications," in *Discovery Science*, Springer, 2015, pp. 209-220.

[14] B. C. Storm and G. Angello, "Overcoming fixation creative problem solving and retrieval-induced forgetting," *Psychological Science ,* 2010.

[15] M. T. Keane, T. Ledgeway and S. Duff, "Constraints on analogical mapping: A comparison of three models," *Cognitive Science,* vol. Cognitive Science, no. 18, pp. 387-438, 1994.

[16] M. Gick and K. Holyoak, "Analogical problem solving," *Cognitive psychology,* vol. 12, no. 3, pp. 306-355, 1980.

[17] V. Chaudhri, S. Heymans, A. Overholtzer, A. Spaulding and M. Wessel, "Large-Scale Analogical Reasoning," in *Proceedings of the Twenty-Eighth {AAAI} Conference on Artificial Intelligence*, Que'bec City, Canada, 2014.

[18] B. Fisas, R. Francesco and S. Horacio, "On the Discoursive Structure of Computer Graphics Research Papers," in *The 9th Linguistic Annotation Workshop held in conjuncion with NAACL 2015.* , 2015.

[19] A. Jordanous, "A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative," *Cognitive Computation,* vol. 4, no. 3, pp. 246-279, 2012.

[20] R. Francesco and S. Horacio, "Knowledge Extraction and Modeling from Scientific Publications," in *In the Proceedings of the Workshop "Semantics, Analytics, Visualisation: Enhancing Scholarly Data" co-located with the 25th International World Wide Web Conference*, Montreal, Canada, 2016.

[21] C. Hamish, T. Valentin, R. Angus and B. Kalina., "Getting more out of biomedical documents with GATE's full lifecycle open source text analytics," *PLoS Comput Biol,* vol. 9, no. 2, pp. 1-16, 2013.

[22] A. Constantin, P. Steve and V. Andrei, "PDFX: fully-automated PDF-to-XML conversion of scientific literature," in *Proceedings of the 2013 ACM symposium on Document engineering.* , 2013.

[23] B. Bohnet, "Very high accuracy and fast dependency parsing is not a contradiction," in *Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 2010*, 2010.

[24] H. Saggion, "SUMMA: A robust and adaptable summarization tool," *Traitement Automatique des Langues,* vol. 49, no. 2, 2008.

[25] D. Gentner, "Structure-mapping: A theoretical framework for analogy," *Cognitive Science,* vol. 7, pp. 155-170, 1983.

[26] A. Ruiz-Iniesta and O. Corcho, "A review of ontologies for describing scholarly and scientific documents.," in *Proceedings of 4th Workshop on Semantic Publishing (SePublica 2014)*, Aachen, Germany, 2014.

[27] B. Agarwal, S. Poria, N. Mittal, A. Gelbukh and A. Hussain, "Concept-level sentiment analysis with dependency-based semantic parsing: a novel approach," *Cognitive Computation,* vol. 7, no. 4, pp. 487-499, 2015.

[28] K. J. Holyoak and P. Thagard, "Analogical mapping by constraint satisfaction," *Cognitive Science,* vol. 13, pp. 295-355, 1989.

[29] D. Gentner and A. B. Markman, "Defining structural similarity," *Journal of Cognitive Science,* vol. 6, pp. 1-20, 2005.

[30] L. Cordella, P. Foggia, C. Sansone and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," in *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2004.

[31] D. Lin, "An Information-Theoretic Definition of Similarity," San Francisco, CA, USA, 1998.

[32] G. A. Miller, "WordNet: A Lexical Database for English.," *Communications of the ACM,* vol. 38, no. 11, pp. 39-41, 1995.

[33] D. O'Donoghue, J. Power, S. O'Briain, F. Dong, A. Mooney, D. Hurley, Y. Abgaz and C. Markham, "Can a Computationally Creative System Create Itself? Creative Artefacts and Creative Processes," in *nternational Conference on Computational Creativity (ICCC)*, Ljubljana, Slovenia, 2014.

[34] K. J. Holyoak, L. R. Novick and E. R. Melz, "Component processes in analogical transfer: Mapping, pattern completion, and adaptation," in *Analogical connections. Advances in connectionist and neural computation theory*, vol. 2, Westport, CT, US, Ablex Publishing, 1994, pp. 113-180.

[35] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," in *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, Santa Cruz, California, 1996.

[36] W. A. Gale, "Good-Turing smoothing without tears," *Journal of Quantitative Linguistics,* vol. 2, no. 3, pp. 217-237, 1995.

[37] P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines," *Bulletin de la Société Vaudoise des Sciences Naturelles,* vol. 37, pp. 241-272, 1901.

[38] L. Meng, R. Huang and J. Gu, "A Review of Semantic Similarity Measures in WordNet," *International Journal of Hybrid Informatioin Technology,* vol. 6, no. 1, 2013.

[39] M. Ramscar and Y. Daniel, "Semantic grounding in models of analogy: an environmental approach," *Cognitive Science,* vol. 27, no. 1, pp. 41-71, 2003.

[40] T. K. Landauer, P. W. Foltz and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes,* vol. 25, pp. 259-284, 1998.

Quality of Context Optimization in Opportunistic Sensing for the Automatization of Sensor Selection over the Internet of Things

Elif Eryilmaz[1] and **Sahin Albayrak**[1]

Abstract. The Internet of Things (IoT) will cover billions of intelligent objects being able to sense, act and communicate with each other. Opportunistic sensing makes use of the IoT by dynamically selecting sensors to derive a piece of information. However, sensors in the IoT differ from each other regarding the quality of data they can provide and existing approaches usually use a simplified metric to optimize the quality of context recognition. In this work, we aim to provide an overview of ongoing research to enable quality of context optimization by an autonomous sensor selection amongst available sensors over the IoT. The evaluation criterion is finding the best fitting sensor combination by means of quality and updating autonomously in case of any change.

1 INTRODUCTION

Classical context-recognition approaches often rely on the deployment of specific sensors for a specific context recognition goal [1]. These approaches are optimized to the sensors selected at design time. For any new recognition goal, they require the deployment of new sensors which can be expensive due to the necessity for additional hardware. Redesigning software applications to work with different sensors increases configuration effort and requires expert knowledge. Therefore, research in context-aware computing is moving towards the development of dynamic methods to utilize available sensors [2]. One of the approaches following this idea is opportunistic sensing, which aims to make use of sensors available via the IoT [3]. Opportunistic sensing uses signal and information processing techniques to enable the involved sensing infrastructures to automatically discover and select sensors or sensor combinations [4].

The IoT provides transparent access to many sensors, processors and actuators using standardized protocols via its underlying infrastructure without considering hardware, operating systems, or locations [5]. The variety and number of sensors available via the IoT offers a good basis for the implementation of an opportunistic sensing approach. However, not all sensors are relevant to the detection of a specific piece of context information and some sensors lead to a better Quality of Context recognition (QoC) than others. Thus, the main challenge is to find the right sensors providing the desired quality data for a specific context recognition task. As detailed in Section 4, in the current opportunistic sensing approaches, this problem is addressed by assuming fixed quality requirements of recognition goals at design time. As proposed by Kim and Lee [6], context information can be defined by many quality aspects, such as accuracy, precision, completeness, access security, and up-to-dateness. However, in context-aware systems, not all context recognition goals require the same QoC. Quality requirements of recognition goals are various and depend on the specification of the recognition goals. The requirements of the recognition goals can also change dynamically depending on the environmental constraints as well as sensors availability. To illustrate this with an example, we consider the user biking in the forest carrying a smartphone (GPS, 3G, accelerometer sensors) with him. Although GPS can provide much more accurate data for position than other sensors, it consumes much power. Therefore, when the battery of the phone becomes less than 20%, the user may want to use a method of position detection with less power consumption even though it is less accurate. In this case, the idea is to automatically substitute one sensor with another to be able to provide the position information (recognition goal) of the user. Additionally, the user can simply loose the GPS signal while going through the forest and not be able to get position information anymore by using GPS. In this case, the quality requirement of the recognition goal needs to be adjusted dynamically to use the available sensors to provide data even with less accuracy.

In this paper, we provided an overview of our ongoing work for QoC optimization in opportunistic sensing to enable autonomous sensor ensemble selection over IoT. Here, sensor ensemble refers to recognition chain to fulfil the recognition goal under certain quality requirements. Recognition chain consists of sensors and processing functions to derive the required information for the recognition goal. The remaining part of this paper is structured as follows. We present a use-case scenario as a motivation and running example for this paper in Section 2. We point out research challenges in Section 3 by providing the insights from the use-case scenario. Section 4 covers the state of the art research addressing the identified challenges and the gaps in the existing studies. Section 5 explains the overview of the approach to achieve the research contributions of the overall work. In Section 6, final remarks and future work are provided as a conclusion.

2 USE-CASE SCENARIO

In this scenario, we aim to motivate the importance of a context-aware sensor selection which can make use of existing sensors based on user requirements for the recognition goal by means of quality and energy consumption.

[1] Distributed Artificial Intelligence Laboratory (DAI-Labor), Technical University of Berlin (TU-Berlin), 10587 Berlin, Germany, email: {elif.eryilmaz, sahin.albayrak}@dai-labor.de

We consider the user having a smartphone (available sensors: GPS, 3G, phone inertial sensors like accelerometer and gyroscope) while biking outside. For our context-aware application, in normal conditions, the application looks for the most accurate sensor (lowest mean error) to get data about the users' speed and calories burned. The recognition goal of the application is the speed and calorie information. Amongst the available sensors on the phone for this purpose, GPS provides the most accurate data for user position and implicitly for speed. The recognition chain in this case is the sensor and data processing function to derive speed information from position. While going through the forest, the user's phone could lose the GPS signal permanently or intermittently which can have a negative impact on the calculated speed. To keep providing data to the application, there needs to be a mechanism for automatically searching amongst available sensors including less accurate data sources. When the GPS signal is available again with a higher quality, the mechanisms shall be able to reuse GPS to provide the most accurate result to the application. However, in many cases, using the most accurate sensor has a disadvantage as it consumes too much battery of the phone. To regulate this for better balance between accuracy and energy consumption, when the battery is lower than specific threshold, the mechanisms should look for the most energy efficient sensor (lowest energy coefficient) to provide the required data for the application even with less accuracy. This implies the change in the quality requirements of the recognition goal. Summarizing, the following situations requiring changes of the recognition chain in the scenario:

- Sensor appearance/disappearance (GPS becomes available and lost)
- Change in quality of sensors providing data (while going through forest, GPS is getting less accurate)
- Change in quality requirements of recognition goals (decrease accuracy of the data to be retrieved due to energy constraint)

If those changes are not known at design time or change dynamically, there is a need for adaptive mechanisms that select and update the recognition chain.

3 RESEARCH CHALLENGES

In this section, we identified four main challenges that need to be addressed to react to the changes mentioned in Section 2.

First, the **characteristics of each sensor** providing data need to be available. The sensor characteristics include, among others, the type of data the sensor provides, how frequent the sensor updates this data, the accuracy and reliability level, the cost of using this sensor, and the constraints/limitations (e.g. battery level, physical position, scope of provided information). The challenge here is the generic representation of sensor data and characteristics in a way that the sensors and the data they provide become comparable. As a result, they can be substituted with each other. An analogous problem has been part of the Semantic Web area, where the right services to provide some required information or functionality are required [7]. Turning back to the use-case scenario, available sensors on the smartphone should be modelled to include those characteristics.

