

# A Survey on Applications of Bipartite Graph Edit Distance

Michael Stauffer<sup>1,4</sup>, Thomas Tschachtli<sup>1</sup>, Andreas Fischer<sup>2,3</sup>, and Kaspar Riesen<sup>1</sup>

<sup>1</sup> University of Applied Sciences and Arts Northwestern Switzerland,  
Institute for Information Systems, Riggenbachstr. 16, 4600 Olten, Switzerland  
`{michael.stauffer,kaspar.riesen}@fhnw.ch`

<sup>2</sup> University of Fribourg, Department of Informatics, 1700 Fribourg, Switzerland  
`andreas.fischer@unifr.ch`

<sup>3</sup> University of Applied Sciences and Arts Western Switzerland, Institute for Complex Systems, 1705 Fribourg, Switzerland

<sup>4</sup> University of Pretoria, Department of Informatics, Pretoria, South Africa

**Abstract.** About ten years ago, a novel graph edit distance framework based on bipartite graph matching has been introduced. This particular framework allows the approximation of graph edit distance in cubic time. This, in turn, makes the concept of graph edit distance also applicable to larger graphs. In the last decade the corresponding paper has been cited more than 360 times. Besides various extensions from the methodological point of view, we also observe a great variety of applications that make use of the bipartite graph matching framework. The present paper aims at giving a first survey on these applications stemming from six different categories (which range from document analysis, over biometrics to malware detection).

**Keywords:** Applications of Bipartite Graph Matching, Graph-based Pattern Representations

## 1 Introduction

Most pattern recognition applications are either based on statistical (i.e. vectorial) or structural data structures (i.e. strings, trees, or graphs). Graphs, in contrast to feature vectors, are able to represent both entities and binary relationships that might exist between subparts of these entities. Moreover, graphs can adapt their size and complexity to the size and complexity of the actual pattern to be modelled. Due to their representational power and flexibility, graphs have found widespread application in pattern recognition and related fields. Prominent examples of classes of patterns, which can be formally represented in a more suitable and natural way by means of graphs rather than with feature vectors, are chemical compounds [1], documents [2], proteins [3], and networks [4] (see [5] for an early survey on applications of graphs in pattern recognition).

The availability of a dissimilarity or similarity measure is a basic requirement for pattern recognition and analysis. For graph dissimilarity computation,

commonly solved via a particular *graph matching* algorithm, no standard model has been established to date. For an excellent and exhaustive review on graph matching methods emerged during the last forty years, the reader is referred to [6, 7].

The present paper is concerned with the graph matching paradigm of *graph edit distance* [8, 9]. In fact, the concept of graph edit distance is considered as one of the most flexible and versatile graph matching models available. Yet, the major drawback of graph edit distance is its computational complexity that restricts its applicability to graphs of rather small size. Graph edit distance belongs to the family of *Quadratic Assignment Problems (QAPs)*, which in turn belong to the class of  $\mathcal{NP}$ -complete problems. That is, an exact and efficient algorithm for the graph edit distance problem can not be developed unless  $\mathcal{P} = \mathcal{NP}$ .

About ten years ago, an algorithmic framework, which allows the approximate computation of graph edit distance in a substantially faster way than traditional methods on general graphs, has been introduced [10, 11]. The basic idea of this approach, termed *Bipartite Graph Edit Distance (BP)*, is to reduce the difficult QAP of graph edit distance computation to a *Linear Sum Assignment Problem (LSAP)*. LSAPs basically constitute the problem of finding an optimal assignment between two independent sets of entities. For LSAPs quite an arsenal of efficient (i.e. polynomial) algorithms exist (see [12] for an exhaustive survey on LSAP algorithms).

The graph dissimilarity framework BP presented in [10, 11] resolves several major issues that appear when graph edit distance is reformulated to an instance of an LSAP. In a first step the graphs to be matched are subdivided into individual nodes including local structural information. Next, in step 2, an algorithm solving the LSAP is employed in order to find an optimal assignment of the nodes (plus local structures) of both graphs. Finally, in step 3, an approximate graph edit distance, which is globally consistent with the underlying edge structures of both graphs, is derived from the assignment of step 2.

The time complexity of this matching framework is cubic with respect to the number of nodes of the involved graphs. Hence, BP is also applicable to larger graphs. Due to this benefit, the underlying methodology has been employed in a great variety of applications. The contribution of the present paper is to give a first survey on these application fields and the corresponding methods that actually use the BP framework.