After having a common level of sensor representation, the second challenge is modelling the **quality properties suitable for the recognition goal**. At this point, there are many different quality aspects (e.g., cost, accuracy, latency, stability) that could cause the preference of one sensor over another, even if they provide the same type and format of data. Recognition goals can be seen as goal-functions in optimization approaches, e.g., a weighted sum of QoC properties. Additionally, modelling of recognition goals requires deriving a degree of fulfilment of the recognition goal and acceptable variation from this fulfilment [8]. Therefore, the techniques addressing this problem should provide the results based on the priorities for the fulfilment by using comparison techniques in combination with sensor characteristics and recognition goals. In the use-case scenario, the detected type of information is speed/calorie information. The quality requirements concern the accuracy and energy consumption of the recognition.

The third challenge is finding mechanisms for **deriving the optimal recognition chain**. This requires different data fusion methods based on the type of available sensors and their impact on the recognition goal. For instance, in the use case scenario the recognition chain based on GPS requires calculating the speed from positions with time stamps, while the recognition chain for the accelerometer is concerned with deriving it from the acceleration values. The impact of the processing functions should be modelled at design time.

The last challenge is **adapting the recognition chain** in case of any change occurring at run time. These changes could concern the sensor infrastructure (availability/unavailability of sensors), the quality characteristics (sensor degradation) or the quality requirements of the recognition goal. A dynamic update of the selected recognition chain requires autonomous control in a way that the recognition chains are constantly checked and adapted to the changing conditions [9]. In the use-case scenario, this is related to the ability to use accelerometer instead of GPS for deriving speed information in case the GPS sensor is unusable, provides lower quality information or uses too much energy.

4 RELATED WORK

To address the mentioned challenges in Section 3, we conducted a state of the art research to present the existing work and the gaps that need to be addressed.

4.1 Characteristics of sensor

In order to have a generic representation of sensor data and characteristics, semantic technologies are commonly used to create universal description models for sensor data. For this purpose, there is some existing research for ontology based sensor description and data modelling for IoT solutions [10, 11]. The main approach aiming to capture the characteristics of a sensor accurately is the metadata annotation for sensor characteristics [12]. Although there is much effort on defining sensor meta-information, the description of observations measured by sensors has not been addressed much. The W3C Incubator Group released the Semantic Sensor Network XG Final Report, which defines the SSN ontology [13] for describing sensors, including their characteristics. The SSN ontology focuses on providing a domain independent ontology which is generic enough to adapt to different use-cases at the sensor and observation levels. The W3C working

group uses SSN ontology as a basis and extended it by addressing semantic interoperability to provide the ability to communicate between different entities without any ambiguity [14]. Current work on SSN ontology is promising but does not yet fully capture a comprehensive list of quality properties for sensors.

4.2 Quality properties of recognition goal

Roggen et al. propose the Opportunity Framework [15] as a reference implementation of an opportunistic sensing system for human activity and context recognition [16]. For a given recognition goal the Opportunity Framework configures the recognition chain dynamically based on the available sensors and domain knowledge. To deal with the quality requirements of the recognition goal, the authors [3] used a numeric value metric as a "degree of fulfilment". Although this can simplify the process of finding the best recognition chain by means of quality, it does not take into account other metrics like the frequency of detection or the costs of detection. In the research conducted by Villalonga et al., the authors tried to match quality parameters used to assess quality of context in general and mathematical parameters from wearable activity recognition systems by defining conversion functions [17]. However, it is not clear how to find extracted QoC parameters for further extensions from this paper or any other reference provided by the authors.

4.3 Deriving the optimal recognition chain

As mentioned in Section 3, recognition goals can be seen as goal-functions in optimization approaches. To address the challenge of finding the optimal recognition chain, modelling the impact of data fusion methods on the recognition goal should take place as a next step. Modelling this impact can be done by applying traditional optimization methods based on a goal function. Multi-objective optimization is one of the methods to address this challenge as there are multiple measures from different resources to decide the best satisfying solution [18].

4.4 Adapting the recognition chain

To provide autonomous control for selecting or updating the recognition chain dynamically, the mechanisms in opportunistic sensing should adapt themselves to the changing conditions particularly the availability of sensors and the quality requirements of context recognition goals. In the research area of self-adaptive software (SAS) systems, several architectures and models have been proposed to implement self-adaptation. One of the approaches proposed by IBM is to define adaptive behaviour for autonomic computing as feedback loops similar to control theory [19]. This idea provides a conceptualization of the reasoning process to decide whether an adaptation is required or not. Adaptation via a controlling feedback loop can be done explicit or implicit in the design of self-adaptive software systems. Making such feedback loops explicit from the system design is also proposed by several authors [20, 21, 22] as leads to a clear separation of concerns between the adapted system and the adaptation mechanisms and can provide standardized components for the adaptation that can be reused by other self-adaptive software systems. As we already discussed and presented in [23], the application of feedback loops to opportunistic sensing makes the required adaptability more explicit and extendable for autonomous control of selecting/updating recognition chain.

5 PROPOSED APPROACH

In this section, we provide an overview of our approach to enable quality optimization for opportunistic sensing. The structure of this section is based on the challenges presented in Section 3. The first two challenges concern representation of sensor characteristics and recognition goal quality requirements. Since they are very similar in nature they are handled jointly in Subsection 5.1. Subsection 5.2 provided the overview about how to find an optimal recognition chain. In Subsection 5.3, the details about how the recognition chain can be adapted autonomously are provided.

5.1 Addressing the Challenges 1&2

To represent sensor characteristics and quality requirements of recognition goal, we propose to model the recognition goal as a tuple of a quality metric, the data type to be detected from sensors and additional contextual parameters. The quality metric consists of different QoC characteristics derived from Quality of Service (QoS) and Quality of Device (QoD) properties. QoC defines the quality of the detected information (e.g. precision, freshness), where QoS defines the quality of the service that provides this information (e.g. error rate, availability) and QoD is the information about the technical properties of the devices that collects the data (e.g. location, battery life) [26]. The quality metric will be a goal function as a weighted sum of those QoC characteristics. The goal function also includes the percentage of fulfilment required by the recognition goal and the acceptable variation from this fulfilment [8]. Therefore, in case of dynamic changes in QoC properties, the ranking of parameters weighted in the goal function should change.

Regarding the data type to be detected from sensors and additional contextual parameters as the remaining variables for modelling the recognition goal, we will use ontologies. To model the sensor characteristics, the SSN sensor ontology will be used as it consists of general information about sensors, the values they measure and measurement capabilities of the sensors [13]. To model the context, we will use CONON (CONtext ONtology) to define the context entities like location, person or activity [24]. As context can affect QoC in quality metric together with the sensor characteristics, context and sensor ontologies should be aligned for the recognition goal. The details about the reasons to choose CONON, how context and sensor ontologies can be merged are provided in our other publication [25].

5.2 Addressing the Challenge 3

To address the challenge about finding an optimal recognition chain, we simplified the problem to an optimization problem. Optimization refers to a selection of a best element with regard to specific criterion from a set of available alternatives [27]. The aim of the optimization is finding a recognition chain fulfilling the quality requirements of the recognition goal as well as the search for the best fitting one.

The impact of the sensors together with the processing functions required to retrieve data defines the search space for the optimization. The optimization criteria is the goal function

mentioned in Subsection 5.1 which consists of the quality metric and is dependent on the other two parameters in the recognition goal tuple namely sensors and the context of use.

To solve this optimization problem by applying traditional AI techniques to the goal function, we will test different optimization algorithms. Those include the algorithms based on finite number of steps, or iterative methods that converge to a solution or heuristics that may provide approximate solutions [28]. We will define and formularize the required algorithms for testing after defining the goal function.

5.3 Addressing the Challenge 4

As mentioned in Section 4.4, one approach for the self-adaptation in software systems derived from control theory is implementing adaptive behaviour as explicit feedback loops. IBM proposed the MAPE-K feedback loop in the scope of autonomous computing for this purpose [19]. In this approach an autonomic manager (cf. Fig. 1) is responsible for the adaptation which ***M**onitors* information about the software system and its context of use, ***A**nalyses* this information, ***P**lans* changes according to the result of the analysis and ***E**xecutes* these changes. A central ***K**nowledge* base is responsible for storing and passing information between those phases of the feedback loop. To interact with the software system, the autonomic manager uses sensors to receive information about the software system and effectors to implement the required changes for the adaptation.

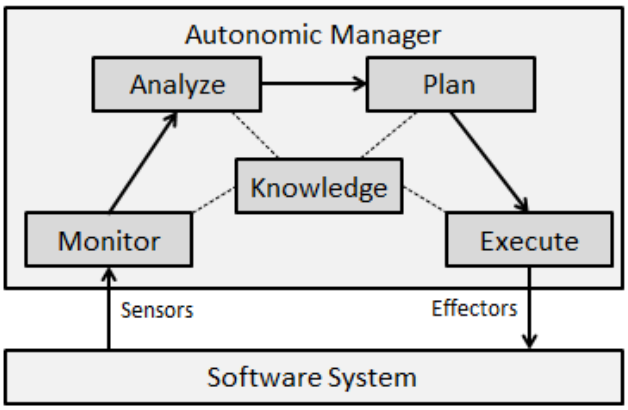

Figure 1. The MAPE-K feedback loop (based on [19]).

As mentioned in Section 4.4., having explicit feedback loops provides extendibility by developing standardized components for each phases of the feedback loop. One of the approaches for this purpose has been proposed by Vogel et.al. to use model-driven techniques for a feedback loop to increase the level of automation for executing a loop [29]. Vogel and Giese proposed to model MAPE-K components directly in a run time model and to refine the knowledge used by all adaptation activities in a feedback loop [30].

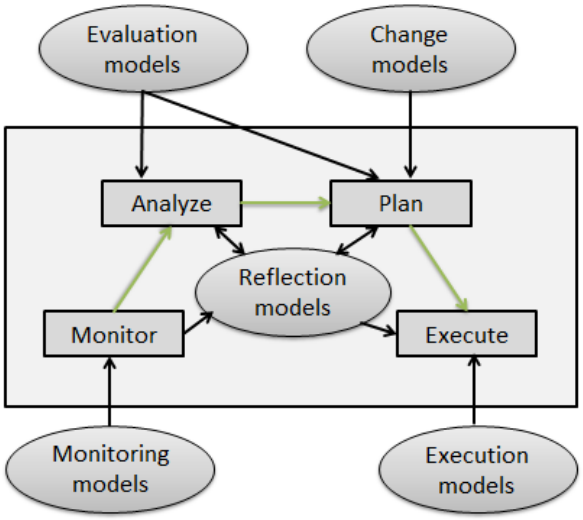

Figure 2. Run-time models based on [29, 30]

As depicted in Figure 2, *Reflection Models* reflect the software and its environment; *Monitoring Models* are responsible to map observations to the reflection models; *Evaluation Models* describes the goal towards to the required adaptations; *Change Models* define the options that are available for planning which can be used to find an appropriate adaptation with the guidance of evaluation models; and *Execution Models* describe to execute the planned adaptation by refining from the model to the adaptable software [30].

By following this model-driven approach for the feedback loops, we will make use of three models as depicted in Figure 3, namely the Evaluation Model, the Change Model and the Reflection Model. For this purpose, the evaluation model covers the modelling of the recognition goal. This includes a tuple of a quality metric, the data type to be detected from sensors and additional contextual parameters as described in Subsection 5.1. This model will be used as a goal for the planning of recognition chains. The change model consists of the descriptions of sensors and processing functions as well as annotations of related quality attributes. It also covers how the context represented by using the availability of sensors. Here, semantic technologies will be used to describe the sensors (SSN) and context (CONON). This model affects the planning phase of the feedback loop to be executed by defining the elements which are available during the construction of the recognition chain. The reflection model covers the representation of connections between sensors and processing functions. It aims to serve as knowledge between the phases of the feedback loop. It will be used to decide where and how to annotate the monitored information.

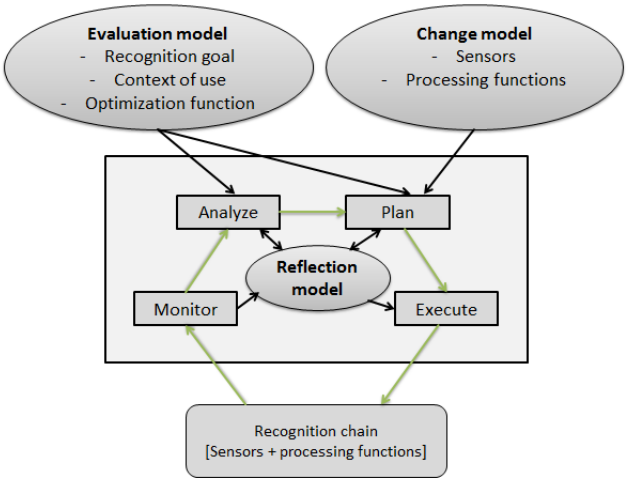

Figure 3. Proposed architecture by using run-time models

To map the models to each phases of the feedback loop (green arrows) in our approach for self-adaptability the monitoring phase is responsible for retrieving the information from deployed components (sensor + processing function) with the quality properties in case of changes (component failure). In order to get information from deployed components, we will use transfer learning. The analysis phase observes changes and decides whether a re-planning is required. Changes can concern the evaluation model (goal function, recognition goal, context of use) the elements in the change model (sensors and processing functions as well as their property) and the reflection models (observations made in deployed recognition chains). The information should be filtered to decide which components are worth to update to find a new recognition chain. This can be dependent on quality of the components as well as context of use and their connection between. Another purpose of the plan phase is to find the optimal recognition chain using the optimization function. This depends on the different quality parameters of components in relation with the context of use. After the planning phase, the execution phase is responsible for realizing the plan by deploying/un-deploying components. This can include start/stop/register of sensors and processing functions and their services.

6 CONCLUSION & FUTURE WORK

In this paper, we provided an overview of ongoing research to enable quality optimization for the opportunistic sensing. We presented our motivational use-case scenario and related research challenges. Based on the gaps in the existing work we give an overview of our approach to achieve an adaptive opportunistic sensing system that is able to optimize the recognition chain for different and changing quality requirements.

As a future work, the approach presented in Section 5 will be fully implemented. The approach will be evaluated to measure the improvement on the quality of context recognition by an autonomous sensor selection that selects a recognition chain over IoT best fitting the requirements of the recognition goal. Regarding the evaluation of the approach, a case study will be defined with specific quality requirements for the recognition goals to enable QoC optimization in opportunistic sensing. The overall approach will be compared to derive the improvement on the current quality results of the existing frameworks.

REFERENCES

[1] D. Roggen, G. Tröster, P. Lukowicz, et al., *Opportunistic Human Activity and Context Recognition*, IEEE Computer 46(2), 2013, pp. 36-45.

[2] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, *Context aware computing for the Internet of Things: A survey*, IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 414–454, 2014.

[3] M. Kurz, G. Hoelzl, A. Ferscha, et al., *The OPPORTUNITY Framework and Data Processing Ecosystem for Opportunistic Activity and Context Recognition*, International Journal of Sensors, Wireless Communications and Control, vol. 1(2), 2011, pp. 102-125.

[4] D. Chen, *Opportunistic Sensing in Wireless Sensor Networks: Theory and Application*, IEEE Transactions on Computers, vol. 63(8), 2014, pp. 2002-2010.

[5] M. Presser, P. M. Barnaghi, M. Eurich, and C. Villalonga, *The SENSEI Project: Integrating the Physical World with the Digital World of the Network of the Future*, IEEE Comm. Magazine, vol. 47, no. 4, 2009, pp. 1-4.

[6] Y. Kim, and K. Lee, *A quality measurement method of context information in ubiquitous environments*, ICHIT'06, IEEE Computer Society, 2006, 576–58.

[7] A. Brogi, S. Corfini, J.F.A. Montes, I.N. Delgado, *A prototype for discovering compositions of semantic web services*, in Tummarello, G., Bouquet, P., and Signore, O., editors, SWAP, volume 201 of CEUR Workshop Proceedings, 2006, CEUR-WS.org.

[8] OpenIoT Consortium, *Open source solution for the internet of things into the cloud*, January 2012, http://www.openiot.eu [Accessed on: 2016-07-22].

[9] IBM Corporation, *An Architectural Blueprint for Autonomic Computing. Technical report*, IBM Corporation, 2006.

[10] J.P. Calbimonte, Z. Yan, H. Jeung, et al., *Deriving Semantic Sensor Metadata from Raw Measurements*, 5th International Workshop on Semantic Sensor Networks (SNN), in conjunction with the 11th International Semantic Web Conference (ISWC), November 2012.

[11] M. Compton, C.A. Henson, L. Lefort, et al., *A Survey of the Semantic Specification of Sensors*, CEUR Workshop Proceedings, October 2009.

[12] P. Barnaghi, S. Meissner, M. Presser, and K. Moessner, *Sense and sens'ability: Semantic data modelling for sensor networks*, in Cunningham, P. and Cunningham, M., editors, ICT-Mobile Summit 2009 Conference Proceedings, pages 1–9. IIMC International Information Management Corporation.

[13] P. Barnaghi, M. Compton, O. Corcho, et. al., *Semantic sensor network XG final report: W3c incubator group report*, June 2011, http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/ [Accessed on: 2016-07-22].

[14] W3 Incubator Group, *Review of Sensor and Observations Ontologies*, http://www.w3.org/2005/Incubator/ssn/wiki/Review_of_Sensor_and_Observations_Ontologies [Accessed on: 2016-07-22].

[15] D. Roggen, K. Förster, A. Calatroni, et al., *OPPORTUNITY: towards opportunistic activity and context recognition systems*, 3rd IEEE WoWMoM Workshop on Autononomic and Opportunistic Communications, 2009, pp. 1-6.

[16] G. Hölzl, M. Kurz, A. Ferscha, D. Roggen, A. Calatroni, G. Tröster, R. Chavarriaga, J. Millàn, H. Sagha, P. Lukowicz, D. Bannach, *A Framework for Opportunistic Context and Activity Recognition*. Adjunct proceedings to the 9th International Conference on Pervasive Computing, 2011.

[17] C. Villalonga, D. Roggen, C. Lombriser, P. Zappi, and G. Tröster, *Bringing quality of context into wearable human activity recognition*

systems, First International Workshop on Quality of Context (QuaCon), 2009.

[18] D.F. Jones, M. Tamiz, *Goal programming in the period 1990-2000*, in Multiple Criteria Optimization: State of the art annotated bibliographic surveys, 2002, 129-170.

[19] J.O. Kephart, and D.M. Chess, *The vision of autonomic computing*, IEEE Computer, vol. 36(1), 2003, pp. 41–50.

[20] H. Giese, Y. Brun, J.D.M. Serugendo, C. Gacek, H. Kienle, H. Müller, M. Pezzè, and M. Shaw, *Engineering Self-Adaptive and Self-Managing Systems*, LNCS vol. 5527, Springer, 2009, pp. 47–69.

[21] J.L. Hellerstein, Y. Diao, S. Parekh, and D.M. Tilbury, *Feedback Control of Computing Systems*, John Wiley & Sons, 2004.

[22] H.A. Müller, H.M. Kienle, and U. Stege, *Autonomic Computing Now You See It, Now You Don't—Design and Evolution of Autonomic Software Systems*, ISSSE 2006-2008. LNCS vol. 5413, Springer, 2009, pp. 32–54.

[23] E. Eryilmaz, F. Trollmann, and S. Albayrak, *Conceptual Application of the MAPE-K Feedback Loop to Opportunistic Sensing*, in: 10th Workshop: Sensor Data Fusion: Trends, Solutions, Applications (SDF 2015), DOI: 10.1109/SDF.2015.7347697, IEEE, 2015, pp. 1-6.

[24] X.H. Wang, D.Q. Zhang, T. Gu, H.K. Pung, *Ontology based context modeling and reasoning using OWL*, in: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004, pp. 18–22.

[25] A. Boytsov, A. Zaslavsky, E. Eryilmaz, and S. Albayrak, *Situation Awareness Meets Ontologies: A Context Spaces Case Study*, in: The Ninth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'15); 2015.

[26] T. Buchholz, A. Küpper, and M. Schiffers, *Quality of Context: What It Is And Why We Need It*, in Proceedings of the workshop of the HP OpenView University Association, 2003.

[27] *The Nature of Mathematical Programming*, Mathematical Programming Glossary, INFORMS Computing Society, http://glossary.computing.society.informs.org/ver2/mpgwiki/index.php?title=Extra:Mathematical_programming [Accessed on: 2016-07-22].

[28] S. Koziel, and X. S. Yang, *Computational Optimization, Methods and Algorithms*, Springer, (2011)

[29] T. Vogel, A. Seibel, and H. Giese, *The role of models and megamodels at runtime*, in Models in Software Engineering. Lecture Notes in Computer Science, vol. 6627, Springer, 224–238, 2011.

[30] T. Vogel, and H. Giese, *Model-Driven Engineering of Self-Adaptive Software with EUREMA*, ACM Transactions on Autonomous and Adaptive Systems vol. 8(4), 2014, Article 18.

Finding Diverse High-Quality Plans for Hypothesis Generation

Shirin Sohrabi and **Anton V. Riabov** and **Octavian Udrea** and **Oktie Hassanzadeh** [1]

Abstract.

New applications that use AI planning to generate explanations and hypotheses have given rise to a new class of planning problems, requiring finding multiple alternative plans while minimizing the cost of those plans. Hypotheses or explanations about a system, such as a monitored network host that could be infected by malware, are generated as candidate plans given a planning problem definition describing the sequence of observations and a domain model capturing the possible state transitions for the modeled system, as well as the many-to-many correspondence between the states and the observations. The plans must minimize both the penalties for unexplained observations and the cost of state transitions. Additionally, among those candidate plans, a small number of the most diverse plans must be selected as representatives for further analysis. To this end, we have developed a planner that first efficiently solves the "top-k" cost-optimal planning problem to find k best plans, followed by clustering to produce diverse plans as cluster representatives. Experiments set in hypothesis generation domains show that the top-k planning problem can be solved in time comparable to cost-optimal planning using Fast-Downward. We further empirically evaluate multiple clustering algorithms and similarity measures, and characterize the tradeoffs in choosing parameters and similarity measures.

1 Introduction

In recent work a new class of AI planning formulations has been developed for solving practical problems in plan recognition, diagnosis of discrete event systems, and explanation generation (e.g., [17, 21, 22]). In these problems, each valid plan can be interpreted as a hypothesis meeting the constraints of the planing task, and providing a possible diagnosis or an explanation.

In prior work, these problems have been studied in satisficing or optimal planning settings. More recently, however, Sohrabi et al. [25] have shown that in malware detection applications, where observations can be noisy or the domain model can be imperfect, finding multiple near-optimal plans makes a significant difference in discovering ground truth scenarios, and therefore improves the overall utility of generated explanations.

Consider the following example of an application where finding multiple low-cost plans is desirable:

Example *In automated malware detection in computer networks, the goal is to provide assistance to network administrators in detecting and predicting behaviors of malware or computer viruses. Observations come from network traffic, but they are unreliable. That is, they can be noisy, incomplete, or ambiguous (indicative of multiple underlying causes). Moreover, the model description may be incomplete. Hence, it may not be possible to explain all observations and some observations may need to be discarded. However, we can help the administrators by providing top alternative hypotheses for further investigation. For example, given an ambiguous observation that could be both a result of normal activity or malware infection, we can present at least two clusters, one that includes the normal activity, and one that includes the possibility of infection. The infection cluster itself can be a result of multiple causes, but we may want to show only one representative per cluster at first, allowing the user to request the remaining hypotheses from the clusters they are interested in. This diverse set of plans will not include unlikely or low-plausibility hypotheses. Hence, it is required to find a set of plausible hypotheses and then group these in some meaningful way before presenting the results. These plans (or equivalently, hypotheses) can be further evaluated automatically, by collecting and analyzing additional data.*

The malware detection problem or more generally the hypothesis generation problem can be encoded as an AI planning problem [25, 19], where the generated plans correspond to the hypotheses, and furthermore, the min-cost plans correspond to the plausible hypotheses. Plausible hypotheses are those that the domain expert believes to be more plausible compared to the other hypotheses. Plausibility can be encoded as action cost, where higher costs indicate lower plausibility. Hence, the notion of the top-k plans maps to finding k plans with the lowest cost.