## 2 Applications

In the last decade, the original paper (that describes BP for the first time) [10] as well as its extended version [11] have been cited more than 360 times. Regarding these citing papers we observe two main categories. The first category is concerned with methodological extensions of BP. There are, for instance, papers that use another basic cost model than proposed in the original framework [13, 14], or works that aim at making the approximation faster [15, 16], or more accurate [17, 18]. The second category of citing papers is concerned with different applications of

the approximate graph matching framework BP. The main focus of the present paper is to review and categorise the papers of this second category.

A taxonomy of the application fields and the corresponding papers (reviewed in the following subsections) is given in Fig. 1. In all of these applications, graphs are used to represent real-word (or abstract) objects or patterns, such as for instance images, proteins, or business processes (to mention just a few examples). Eventually, the BP framework is used to measure the (dis)similarity between pairs of graph-based representations.

Applications of the Bipartite Graph Edit Distance (BP)

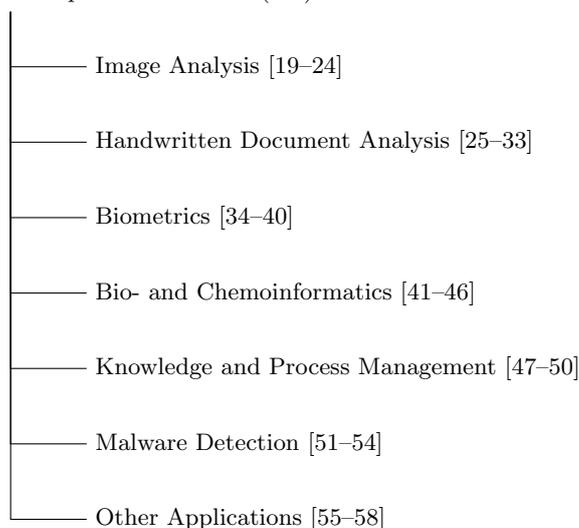


Fig. 1: Taxonomy of the reviewed application fields and papers that use the framework for Bipartite Graph Edit Distance (BP).

## 2.1 Image Analysis

Image analysis is often based on measuring the similarity of objects represented by 2D- or 3D-images. In the present scenario, graphs are used to represent these images, while the dissimilarity between pairs of images is measured by BP. In fact, many of the reviewed application fields presented below can be seen as special case of image analysis. In the present section, however, we present applications that are not part of any of the following subsections.

In [19], for instance, graphs are used to represent envelope images. That is, segmented regions are represented by nodes, while edges are inserted between specific pairs of nodes. The dissimilarities returned by BP are finally used to build a retrieval system. Another image analysis application is presented in [20]. In

this case lunar surface images are formalised with graphs, where nodes represent SIFT-keypoints, while edges represent a Delaunay triangulation of the nodes. The BP distances are eventually used for localisation tasks. In [21] graphs are used to represent thinned images of archaeological structures (so called *Kites*) in order to find similar structures in large aerial image databases. Finally, in [22] the BP framework has been employed for shoe print classification, while in [23] the BP algorithm is used for the computation of similarities between petroglyphs.

Graphs are also used for 3D-images analysis. In [24], for instance, graphs are used to represent topological building structures by a so called *Room Connectivity Graph*. To this end, nodes represent rooms and are labelled by three-dimensional characteristics of the room. The edges are used to represent the connectivity between rooms labelled by two-dimensional features (i.e. width and height). BP is then employed in a retrieval scenario.

## 2.2 Handwritten Document Analysis

In recent years, handwritten (historical) documents have become increasingly digital available. However, the accessibility to these digitised documents with respect to browsing and searching is often limited. A first approach to bridge this gap is presented in [25], which aims at the recognition of unconstrained handwriting images. In this approach, nodes represent keypoints on the skeletonised word images, while edges represent strokes between keypoints. The BP framework (which has been extended in this particular case) is eventually used to define graph similarity features.

*Keyword spotting* allows to retrieve arbitrary keywords in handwritten documents. In case of graph-based keyword spotting, graphs commonly represent (parts of) segmented word images. The nodes of these graphs are, for instance, based on keypoints [26–29] or prototype strokes [30, 31], while edges are commonly used to represent the connectivity between pairs of keypoints or prototype strokes. The actual spotting for certain words in a document is then based on dissimilarity computations between the query graph and document graphs using BP.