Computing a set of low-cost plans or the top-k plans has the following benefits:

1. one can find plans that satisfy constraints that are not known apriori or are not easy to formalize;
2. by providing a list of alternative plans, one can explore the space of alternatives and hence gain better understanding of the properties of the problem and its optimal solution; and
3. in the hypothesis generation problem, finding the set of top plans is necessary to find the most accurate hypothesis, especially when the observations are not reliable and the model is incomplete.

Furthermore, grouping the top plans or the top-k plans into clusters adds the following benefits:

1. it helps users quickly navigate through the alternatives via cluster hierarchies,
2. the automated system, if in place, can also benefit from exploring cluster representatives rather than all plans.

In this paper, we propose an approach for finding a set of low-cost diverse plans for hypothesis generation. To this end, we propose

[1] IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, email: {ssohrab, riabov, udrea, hassanzadeh}@us.ibm.com

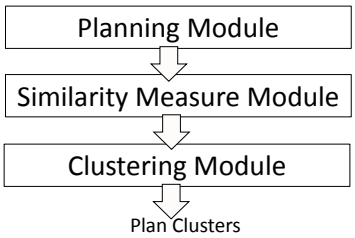

Figure 1. Framework

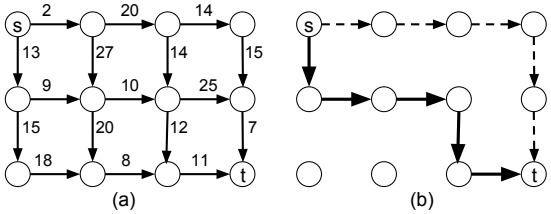

Figure 2. (a) shows a graph with source node s and terminal node t with edge lengths specified on the edges; (b) shows the shortest path in bold arrows and the second shortest path in dashed arrows.

a modular framework, as shown in Figure 1, where we first find a bounded set of low-cost plans, which we refer to as top-k plans, with respect to the given cost metric. We then cluster these plans based on their similarity and present the diverse plans by picking the representative plans from each cluster. The framework allows the use of different planning algorithms, similarity measures, and clustering algorithms in different combinations.

The *planning module* takes as input the planning problem with costs and produces a set of low cost plans. To solve the top-k planning problem, the problem of finding a set of k distinct plans with lowest cost, we propose use of a k-shortest paths algorithm. In particular, we developed an approach that allows us to solve the top-k planning problem by efficiently translating it into a k shortest path problem, and then solving that problem using the K^* algorithm [1]. We call the resulting top-k planner *TK**. Although K^* was developed for the k shortest paths problem, and has not been previously used in AI planning, it is efficient enough to be used in hypothesis generation problems of practical size, as experiments show.

The *similarity measure* module takes as input a pair of plans and decides whether the two are similar, by computing a similarity score and applying a threshold. Multiple similarity measures can be used in combination, and we evaluate a variety of domain-independent and domain-dependent measures.

Finally, the *clustering module* works with the result of the similarity measure module to produce the plan clusters. We evaluate the generated clusters for a set of hypothesis generation problem instances using several domain-independent and domain-dependent evaluation criteria including performance, number of clusters, and plan diversity. We also compare the performance and quality of solutions produced by our top-k planning framework and diverse planners.

The contributions of this paper are:

1. the decomposition of the problem of finding diverse high-quality plans into top-k planning and clustering stages, with configurable similarity measures;
2. a new top-k planner, *TK**, that applies K^* to planning problems;
3. efficient clustering algorithms for forming a set of diverse plans from a larger set of high quality plans; and
4. the evaluation of solution quality and performance of individual stages and overall framework on both manually crafted and random hypothesis generation problems and comparison to existing diverse planners. We find that our approach performs comparably to diverse planners in planning time and diversity, while finding solutions with consistently lower cost.

In what follows, we present the algorithms we use for finding the top-k plans, then we describe several relevant approaches that can be used for computing plan similarity, followed by an introduction of several single-pass algorithms for clustering plans. The experimental evaluation section includes separate subsections for top-k planning and plan clustering, as well as the comparison of the overall framework to diverse planners. We conclude with a discussion of related work and outline new opportunities for future research.

2 Top-k Planning

In this section, we will first formally define the top-k planning problem and then give the necessary background on the k shortest paths problem. We will then describe our top-k planning algorithm, TK*, that uses the K^* algorithm.

Definition 1 *We define the top-k planning problem as* $R = (F, A, I, \mathcal{G}, k)$*, where* F *is a finite set of fluent symbols,* A *is a set of actions with non-negative costs,* I *is a clause over* F *defining the initial state,* $\mathcal{G}$ *is a clause over* F *defining the goal state, and* k *is the number of plans to find. Let* $R' = (F, A, I, \mathcal{G})$ *be the cost optimal planning problem with* n *valid plans. The set of plans* $\Pi = \{\alpha_1, ..., \alpha_m\}$*, where* $m = k$ *if* $k \leq n$*,* $m = n$ *otherwise, is the solution to* R *if an only if each* $\alpha_i \in \Pi$ *is a plan for the cost-optimal planning problem* R' *and there does not exist a plan* α' *for* R'*,* $\alpha' \notin \Pi$*, and a plan* $\alpha_i \in \Pi$ *such that* $cost(\alpha') < cost(\alpha_i)$ *.*

When $k > n$, Π contains all n valid plans, otherwise it contains k plans. Π can contain both optimal plans and sub-optimal plans, and for each plan in Π all valid plans of lower cost are in Π. If $\Pi \neq \emptyset$, it contains at least one optimal plan.

Note, while we indicated that the goal state, $\mathcal{G}$, is in a form of a final-state goal in the definition of R, our approach can be applied to temporally extended goals as well. Temporally extended goals, such as a sequence of observations of a system, either totally ordered or partially ordered, can be compiled away to a final-state goal following a compilation technique discussed in several papers (e.g., [21, 9]).

2.1 Background: K Shortest Path Problem

K shortest paths problem is an extension of the shortest path problem where in addition to finding one shortest path, we need to find a set of paths that represent the k shortest paths [12]. The following is a formal definition taken from Eppstein [6].

Definition 2 (K Shortest Path Problem) *k shortest path problem is defined as 4-tuple $Q = (G, s, t, k)$, where $G = (V, E)$ is a graph with a finite set of n nodes (or vertices) V and a finite set of m edges E, s is the source node, t is the destination node, and k is the number of shortest paths to find. Each edge $e \in E$ has a* length *(or* weight *or*

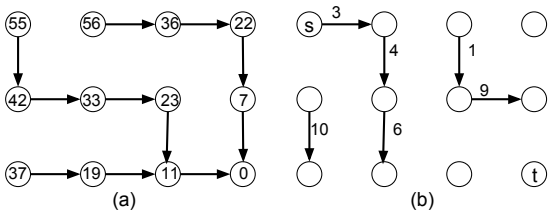

Figure 3. (a) shows the shortest path tree T and distance to destination t; (b) shows the side edges with their associated detour cost.

cost*), which is denoted by* $l(e)$*. The length of a path* p*,* $l(p)$*, is consequently defined by the sum of its edge lengths. The* distance $d(u, v)$ *for any pair of nodes* u *and* $v \in V$ *is the length of the shortest path between the two nodes. Hence,* $d(s, t)$ *is the length of the shortest path for the problem* Q*. Let* n *= size of the set of all s-t paths in graph* G*. Then, the set of paths* $P = \{p_1, p_2, ..., p_m\}$*,* $m = k$ *if* $k \leq n$*,* $m = n$ *otherwise, is the solution to the* k *shortest paths problem* Q *if and only if each* $p_i \in P$*, is a s-t path in graph* G *and there does not exist a s-t path* p' *in graph* G*,* $p' \notin P$ *and a path* $p_i \in P$ *such that* $l(p') < l(p_i)$ *.*

Note that if $k > n$, then P contains all s-t paths, otherwise P contains k shortest paths from node s to node t. It follows from the definition that at least one shortest path with length $d(s, t)$ is in the set P if $m > 0$.

Figure 2 shows an example from Eppstein [6] illustrating the terminology. The distance $d(s, t) = 55$, is the length of the shortest path shown in bold; the length of the second shortest path is 58.

2.2 Top-k Planning Using K^*

The K^* algorithm [1] is an improved variant of the Eppstein's k shortest paths algorithm [6] and hence uses many of the same concepts as in the Eppstein's algorithm (which we refer to as EA). Here, we first outline the EA algorithm, and then discuss K^*.

Given a k shortest paths problem $Q = (G, s, t, k)$, the EA algorithm first computes a single-destination shortest path tree with t as the destination (or the reversed single-source shortest path tree) by applying Dijkstra's algorithm on G. The edges in the explored *shortest path* tree T are called *tree* edges while all the missing edges (i.e., the edges in $G - T$) are called *sidetrack* edges. Each edge in G is assigned a number that measures the detour cost of taking that edge. Consequently, the detour cost of the tree edges is 0, while the detour cost of the sidetrack edges is greater than 0. Figure 3 shows the shortest path tree T and the sidetrack edges along with their detour cost of our earlier example.

The EA algorithm then constructs a complex data structure called *path graph* $P(G)$ that stores the all paths in G, where each node in $P(G)$ represents a sidetrack edge. This is followed by the use of Dijkstra search in $P(G)$ to extract the k shortest paths. An important property is that given a sequence of sidetrack edges representing a path in $P(G)$ and the shortest path tree T, it is possible to uniquely construct a s-t path in G. This can be done by using sub-paths from T to connect the endpoints of sidetrack edges.

Given this property and the special structure of $P(G)$, it is ensured that the i-th shortest path in $P(G)$ results in a sidetrack sequence which can be mapped to the i-th shortest path in G. By construction, $P(G)$ provides a heap-ordered enumeration of all paths

```
0. Read planning problem R = (F, A, I, G, k).
1. Expand the state graph G by using A*
   and applying actions to compatible states
   starting from I, and until G is reached.
2. Continue applying A* to expand G
   until 20% increase in links or nodes.
3. Update P(G) based on new links in G.
4. Apply Dijkstra step
   to extract the next path from P(G).
5. If k paths are found
6.      Goto step 10.
7. If K* scheduling condition is reached
8.      Goto step 2.
9. Goto step 4.
10. Return at most k plans (one plan per path).
```

Figure 4. *TK** planning algorithm applies K^* to search in planning state space.

in G, and since every node of $P(G)$ has limited out-degree (at most 4), the complexity of enumerating paths in increasing cost order is bounded. The worst-case runtime complexity of the EA algorithm is $O(m + n \log n + kn)$. This complexity bound depends on a compact representation of the resulting k paths, and can be exceeded if the paths are written by enumerating edges. For more details see [6].

The major bottleneck of the EA algorithm is the construction of the complete state transition graph, which may include a huge number of states that are very far away from the goal. Planners commonly deal with this challenge by relying on heuristic search algorithms like A* to dynamically expand only the necessary portion of the state graph during search, while being guided by a heuristic toward the goal (e.g., Fast-Downward [11]). The K^* algorithm combines the best of both worlds: it allows constructing the graph G dynamically using heuristic-guided A* search, while updating its equivalent of $P(G)$ to find k shortest paths.

In short, the K^* algorithm works as follows. The first step is to apply a forward A* search to construct a portion of graph G. The second step is suspending A* search, updating $P(G)$ similarly to EA, to include nodes and sidetracks discovered by A*, applying Dijkstra to $P(G)$ to extract solution paths, and resuming the A* search. The use of A* search to dynamically expand G enables the use of heuristic search and also allows extraction of the solution paths before G is fully explored. While K^* algorithm has the same worst-case complexity as the EA algorithm, it has better performance in practice because unlike the EA algorithm, K^* does not require the graph G to be completely defined when the search starts.

Our planner, *TK**, applies K^* to search in state space, with dynamic grounding of actions, similarly to how Fast-Downward and other planners apply A^*, following the algorithm above.

The K^* scheduling condition is evaluated by comparing the state of A* and Dijkstra searches, as defined in K^* algorithm. It determines whether new links must be added to G before resuming Dijkstra search on updated $P(G)$. There is no separate grounding stage, since actions are ground at the same time when they are applied during A* search. The amount of A* expansion required before resuming Dijkstra (in our implementation, 20%) controls the efficiency tradeoff, and 20% is the same value that was used in experiments in the original K^* paper [1]. Of course, step 2 may also terminate if no new links can be added.

Soundness and completeness of *TK** follows directly from the soundness and completeness of the K^* algorithm.

In our experiments, *TK** with constant 0 heuristic performs very well, and we have not experimented with other, potentially better performing heuristics. This is an interesting direction for improvement that could be explored in future work. Even though this is not a requirement for K^* in general, our implementation requires a consistent heuristic, which did not allow us to experiment with, for example, lookahead heuristics.

3 Finding Diverse Plans via Clustering

Given the set of top-k plans, in this section, we will discuss how to group the similar plans using clustering techniques. In practice, many of the generated top-k plans are only slightly different from each other. That is, they do seem to be duplicates of each other, except for one or more states or actions that are different. This may be the result of the underlining AI planner which tries to generate all alternative low-cost plans, and while this generates distinct low-cost plans, it does not always mean that these plans are significantly different from each other. Hence, instead of presenting large number of plans, some of which could be very similar to each other, with the help of clustering, we can present clusters of plans, where each cluster can be replaced by its representative plan.

Clustering has been a topic of interest in several areas of research within several communities such as Information Retrieval (e.g, [2]), machine learning, and Data Management as part of the data cleaning process (e.g., [10]). Many survey papers exist on clustering algorithms (e.g, [28, 7]). While most, if not all, clustering algorithms share a common goal of creating clusters that minimize the intra-cluster distance (distance between members of the same clusters) and maximize the inter-cluster distance (distance between members of different clusters), the assumptions and inputs for these clustering algorithm are often different. For example, several of these approaches assume some given input parameters such as the number of clusters or a cluster diameter. In this paper, we cluster the plans without specifying input parameters such as the number of clusters. This is because no prior knowledge on the number of clusters or the size of the cluster is available. Depending on the domain, there could be cases where many plans can be put into a single cluster due to high similarity, and there are also cases that the plans are all different, and the output must contain clusters of size 1.

To consolidate similar plans produced by the top-k planner, we apply a clustering algorithm that must satisfy the requirements stated below. One representative plan from each cluster is selected to be included in the final set of diverse plans.

Definition 3 (Clustering Requirements) *Given a set of k sorted plans, Π, create clusters of plans $\mathcal{C} = \{c_1, ..., c_o\}$ where the value of o is unknown ahead of time. Further, for each two clusters c, $c' \in \mathcal{C}$, $c \cap c' = \emptyset$ and $\forall \pi \in \Pi$, $\exists c \in \mathcal{C}$ such that $\pi \in c$. Hence, the clusters are disjoint and each plan belongs to one cluster.*

We then may choose to present only a subset of these clusters to the user or to the automated system for further investigation.

3.1 Plan Similarity

Finding if two plans are similar has been studied mainly under two categories: plan stability for replanning (e.g., [8]) and finding diverse plans (e.g., [16]). While some domain-dependent approaches exist (e.g., [15]), majority of recent research has focused on domain-independent measures. In this section, we first briefly discuss ways of representing a plan, and then discuss several similarity measures we consider.

Two plans can be compared based on their actions, states, or causal links [16]. In this paper, we focus on actions and states considering them as both sets and sequences. That is we consider both representing a plan by its set of actions as well as its set of states. We also consider representing a plan by its sequence of actions as well as its sequence of states. Our work is in line with prior work, except our states are not planning states (or set of propositions), but rather a possibly hidden behavioral Finite State Machine (FSM) states. They can be inferred from the semantics of the domain using machine learning or process mining. For example, in the malware detection example, a state can be "crawling", "infectionByNeighbor", or "infectionByDownload". Further, we represent a sequence of actions or states as a sequence of strings by treating each action or state as a symbol. This allows us to use a string similarity measure to compare plans. We also consider comparing plans solely based on their costs or their final states, as it may be enough to group plans based on just their costs (notion of plausibly) or the final state in the plan, a major factor in deciding what to do next in order to detect or predict malware.

Next, we go over the similarity measures we consider. Each similarity measure assigns a number between 0 (unrelated) or 1 (if they are the same). Two plans are said to be similar if their similarity score is above a predefined threshold θ. The similarity measures can be used individually or be combined using a weighted average.

As we will see in the experiments, the choice of similarity measure influences the quality of the clusters, and our framework allows the users to choose any similarity measure or their combination.

3.1.1 Generalized Edit Similarity (GES)

GES [4] can be used to compare sequences of states (or actions) by viewing each state (or action) as a "token" in a string, and the sequence itself as a sequence of tokens. An important reason for choosing GES is that it not only considers the similarity between sequences, but also considers the similarity between tokens (i.e., states). Therefore, we are able to use any extra domain-dependent knowledge at hand about the relationship between states (or actions) to determine if two plans belong to the same cluster. This allows further semantic information to be included in similarity calculations.

GES takes two strings r and r', in our case the two strings represent sequence of states or actions, and computes their similarity score as a minimum transformation cost required to convert string r to r'. The two strings are first tokenized and then assigned a weight $w(t)$. We use a weight of 1 in our experiments. There are three kinds of transformations: insertion, deletion, and replacement. The token insertion cost is $w(t) \cdot c_{ins}$ where t is the inserted token in r and c_{ins} is the insertion factor which we set to 1. Token deletion has a cost of $w(t)$, where t is the deleted token from r. The replacement cost is $(1 - \text{similarity}(t_1, t_2)) \cdot w(t)$. We can use state/action relationships to determine the similarity between t_1 and t_2. For example, if one state is a child or a parent of another state (or if the two states share a same parent), similarity score is set to a higher number (for example, 0.5), else it is either 0 (if they are unrelated) or 1 (if they are the same).

Let r, r' be defined as the sequence of states (or actions) in plans π and π' respectively, then:

$$\text{sim}_{\text{GES}}(\pi, \pi') = 1 - \min\left(\frac{mct(r, r')}{wt(r)}, 1.0\right) \quad (1)$$

where $mct(r, r')$ is the minimum cost of the transformation between

	Gamer, top-1			Fast-Downward(A^*), top-1			*TK* *, top-50			*TK* *, top-1000		
Problem (# st., # obs.)	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
random (10,10)	0.65	0.85	1.41	0.15	0.20	0.23	0.05	0.06	0.12	0.32	0.38	0.42
malware (25,10)	1.09	1.63	1.86	0.49	0.49	0.50	0.06	0.07	0.11	0.23	0.32	0.44
random (50,10)	2.03	2.70	3.90	1.20	1.36	1.59	0.09	0.11	0.13	0.43	0.48	0.53
random (100,10)	11.70	15.27	23.64	4.09	4.85	5.27	0.18	0.29	0.44	0.67	0.75	0.81
random (10,60)	2.65	3.30	4.30	0.64	0.79	0.99	0.16	0.19	0.22	1.98	2.10	2.24
malware (25,60)	7.22	12.48	22.80	2.57	2.60	2.62	0.08	0.15	0.23	1.08	1.62	2.27
random (50,60)	110.95	203.40	291.04	7.65	8.65	9.59	0.36	0.53	0.68	2.24	2.52	2.75
random (100,60)	-	-	-	26.15	29.20	32.71	0.94	1.66	2.23	2.96	4.07	4.73
random (10,120)	6.22	10.82	17.22	1.25	1.60	2.01	0.32	0.36	0.40	4.07	4.24	4.44
malware (25,120)	39.58	83.25	164.48	5.48	5.51	5.56	0.14	0.23	0.40	2.04	2.86	4.19
random (50,120)	-	-	-	15.67	18.10	19.55	0.80	1.27	1.83	4.67	5.40	6.01
random (100,120)	-	-	-	69.96	75.25	79.57	2.31	4.27	6.04	6.55	9.13	11.37

Table 1. Top-k Planning Performance: minimum, average, and maximum planning time, in seconds, for 15 instances of each problem.

the two strings, and $wt(r)$ is the total weight of the string r. Note, this calculation normalizes the similarity score. This normalization is helpful since it allows to choose similarity threshold independently of the size of the plan.

Note that while sim_{GES} is asymetric, the effect of this is insignificant due to the use of single pass clustering algorithms that calculates each similarity score only once and that each clustering algorithm iterates over the top-k plans starting with the lowest-cost plan. Clustering algorithms will be described in the next section.

3.1.2 Jaccard Similarity

Jaccard similarity (inverse of the plan distance from [16]) measures the ratio of the number of actions (or states) that appear in both plans to the total number of actions (or states) appearing in one of them. Let A(π) be the set of actions (or states) in π, then:

$$\text{sim}_{\text{Jaccard}}(\pi, \pi') = \frac{|A(\pi) \cap A(\pi')|}{|A(\pi) \cup A(\pi')|} \tag{2}$$

3.1.3 Simple Equality

Let q and q' be defined as the final state (or the total cost) of plans π and π', then:

$$\text{sim}_{\text{Equality}}(\pi, \pi') = \begin{cases} 1 & \text{if } q = q' \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

3.2 Clustering Algorithms

We propose the use of the following three non-hierarchical clustering algorithms. Each of these algorithms require visiting each plan only once in order to decide to which cluster they belong to; hence, are called single-pass algorithms. Note, when we refer to computation of similarity between plans, it could be that one similarity measure or a weighted combination of similarity measures is used.

3.2.1 Center-Link

Center-Link clustering algorithm iterates over the top-k plans starting with the lowest-cost plan. For each plan, it computes the similarity to a representative of each cluster created in previous iterations. If there are no clusters that have a representative similar to the plan (i.e., their similarity score is above the threshold θ), a new cluster is created and the plan becomes the representative of that cluster. Otherwise the plan is added to the first cluster whose cluster representative is similar to this plan. Cluster representatives are chosen to be the lowest-cost plans in each cluster. Due to the order of iteration, stating from the lowest-cost plans, the cluster representative is always the first added plan to the cluster. This algorithm is similar to the CENTER algorithm [10], however, the sorted input is different (i.e., plans, as opposed to records in a database). The Center-Link algorithm could result in small number of similarity comparisons because each plan is only compared to the representative plan of each cluster.

3.2.2 Single-Link

Single-Link clustering algorithm is an extension of the Center-Link algorithm, where instead of comparing only with the representative of a cluster, each plan is compared with all members of a cluster, and if the plan is found to be similar to any of the members of that cluster, then it is assigned to that cluster. Single-Link algorithm is a non-hierarchical variation of single-linkage algorithm [28]; the node joins a cluster as long as there is a single link with one of the members of the clusters. This algorithm could result in the smallest number of clusters.

3.2.3 Average-Link

Average-Link algorithm is a simple extension of the Single-Link algorithm, where each plan is compared with all the members of a cluster and the average similarity score is used to determine if the plan belong to that cluster or not. This algorithm results in many similarity comparisons, and could result in large number of clusters. Note, Average-Link clustering is a non-hierarchical variant of hierarchical average-linkage clustering [28].