The BP dissimilarity framework has not only been applied for spotting keywords in historical documents, but also for clustering ancient ornamental initials (so called *lettrines*) [32]. In this particular case, each *lettrine* is represented by a *Region Adjacency Graph (RAG)*, where nodes are used to represent homogenous regions. Finally, edges are inserted based on the adjacency of regions.

In [33] a graph database for ancient handwritten documents is proposed and evaluated by means of a word classification experiment using BP. In particular, on the basis of segmented word images, six different graph representation formalisms are proposed and compared with each other using the BP dissimilarity model.

## 2.3 Biometrics

Biometrics are often used to verify or identify an individual based on certain biometrical characteristics (e.g. iris, fingerprint, or signature). In [34, 35], retina vessels are used as a biometric trait for user verification. Formally, nodes are

used to represent keypoints in skeletonised vessel images, while edges represent the vessels between selected keypoints. Finally, BP is used to match the retina vessel graphs [34] or to derive different graph similarity measures for building a multiple classifier system [35]. In [36,37] a very similar approach is applied on palm veins rather than retina vessels.

Moreover, graphs are also used to for fingerprint identification as introduced in [38]. In particular, fingerprint images are segmented into core areas (i.e areas with same ridge direction), which are in turn represented by nodes, while edges are inserted between adjacent areas. The resulting fingerprint graphs are then classified using the distances derived from the BP framework.

In recent years, a trend towards high coverage of video surveillance can be observed. Thus, person re-identification over several camera scenes evolved to a crucial task. In [39] a graph-based approach is presented for this task. To this end, segmented camera images are represented by means of a RAG [32]. The BP framework is then used in conjunction with a *Laplacian*-kernel in order to re-identify persons.

Last but not least, BP is also used for on-line signature verification [40]. First, the signatures are divided into segments which are in turn represented by graphs. That is, nodes represent the sample points of a segment, while edges are inserted between specific pairs of nodes. Finally, two signatures are compared with each other by measuring a sum of BP dissimilarities between pairs of graphs.

## 2.4 Bio- and Chemoinformatics

Bio- and chemoinformatics combine approaches and techniques of a broad field to analyse and interpret biological (i.e. DNA, protein sequences) or chemical structures (i.e. molecules), respectively. An important application in the field of bioinformatics is the analysis of deviations in biological structures to detect cancer. In [41], for instance, graphs are used to represent tissue image of biological cells. In this case nodes are used to represent tissue components, while their spatial relationship is represented by edges. Subsequently, the BP framework is used to classify graphs representing normal, low-grade and high-grade cancerous tissue images. In [42], a similar approach is introduced to detect irregularities in blood vessels rather than biological cells.

Chemoinformatics has become a well established field of research. Chemoinformatics is mainly concerned with the prediction of molecular properties by means of informational techniques. The assumption that two similar molecules should have similar activities and properties, is one of the major principles in this particular field. Clearly, molecules can be readily described with labelled graphs, where atoms are represented by nodes, while bonds between atoms (e.g. *single*, *double*, *triple*, or *aromatic*) are represented by edges.

In [43,44] the approximation of graph edit distance by means of BP is used to build a novel graph kernel for activity predictions of molecular compounds. In [45] various graph embeddings methods and graph kernels, which are in part built upon the BP framework, are evaluated in diverse chemoinformatics applications. Finally, in [46] an algorithm to compute single summary graphs from a collection

of molecule graphs has been proposed. The formulation of the cost of a matching, which is actually used in this methodology, is based on BP.

## 2.5 Knowledge and Process Management

In the last decades, a trend towards digitalisation of business models can be observed throughout most industries. Knowledge and process management ensures a thorough information flow, which is actually needed to manage both physical and intellectual resources. Nowadays, business processes are often supported (or completely created) by means of web services. Thus, the re-discoverability of composite web services (described by means of an OWL-S process) is of high relevance and issued in [47]. To this end, a graph is used to represent a composite process. Nodes represent process states and atomic services, while directed edges are used to represent the control flow. By a similar principle, business (sub)-processes rather than web services are retrieved in [48]. In particular, business process activities represent nodes, while directed edges are used to represent the business process flow. Finally, the BP framework is used to find similarities between business (sub)-processes.

Another application in this field is presented in [49], where semantical enriched documents (so called *Resource Description Framework (RDF)* ontologies) are represented by graphs. To this end, document key concepts (e.g. `db:Bob_Dylan`, `db:Folk_Music`) are represented by nodes, while directed edges are used to represent semantic relations (e.g. `dbp:genre`) and labelled by their importance. Finally, the similarity of documents is computed by an adapted BP matching framework.

Based on (similar) RDF ontologies, an approach to estimate the execution time of SPARQL (the RDF query language) queries is presented in [50]. In this scenario the BP distances of an unknown query to a set of training queries are used as query features.

## 2.6 Malware Detection

Anti-virus companies receive huge amounts of samples of potentially harmful executables. This growing amount of data makes robust and automatic detection of malware necessary.

The differentiation between malicious and original binary executables is actually another field where the framework BP has been extensively used. In [51–53], for instance, malware detection based on comparisons of call graphs has been proposed. In particular, the authors propose to represent malware samples as call graphs such that certain variations of the malware can be generalised. This approach enables the detection of structural similarities between samples in a robust way. For pairwise comparisons of these call graphs the approximation of BP is employed.

In [54] a similar approach has been pursued for the detection of malware by using weighted contextual API dependency graphs in conjunction with an extended version of BP for graph comparison. Finally, in [51] BP has been

employed for the development of a polynomial time algorithm for calculating the differences between two binaries.

## 2.7 Other Applications

A further application where the BP matching algorithm has been applied, is for instance, the retrieval of stories (for storytelling) [55]. In this scenario, nodes represent goals and actions, while edges represent time and order. Thus, similar stories can be retrieved by means of the BP framework. In [56] a similar approach is introduced to retrieve sketches used to define the building behaviour of non-player characters in computer games. Finally, the BP framework is also used to detect plagiarism. In particular, [57] and [58] BP is used to detect plagiarism in Haskell programs and in textual documents, respectively.

## 3 Conclusion

In this paper we have reviewed about 40 different papers applying the BP matching framework. The reviewed applications stem from different fields like image analysis, handwritten document analysis, biometrics, bio- and chemoinformatics, knowledge and process management, malware detection, and others. In future work, we plan to extend this survey by integrating not only applications, but also methodological extensions of the BP matching algorithm.

**Acknowledgments.** This work has been supported by the Hasler Foundation Switzerland.

## References

1. Mahé, P., Ueda, N., Akutsu, T., Perret, J.L., Vert, J.P.: Graph kernels for molecular structure-activity relationship analysis with support vector machines. *Journal of Chemical Information and Modeling* **45**(4) (2005) 939–951
2. Schenker, Adam: *Graph-Theoretic Techniques for Web Content Mining*. Volume 62. (2005)
3. Borgwardt, K.M., Kriegel, H.P., Vishwanathan, S.V.N., Schraudolph, N.N.: Graph kernels for disease outcome prediction from protein-protein interaction networks. In: *Pacific Symposium on Biocomputing*. (2007) 4–15
4. Dickinson, P.J., Bunke, H., Dadej, A., Kraetzl, M.: Matching graphs with unique node labels. *Pattern Analysis and Applications* **7**(3) (2004) 243–254
5. Conte, D., Foggia, P., Sansone, C., Vento, M.: Graph matching applications in pattern recognition and image processing. In: *International Conference on Image Processing*. Volume 3. (2003) 21–24
6. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty Years Of Graph Matching In Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence* **18**(03) (2004) 265–298