4 Experimental Evaluation

We have four objectives in our experiments: (1) evaluate the performance of top-k planning by comparing it to planners finding a single cost-optimal plan, (2) evaluate the clustering algorithms and the sensitivity of the results to the threshold, (3) evaluate the different similarity measures we used, (4) evaluate against different diverse planners. In all experiments we used a dual 16-core 2.70 GHz Intel(R) Xeon(R) E5-2680 processor with 256 GB RAM.

4.1 Planning Problems

We used both manually crafted and random problems to create our evaluation benchmark. Our problems are based on the hypothesis generation application described by Sohrabi et al. [25]. This application is a good example of a challenging top-k planning problem, and generated problems typically have a very large number of possible plans with different costs. The planning problems were represented in a STRIPS-like planning language recognized by our planner, as well as in PDDL[14] for Gamer and Fast-Downward.

To generate a random problem instance, we generated a random state transition system with a given number of states. In this setting the states of the state transition system do not map directly to planning states, instead we apply a domain transformation [25], compiling away temporary extended goals and adding penalty actions for imperfect explanations to generate the planning problem from the state transition system and the sequence of observations. As the result, the planning states combine the state of the state transition system with position in observation trace and other context information necessary to link observations to system state, generating a much larger state space for the planner.

We varied the size of the problem by changing the number of the states of the state transition system (for random systems) and the number of observations (for both random and manually crafted systems). Further, in all problems we randomly introduced a small fraction of random and missing observations in the generated observation sequence, to better simulate the conditions where generating multiple hypotheses is required, namely the presence of noise or incompleteness of models.

In addition to randomly generated problems, we used the manually crafted malware detection problem, described in [25] (also in Example), and referred to as "malware" in results. The malware detection problem requires generating hypotheses about the network hosts by analyzing the network traffic data. To make that possible, the state transition system includes the states of the host (e.g., infected with malware due to downloading an executable file or the *Command & Control Rendezvous* state via Internet Relay Chat (IRC)) and transitions between these states, as well as a many-to-many correspondence between states and observations.

4.2 Top-k Planning Performance

In Table 1, we compare the performance of our top-k planner, ***TK****, with k=50 and k=1000. We compare to Gamer [13] (Gamer 2014 version, seq-opt-gamer-2.0) and Fast-Downward [11] (2015 version, with A^*). Both find a single cost-optimal plan, which is equivalent to k=1. Planning time was measured on the same randomly generated problem instances for two different kinds of domains, "malware" and "random", and aggregated over 15 instances of each size, where size was controlled by two domain-specific parameters (the number of system states and observations). We enforced a time limit of 300 seconds. Rows containing "-" are those where none of 15 instances were solved within the time limit.

Overall, ***TK**** is very efficient at finding top-k plans, and in our implementation and our set of problems performs at least as fast or faster than Fast-Downward and Gamer, which is essential for use in applications. Due to soundness and completeness of ***K****, ***TK**** is guaranteed to produce top-k plans and that was confirmed in our experiments. Some of the larger instances proved too difficult for Gamer, and it exceeded the time limit. We can also observe that while the worst-case complexity of ***TK**** includes $O(kn)$ term, we have observed relatively small relative differences in planning time with increasing k, with absolute difference limited by a few seconds, and with relative difference decreasing as problem size increased.

Since ***TK**** performs A^* search to find top-k plans when k=1, and ***TK**** top-50 performs similarly to top-1, ***TK**** top-50 can be expected to perform similarly to Fast-Downward A^*, which we have observed. Although ***TK**** is not fully PPDL compliant, there is no significant difference in language expressivity or knowledge provided to planners, and the difference in performance most likely is explained by more efficient implementation and differences in preprocessing in ***TK****. We do not fully understand why Gamer performed relatively poorly on large problem instances. It was natural to expect a cost-optimal planner to find one optimal plan just as fast or faster than a top-k planner would require to find k plans. Overall, these experiment results support our claim that top-k problems can be solved just efficiently as cost-optimal ones, at least within a certain class of planning domains.

4.3 Evaluation of Clusters

We separate the evaluation of the clustering algorithms from the similarity measures. However, we use the following sets of evaluation measures in both cases: time, measured in second, number of similarity comparisons (# Comp) in thousand, number of clusters (# C), and the following six metrics:

- M1: percentage of clusters with the same final state,
- M2: percentage of clusters with the same last three states,
- M3: inter-cluster diversity via uniqueness,
- M4: inter-cluster diversity via stability,
- M5: intra-cluster diversity via uniqueness, and
- M6: intra-cluster diversity via stability.

M1 and M2 are examples of a domain-dependent metric while the rest could be thought of as domain-independent measures. We measure stability and uniqueness using the following formula from [20]. Note, we modified these formula to make it a number between 0 and 1. Also for intra-cluster evaluations, Π is the set of plans within a cluster and we take the average over all clusters. For inter-cluster evaluations, Π is the set of all cluster representative plans which we take to be the lowest-cost plan in each cluster. Let $\Pi = \{\pi_1, ..., \pi_m\}$ be the set of plans. If $|\Pi| = 1$, $\text{Diversity}_{\text{stability}}(\Pi) = 1$, and $\text{Diversity}_{\text{uniqueness}}(\Pi) = 1$, otherwise for $|\Pi| \geq 1$:

$$\text{Diversity}_{\text{stability}}(\Pi) = \frac{\sum_{\pi_i, \pi_j \in \Pi, i \neq j} [1 - \text{sim}_{\text{Jaccard}}(\pi_i, \pi_j)]}{|\Pi| \times (|\Pi| - 1)} \quad (4)$$

$$\text{Diversity}_{\text{uniqueness}}(\Pi) = \frac{\sum_{\pi_i,\pi_j \in \Pi, i \neq j} \begin{cases} 0 & \text{if } \pi_i \setminus \pi_j = \emptyset \\ 1 & \text{otherwise} \end{cases}}{|\Pi| \times (|\Pi| - 1)} \quad (5)$$

	θ	Time	# of	# of	Last state(s)		Inter-cluster		Intra-cluster	
		(sec)	Comp	C	M1	M2	M3	M4	M5	M6
Center	0.65	**0.66**	**10K**	38	74%	45%	0.77	0.51	**0.60**	0.20
Link	0.75	0.82	17K	67	80%	56%	**0.78**	0.49	**0.60**	0.15
	0.85	1.32	36K	142	89%	75%	0.77	0.46	0.64	**0.09**
Single	0.65	1.83	48K	**26**	72%	43%	0.76	**0.54**	0.62	0.20
Link	0.75	2.18	67K	48	77%	54%	0.77	0.52	0.62	0.16
	0.85	3.28	106K	115	86%	71%	0.77	0.49	0.65	**0.09**
Avg-	0.65	14.27	356K	41	75%	47%	0.76	0.50	0.61	0.20
Link	0.75	12.14	329K	72	82%	60%	0.77	0.47	0.61	0.15
	0.85	11.37	330K	152	**91%**	**77%**	0.77	0.46	0.64	**0.09**

Table 2. Comparisons of the clustering algorithms.

First	Second	Time	# of	Last state(s)		Inter-cluster		Intra-cluster	
		(sec)	C	M1	M2	M3	M4	M5	M6
GES-S		4.04	26	52%	37%	0.87	**0.68**	0.61	0.23
GES-A		3.73	63	57%	47%	0.84	0.63	0.62	0.18
Jaccard-S		4.55	79	65%	58%	**0.96**	0.65	**0.55**	0.08
Jaccard-A		4.11	111	67%	61%	0.83	0.57	0.66	0.11
Last State		6.23	**9**	**100%**	33%	0.70	0.43	0.59	0.26
Cost		**2.32**	13	62%	40%	0.62	0.34	0.65	0.25
GES		3.87	37	54%	41%	0.87	0.67	0.60	0.20
Jaccard		4.28	92	66%	59%	0.90	0.61	0.58	0.09
All		2.95	63	82%	58%	0.76	0.47	0.61	0.17
Last State	GES-S	9.08	33	**100%**	53%	0.71	0.46	0.60	0.20
Last State	GES-A	8.81	74	**100%**	65%	0.70	0.43	0.62	0.15
Last State	Jaccard-S	9.39	106	**100%**	**76%**	0.85	0.54	0.56	**0.06**
Last State	Jaccard-A	9.06	131	**100%**	**76%**	0.73	0.43	0.66	0.09
Cost	GES	3.55	63	73%	60%	0.69	0.41	0.64	0.17
Cost	Jaccard	3.64	155	82%	**76%**	0.77	0.45	0.65	**0.06**

Table 3. Comparisons of the similarity measures.

For both uniqueness and stability we compare plans while representing them by their set of states. We also tested with actions but the results were comparable and not shown. M3-M6 are distance measures with values between 0 (the same) and 1 (different - farthest apart). For M3 and M4, larger the number, more diverse the plans are since we find the diverse plans by presenting only the representative plans from each cluster. For M5 and M6, smaller the number, similar the plans are within a cluster. Hence, the ideal algorithm or clustering measure maximizes M3 and M4 and minimizes M5 and M6. The numbers shown in Table 2 and 3 are averages over all planning problems (5 instances of each size). The bold numbers indicate the best numbers in each case.

Summary of our results with respect to the clustering algorithms is shown in Table 2. Center-Link algorithm is the best algorithm with respect to time as fewer number of similarity comparisons is performed since each plan is only compared to the representatives. Average-Link produces more clusters compared to the other two. As the threshold increases, the number of clusters also increases for all algorithms. With respect to the evaluation metrics, the results does not show a superior clustering algorithm: Average-Link is slightly better with respect to M1 and M2, Single-Link produces slightly more diverse plans with respect to M4, and Center-Link also has slightly better numbers with respect to M5. Hence, the clustering algorithms alone do not seem to influence the metric evaluations. However, Center-Link is the best performing algorithm with respect to time. It also compares fewer plans and produces medium size clusters.

Summary of our results with respect to the similarity measures is shown in Table 3. The top part of the table shows the result where a particular similarity measure is used: GES-S and GES-A indicate that we used equation 1, representing the plan by its sequence of states (or actions); Jaccard-S and Jaccard-A indicate that we used equation 2, representing the plan by its set of states (or actions); and "Last State" and "Cost" indicate we used equation 3. The middle part of the table indicates that we used a combination of similarity measures: GES indicates that we used both GES-S and GES-A (assigning equal weights to both); Jaccard indicates that we used both Jaccard-S and Jaccard-A; and "All" indicates that we used all six similarity measures assigning equal weights to each. The bottom part of the table shows the results for when we first cluster all plans based on the similarity measure shown under the "First" column, then within each cluster, run the clustering algorithm again using the similarity measure shown under the "Second" column.

The results show that grouping based on cost may be fastest and grouping based on the last state satisfies M1 (it forces it to be true). However, these similarity measures give the worst results with respect to the nearly all other metrics. On the other hand, using just Jaccard-S produces most diverse plans with respect to uniqueness (best number for M3) and produces similar plans within a cluster (best numbers for M5 and M6) but it suffers in the M1 and M2 categories. GES-S also produces most diverse plans with respect to stability (largest M4 value). While the time and number of clusters is still reasonable, combining all of the metrics (middle part of the table) do not provide better results. However, the best results are found when we combine measures and run the clustering algorithm for the second time. At the expense of time increase, M1, M2, and M6 results are best when we first group based on the last state and then use Jaccard-S. The M3 and M5 numbers are also close to the best numbers. In conclusion, if time is most important then one can just group based on cost. If having the best results for domain-dependent measures such as M1 and M2 is important, one can enforce these metrics when clustering. To only find diverse plans, you can use either GES-S or Jaccard-S. Finally, if satisfying both the domain-dependent and domain-independent metrics is important then combining similarity measures and using for example "last state" followed by "Jaccard-S" will give the best results.

4.4 Comparison With Diverse Planners

We selected two representative diverse planners, LPG-d [16] (with d=0.1) and Div (Multi-queue A^* MQA_{TD}) [20], and compared to our implementation that included top-k and Average-link clustering,

	Top-k + Average Link				LPG-d				Div			
	T	**Cost**	**M4**	**M3**	**T**	**Cost**	**M4**	**M3**	**T**	**Cost**	**M4**	**M3**
(25, 5)	1	1502	0.51	1	1	3513	0.80	1	1	1789	0.36	0.37
(25, 10)	1	1586	0.41	0.99	59	8426	0.84	1	1	3861	0.44	0.54
(25, 20)	3	1492	0.20	0.99	384	16520	0.87	1	1	7262	0.46	0.53

Table 4. Comparison to diverse planners: planning time, T, in seconds, average plan cost and plan diversity on the malware domain. M3 measures plan diversity via uniqueness. M3 measures plan diversity via stability.

using measures M3 and M4 based on Jaccard similarity. The results in Table 4 are averaged over 5 instances of each size, with 30 minutes time limit. The top-k approach produced 50 plans while LPG-d and Div produced at most 10.

Div places greater emphasis on plan cost, and indeed average plan cost is lower than for LPG-d. However it sometimes produces multiple copies of the same plan, resulting in poor diversity. As expected, the top-k approach produces the lowest average cost with somewhat lower diversity.

5 Related Work

In prior work, we have looked at several problems involving hypothesis generation by planning, including a short version of the present work [24], a study of planner-generated hypotheses in goal and plan recognition settings [23], and applications in malware detection and healthcare [25, 19, 18].

Generating a plan set rather than just one plan has been a subject of interest in several recent papers in the context of generating diverse plans (e.g., [20, 5, 5]). When no preferences or quality or cost metric is provided, it is argued that generating a set of diverse plan is the right approach [16]. Several plan distance measures most of which are domain-independent have been proposed to both guide the search and evaluate the set of diverse of plans (e.g., [26, 3]). On the other hand, given some partial preferences or multiple dimensions of quality such as cost or time, the problem becomes a multi-objective optimization problem where diverse plans should form a Pareto-optimal set [16]. In particular, Sroka and Long [27] argue that the previous work will not find good-quality plans as they are more focused on finding diverse plans since it is "easier to find diverse sets father away from optimal". The work we presented in this paper falls in between. While we are given some notion of quality as measured by cost, the cost function itself is imperfect, and we are not given other objective functions besides costs. So finding one min-cost plan is not enough, nor is finding a diverse set of plans without taking into consideration the cost function. Hence, finding a set of diverse low-cost plans is required.

6 Conclusions

The contributions of this paper are the following: 1) the planning framework based on the decomposition of the problem of finding diverse high-quality plans into top-k planning and clustering stages, with configurable similarity measures; 2) a new top-k planner, *TK**, that applies K^* algorithm to planning problems; 3) efficient clustering algorithms for forming a set of diverse plans from a larger set of high quality plans; and 4) the evaluation of solution quality and performance of individual stages and overall framework on both manually crafted and random hypothesis generation problems and comparison to existing diverse planners.

Our framework allows plugging in different top-k planning techniques, different clustering algorithms, and different similarity measures. We evaluate each of these components separately before carrying out the end-to-end evaluation. Our experiments show that planning time required for top-k planning is comparable to cost-optimal planning that finds a single cost-optimal plan using, for example, Fast-Downward. Our empirical evaluation of the three clustering algorithms we proposed for this task show that Center-Link is the best performing algorithm for our setting as it requires less time, compares fewer plans, and produces medium size clusters, while performing similarly to other algorithms in all evaluation metrics. Our findings with respect to similarity measures show that depending on what is most important, the user can choose the best similarity measure (or a combination). Finally, comparing the end-to-end performance of our framework to diverse planners we find that our approach performs comparably to diverse planners in planning time and diversity, while producing diverse plans with consistently lower cost.

While we considered clustering as a post-processing step to finding top-k plans, it might be possible to both guide the search towards diverse plans as well as towards min-cost plans. In future we plan to study this problem and evaluate whether it provides any significant improvements to our results.

REFERENCES

[1] Husain Aljazzar and Stefan Leue, 'K*: A heuristic search algorithm for finding the k shortest paths', *Artificial Intelligence*, **175**(18), 2129–2154, (2011).

[2] J. A. Aslam, E. Pelekhov, and D. Rus, 'The Star Clustering Algorithm For Static And Dynamic Information Organization', *Journal of Graph Algorithms and Applications*, **8**(1), 95–129, (2004).

[3] Daniel Bryce, 'Landmark-based plan distance measures for diverse planning', in *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 56–64, (2014).

[4] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, 'Robust and Efficient Fuzzy Match for Online Data Cleaning', in *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 313–324, (2003).

[5] Alexandra Coman and Hector Muñoz-Avila, 'Generating diverse plans using quantitative and qualitative plan distance metrics', in *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, pp. 946–951, (2011).

[6] David Eppstein, 'Finding the k shortest paths', *SIAM Journal on Computing*, **28**(2), 652–673, (1998).

[7] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, 'A Survey of Kernel and Spectral Methods for Clustering', *Pattern Recognition*, **41**(1), 176–190, (2008).

[8] Maria Fox, Alfonso Gerevini, Derek Long, and Ivan Serina, 'Plan stability: Replanning versus plan repair', in *Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 212–221, (2006).

[9] Patrik Haslum and Alban Grastien, 'Diagnosis as planning: Two case studies', in *International Scheduling and Planning Applications woRKshop (SPARK)*, pp. 27–44, (2011).

[10] Oktie Hassanzadeh and Renée J. Miller, 'Creating Probabilistic Databases from Duplicated Data', *VLDB Journal*, **18**(5), 1141–1166, (2009).

[11] Malte Helmert, 'The Fast Downward planning system', *Journal of Artificial Intelligence Research*, **26**, 191–246, (2006).

[12] Walter Hoffman and Richard Pavley, 'A method for the solution of the *n*th best path problem', *Journal of the ACM*, **6**(4), 506–514, (1959).

[13] Peter Kissmann, Stefan Edelkamp, and Jörg Hoffmann, 'Gamer and Dynamic-Gamer symbolic search at IPC 2014', in *8th International Planning Competition Booklet (IPC-2014)*, (2014).

[14] Drew V. McDermott, 'PDDL — The Planning Domain Definition Language', Technical Report TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, (1998).

[15] Karen L. Myers and Thomas J. Lee, 'Generating qualitatively different plans through metatheoretic biases biases', in *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI)*, pp. 570–576, (1999).

[16] Tuan Nguyen, Minh Do, Alfonso Gerevini, Ivan Serina, Biplav Srivastava, and Subbarao Kambhampati, 'Generating diverse plans to handle unknown and partially known user preferences', *Artificial Intelligence*, **190**, 1–31, (2012).

[17] Miquel Ramírez and Hector Geffner, 'Plan recognition as planning', in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1778–1783, (2009).

[18] Anton Riabov, Shirin Sohrabi, Octavian Udrea, and Oktie Hassanzadeh, 'Efficient high quality plan exploration for network security', in *International Scheduling and Planning Applications woRKshop (SPARK)*, (2016).

[19] Anton V. Riabov, Shirin Sohrabi, Daby M. Sow, Deepak S. Turaga, Octavian Udrea, and Long H. Vu, 'Planning-based reasoning for automated large-scale data analysis', in *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 282–290, (2015).

[20] Mark Roberts, Adele E. Howe, and Indrajit Ray, 'Evaluating diversity in classical planning', in *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 253–261, (2014).

[21] Shirin Sohrabi, Jorge Baier, and Sheila McIlraith, 'Diagnosis as planning revisited', in *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, pp. 26–36, (2010).

[22] Shirin Sohrabi, Jorge A. Baier, and Sheila A. McIlraith, 'Preferred explanations: Theory and generation via planning', in *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, pp. 261–267, (2011).

[23] Shirin Sohrabi, Anton Riabov, and Octavian Udrea, 'Plan recognition as planning revisited', in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, (2016).

[24] Shirin Sohrabi, Anton Riabov, Octavian Udrea, and Oktie Hassanzadeh, 'Finding diverse high-quality plans for hypothesis generation', in *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, (2016).

[25] Shirin Sohrabi, Octavian Udrea, and Anton Riabov, 'Hypothesis exploration for malware detection using planning', in *Proceedings of the 27th National Conference on Artificial Intelligence (AAAI)*, (2013).

[26] Biplav Srivastava, Tuan Anh Nguyen, Alfonso Gerevini, Subbarao Kambhampati, Minh Binh Do, and Ivan Serina, 'Domain independent approaches for finding diverse plans', in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2016–2022, (2007).

[27] Michal Sroka and Derek Long, 'Exploring metric sensitivity of planners for generation of pareto frontiers', in *Proceedings of the 6th Starting AI Researchers' Symposium (STAIRS)*, pp. 306–317, (2012).

[28] R. Xu and I. Wunsch, 'Survey of Clustering Algorithms', *IEEE Transactions on Neural Networks*, **16**(3), 645–678, (2005).

Multi-modal Markers for Meaning: using behavioural, acoustic and textual cues for automatic, context dependent summarization of lectures

Rebekah Wegener[1] and **Jörg Cassens**[2]

Abstract. Meaning making is most often multi-modal and it is this feature that we make use of in outlining a model for an automatic and context dependent note-taking system for academic lectures. Drawing on semiotic models of gesture and behaviour, linguistic models of text structure and sound, and a rich model of context, we argue that the combination of information from all of these through data triangulation provides a better basis for information extraction and summarization than each alone. Further, we suggest that by using a rich model of context that maps the unfolding of the text in real time with features of the context, we can produce query driven summarization. While this outlines research on academic lectures, future work will focus on optimizing this for different domains such as tele-medicine, minuting for meetings and services for the Deaf.

1 Motivation and research questions

The aim in this research is to examine the potential for using multi-modal markers of importance (behaviour, acoustic, and language markers) to automatically detect the structure of a spoken text and extract contextually important segments from the text. Although the proposed use case for the research is lectures and presentations, there are clear use cases and extensions in automatic tagging and summarization of content from online video streaming, patient notes generation and focus guidance for tele-medicine, minuting and note taking for meetings, and query based summarization for the production of contextually appropriate summarization. The research questions can be expressed as two interrelated questions: firstly, to what extent is it possible to use multi-modal markers (behavioural, acoustic/prosodic and textual) to detect importance in a spoken text and can this be used for accurate automatic summarization? Secondly, to what extent is it possible to utilise multi-modal markers together with a rich context model to develop a query driven summariser for the production of contextually appropriate summarization?

2 Background and state of the art

Text summarization forms an important area of research in both linguistics and natural language processing. Summarization is a difficult task and varied widely depending on the purpose or function of the summarization. Most recent work in natural language processing now integrates lexical, acoustic/prosodic, textual and discourse features for effective summarization [14]. Only recently however, are behavioural features being taken into consideration [10] and here only to summarise the movement in a video. Behaviour is frequently under-utilised as a modality because it is treated as a contextual footnote to speech when it can be equally meaning bearing and can often signal meaning prior to verbalisation [13]. Previous research by us [2, 11] show it is possible to utilise the shared features of behaviour. Further on, it is shown that the individual components of behaviour can be utilised for identification and security ([2] together with industry based extensions of [11]). The novel features of this research include: dynamic tracking of important features in real-time; the integration of behavioural markers with acoustic/prosodic and textual markers for focus direction and summarization triggering; query based summarization for the production of contextually appropriate summarization and the inclusion of human centric computing principles in implementation. Future work will focus on the potential for creating and storing re-combinable meaningful audiovisual snippets for reusable summarization on demand and extension to other domains.

3 Data Triangulation Approach to Meaning

Most recent work on sumarization in the field of natural language processing integrates lexical, acoustic/prosodic, textual and discourse features for effective summarization (Maskey & Hirschberg, [14]). These researchers have shown that the combination of all modalities together achieves better results than any single modality alone, but that each modality has its own specific use in deriving meaning that can be further optimized. Maskey & Hirschberg tested the efficacy of the different modalities (lexical, acoustic/prosodic, structural and discourse features) and found that "a summarization system that uses a combination of these feature sets produces the most accurate summaries, and that a combination of acoustic/prosodic and structural features are enough to build a 'good' summarizer when speech transcription is not available." [14]. Their findings suggest that we may gain considerable vantage by combining these features and augmenting these with behavioural features. No other studies have included behavioural features in their data triangulation and we hope to test the value of this addition in future studies. We also hope to change some of the features included in the other modalities (lexical, acoustic/prosodic, structural and discourse) based on linguistic findings regarding summarization.

Our previous research [2, 11, 21] has focused on demonstrating the effectiveness of modelling behaviour semantically. While behaviour is often treated as a background (contextual) to other modalities, we instead treat behaviour as inherently meaning bearing in its own right

[1] RWTH Aachen University, Germany, email: wegener@anglistik.rwth-aachen.de
[2] University of Hildesheim, Germany, email: cassens@cs.uni-hildesheim.de

and as having its own contextual features. We believe that behaviour, treated semantically, can be used to predict areas of significance and intention. A better understanding of how different modalities interact also leads us to expect that an ensemble model will give us better results than existing models.

While there is metaredundancy between the different modalities [12], the modalities actually do have different semantic potentials and provide us with different sorts of information in any given situation. Since we restrict ourselves in the first step to live, academic lectures, we believe that a useful model for such restricted context can be built. In addition, we can compare our work with the established body of research, thereby testing the effectiveness of each inclusion. An extension to other domains is under consideration once the feasibility of the approach is shown.

4 Field, Tenor and Mode: A rich model of context for person centric computing

Hasan [9] makes a distinction between the social action and those aspects of social action which relate specifically to discourse, or to use her words are 'construed by discourse'. This distinction poses some problems for multi-modal research in that it restricts context of situation to discourse. Indeed, context of situation as Hasan models it includes the other modalities as part of the context rather than as the discourse. This restriction causes some problems for modelling alternate forms of communication such as communication through challenging behaviour, augmented communication or computer mediated communication of some forms.

Because of this limitation, we have previously suggested [3] that Hasan's model of context [8] be combined with Activity theory as outlined by Engeström et al. [4]. Combining the two theories allows for at least two important extensions. Firstly it allows for a much broader definition of discourse to include all forms of social action and secondly, it includes non-human actors as potential meaning makers, an important inclusion for ambient intelligence research.

Hasan maps description at the level of context as a set of system networks [6]. Her contextual configuration (henceforth CC) is a systemic approach to the specification of similarity and contrast across contexts, with the features themselves drawn from networks of field, tenor and mode. This is to present context as if it could be represented through paradigms and realisation rules much as can be now seen in Hasan's own mappings between semantic networks and the lexicogrammar [6]. Hasan's model of context [8] sets out from the traditional Hallidayian conception of context as being "a theoretical construct with three variables". Building on the classical Hallidayian approach, Hasan structures her model of context as follows: field tenor and mode. Together, they can be referred to as the contextual construct. She then goes on to define the "totality of its detailed features - the specific values of field, tenor and mode relevant to any any particular instance of speaking - as the contextual configuration."

5 Generic Structure Potential: a model for dynamic text progression and sequence

Hasan claims that for institutional settings it is possible to state a Generic Structure Potential (GSP) [7]. This is simply to say that there are some contexts which tend more towards being heavily structured and organised and thus are less likely to be open for individual negotiation and more likely to have a recognisable generic structure that is reasonably predictable. Institutional settings are here defined as situations that are multiply coded for context and that have convergent coding [7]. The GSP is an abstraction that represents the 'total range of textual structures available within a genre' [7].

Once the generic structure potential is established, it is possible to outline the structural selection for further instances, mapping the choices for each participant and how these work within the context, after all, 'each text is an individual; each has a distinct identity, in the sense that it is not the replication of any other text' [5]. Thus, while our model of the generic structure provides us with a generic and reusable model of the sequencing of a text, our picture of the context provides us with an idea of the the variation in the paths that a text might take through that generic structure potential.

The first step in the modelling process is to establish the GSP for live academic lectures. This will provide a structural segmentation for the lecture. It also provides a model that can be filled with observations. By doing this, we can improve the machine learning in a threefold way: speed, accuracy, data needed.

The GSP is a representation of the contextual configuration. This means that on top of the general model, we can represent variations across field, tenor, mode and material situational setting. For instance, we expect to encounter discipline specific variations in structure (field related), person-specific variations depending on the experience of the lecturer or the student level (tenor related) and variations according to time of day or semester (related to the material setting). By focusing on live academic lecture, we keep the mode constant.

This will provide rich and contextually sensitive model of structure that will allow us to augment existing models, such as the model proposed by Maskey & Hirschberg [14]. This also lets us map the key lexical items, using a rich annotation utilizing concepts such as cohesion and theme.

In addition to acoustic features such as prosody and intonation commonly used in such tasks, we make use of a semantic model of behaviour, facial expressions and gestures that co-occur with speech. For example, preliminary analysis shows that facial expression could be correlated to the rhetoric thrust and might provide information about which aspects of the lecture the lecturer considers pertinent. It also appears that lecturer behaviour is potentially predictive of shifts in the Generic Potential (for example, moving from a definition to an example in the lecture). Furthermore, gestures that co-occur with speech appear to provide important semantic augmentation of key lexical items.

Combining all of these markers in an ensemble approach should allow us to gain further insight into how the modalities work together to create meaning. At the same time, this conceptual model will allow us to build knowledge representation that can be used in run-time systems for the analysis of live lectures.

6 Conclusions

This research follows established work in the area [18, 20] in combining intelligent signal processing with machine learning and builds on our existing work [2, 11, 21] that demonstrates the effectiveness of modelling behaviour semantically and combining this with computer vision and machine learning approaches. To achieve context sensitive query driven summarization, the research builds on our own existing work on modelling context for context aware computing [1, 20, 21, 22]. To enable comparisons with existing research [15, 16, 17, 19] and provide a realistic use case, this research uses audio visual recordings of academic lectures as the data set. Future work will however extend the test cases to include more challenging domains such as tele-medicine, medicine in general and minuting during business meetings.

REFERENCES

[1] David Butt and Rebekah Wegener, 'The work of concepts: context and metafunction in the systemic functional model', in *Continuing Discourse on language: a functional perspective (vol. 2)*, eds., R. Hasan, C.M.I.M. Matthiessen, and J. Webster, London, (2007). Equinox.

[2] David Butt, Rebekah Wegener, and Jörg Cassens, 'Modelling behaviour semantically', in *Proceedings of CONTEXT 2013*, eds., P. Brézillon, P. Blackburn, and R. Dapoigny, volume LNCS, pp. 343–349, Annecy, France, (2013). Springer.

[3] Jörg Cassens and Rebekah Wegener, 'Making use of abstract concepts – systemic-functional linguistics and ambient intelligence', in *Artificial Intelligence in Theory and Practice II – IFIP 20^{th} World Computer Congress, IFIP AI Stream*, ed., Max Bramer, volume 276 of *IFIP*, pp. 205–214, Milano, Italy, (2008). Springer.

[4] Y. Engeström, R. Miettinen, and R.L. Punamäki, *Perspectives on Activity Theory*, Learning in Doing: Social, Cognitive and Computational Perspectives, Cambridge University Press, 1999.

[5] Ruqaiya Hasan, 'Contexts for meaning', in *Georgetown University Round Table on Language and Linguistics 1992: Language, communication and social meaning*, ed., J. E. Alatis, Washington, (1993). Georgetown University Press.

[6] Ruqaiya Hasan, 'Semantic networks: A tool for the analysis of meaning', in *Ways of Saying, Ways of Meaning: Selected papers of Ruqaiya Hasan*, eds., C. Cloran, D. G. Butt, and G. Williams, London, (1996). Cassell.

[7] Ruqaiya Hasan, 'What's going on: a dynamic view of context in language', in *Ways of Saying, Ways of Meaning: Selected papers of Ruqaiya Hasan*, eds., C. Cloran, D. G. Butt, and G. Williams, London, (1996). Cassell.

[8] Ruqaiya Hasan, 'Speaking with reference to context', in *Text and Context in Functional Linguistics: systemic perspectives*, ed., Mohsen Ghadessy, Amsterdam, (1999). John Benjamins.

[9] Ruqaiya Hasan, 'Analysing discursive variation', in *Systemic Functional Linguistics and Critical Discourse Analysis: studies in social change*, eds., L. Young and C. Harrison, London, (2004). Continuum.

[10] Fairouz Hussein, Sari Awwad, and Massimo Piccardi, 'Joint action recognition and summarization by sub-modular inference', in *ICASSP*, (2016).

[11] Anders Kofod-Petersen, Rebekah Wegener, and Jörg Cassens, 'Closed doors – modelling intention in behavioural interfaces', in *Proceedings of the Norwegian Artificial Intelligence Society Symposium (NAIS 2009)*, eds., Anders Kofod-Petersen, Helge Langseth, and Odd Erik Gundersen, pp. 93–102, Trondheim, Norway, (November 2009). Tapir Akademiske Forlag.

[12] Jay L. Lemke, 'Text production and dynamic text semantics', in *Systemic Linguistics: Approaches and Uses*, ed., E. Ventola, Berlin, (1991). Mouton/deGruyter.

[13] Annabelle Lukin, Alison Moore, Maria Herke, Rebekah Wegener, and Canzhong Wu, 'Halliday's model of register revisited and explored', *Linguistics and the Human sciences*, (2011).

[14] Sameer Maskey and Julia Hirschberg, 'Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization', in *INTERSPEECH*, (2005).

[15] Toni-Jan Keith Monserrat, Shengdong Zhao, Kevin Mcgee, and Anshul Vikram Pandey, 'Notevideo: Facilitating navigation of blackboard-style lecture videos', in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pp. 2897–2898, New York, NY, USA, (2013). ACM.

[16] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala, 'Video digests: A browsable, skimmable format for informational lecture videos', in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pp. 573–582, New York, NY, USA, (2014). ACM.

[17] Kannan Rajkumar and Frédéric Andrès, 'Towards automated lecture capture, navigation and delivery system for web-lecture on demand', *CoRR*, **abs/1003.3533**, (2010).

[18] B.W. Schuller, *Intelligent Audio Analysis*, Signals and Communication Technology, Springer Berlin Heidelberg, 2013.

[19] Hijung Valentina Shin, Floraine Berthouzoz, Wilmot Li, and Frédo Durand, 'Visual transcripts: Lecture notes from blackboard-style lecture videos', *ACM Trans. Graph.*, **34**(6), 240:1–240:10, (October 2015).

[20] R. Wegener, C. Kohlschein, S. Jeske, and B Schuller, 'Automatic detection of textual triggers for reader emotion in short stories', in *LREC*, (2016).

[21] Rebekah Wegener, 'Studying language in society and society through language: context and multimodal communication', in *Society in language, language in society: essays in honour of Ruqaiya Hasan*, eds., Wendy L. Bowcher and Jennifer Yameng Liang. Palgrave Macmillan, (2016).

[22] Rebekah Wegener, Jörg Cassens, and David Butt, 'Start making sense: Systemic functional linguistics and ambient intelligence', *Revue d'Intelligence Artificielle*, **22**(5), 629–645, (2008).