7. Foggia, P., Percannella, G., Vento, M.: Graph Matching and Learning in Pattern Recognition in the last 10 Years. *International Journal of Pattern Recognition and Artificial Intelligence* **28**(01) (2014)
8. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters* **1**(4) (1983) 245–253
9. Sanfeliu, A., Sanfeliu, A., Fu, K.S.: A Distance Measure Between Attributed Relational Graphs for Pattern Recognition. *IEEE Transactions on Systems, Man and Cybernetics* **13**(3) (1983) 353–362
10. Riesen, K., Neuhaus, M., Bunke, H.: Bipartite Graph Matching for Computing the Edit Distance of Graphs. In: *Graph Based Representations in Pattern Recognition*. Volume 6., Berlin, Heidelberg (2007) 1–12
11. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing* **27**(7) (2009) 950–959
12. Burkard, R., Dell’Amico, M., Martello, S.: *Assignment Problems*. (2009)
13. Serratos, F.: Fast computation of Bipartite graph matching. *Pattern Recognition Letters* **45**(1) (2014) 244–250
14. Gaüzère, B., Bougleux, S., Riesen, K., Brun, L.: Approximate Graph Edit Distance Guided by Bipartite Matching of Bags of Walks. In: *International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*. (2014) 73–82
15. Riesen, K., Ferrer, M., Bunke, H.: Approximate Graph Edit Distance in Quadratic Time. *IEEE Transactions on Computational Biology and Bioinformatics* (99) (2015) 1–1
16. Fischer, A., Riesen, K., Bunke, H.: Improved quadratic time approximation of graph edit distance by combining Hausdorff matching and greedy assignment. *Pattern Recognition Letters* **In Press** (2016)
17. Riesen, K., Bunke, H.: Improving bipartite graph edit distance approximation using various search strategies. *Pattern Recognition* **48**(4) (2015) 1349–1363
18. Riesen, K., Fischer, A., Bunke, H.: Estimating Graph Edit Distance Using Lower and Upper Bounds of Bipartite Approximations. *International Journal of Pattern Recognition and Artificial Intelligence* **29**(02) (2015) 1550011
19. Liu, L., Lu, Y., Suen, C.Y.: Retrieval of Envelope Images Using Graph Matching. In: *International Conference on Document Analysis and Recognition*. (2011) 99–103
20. Zhang, Y., Yang, X., Qiao, H., Liu, Z., Liu, C., Wang, B.: A graph matching based key point correspondence method for lunar surface images. In: *World Congress on Intelligent Control and Automation*. (2016) 1825–1830
21. Madi, K., Seba, H., Kheddouci, H., Barge, O.: A Graph-based approach for Kite recognition. *Pattern Recognition Letters* **In Press** (2016)
22. Hasegawa, M., Tabbone, S.: A Local Adaptation of the Histogram Radon Transform Descriptor: An Application to a Shoe Print Dataset. In: *International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*. (2012) 675–683
23. Seidl, M., Wieser, E., Zeppelzauer, M., Pinz, A., Breiteneder, C.: Graph-Based Shape Similarity of Petroglyphs. In: *European Conference on Computer Vision*. (2015) 133–148
24. Wessel, R., Blümel, I., Ochmann, S., Vock, R.: Efficient Retrieval of 3D Building Models Using Embeddings of Attributed Subgraphs. In: *ACM Conference on Information and Knowledge Management*. (2011) 2097–2100
25. Fischer, A., Suen, C.Y., Frinken, V., Riesen, K., Bunke, H.: A fast matching algorithm for graph-based handwriting recognition. In: *Graph Based Representations in Pattern Recognition*. Volume 7877. (2013) 194–203

26. Riesen, K., Brodić, D., Milivojević, Z.N., Maluckov, Č.A.: Graph Based Keyword Spotting in Medieval Slavic Documents – A Project Outline. In: International Conference on Cultural Heritage. (2014) 724–731
27. Wang, P., Eglin, V., Garcia, C., Largeron, C., Lladós, J., Fornes, A.: A Novel Learning-Free Word Spotting Approach Based on Graph Representation. In: International Workshop on Document Analysis Systems. (2014) 207–211
28. Wang, P., Eglin, V., Garcia, C., Largeron, C., Lladós, J., Fornes, A.: A Coarse-to-Fine Word Spotting Approach for Historical Handwritten Documents Based on Graph Embedding and Graph Edit Distance. In: International Conference on Pattern Recognition. (2014) 3074–3079
29. Stauffer, M., Fischer, A., Riesen, K.: Graph-based Keyword Spotting in Historical Handwritten Documents. In: International Workshop on Structural, Syntactic, and Statistical Pattern Recognition. (2016)
30. Bui, Q.A., Visani, M., Mullet, R.: Unsupervised word spotting using a graph representation based on invariants. In: International Conference on Document Analysis and Recognition. (2015) 616–620
31. Riba, P., Lladós, J., Fornes, A.: Handwritten word spotting by inexact matching of grapheme graphs. In: International Conference on Document Analysis and Recognition. (2015) 781–785
32. Jouli, S., Coustaty, M., Tabbone, S., Ogier, J.M.: NAVIDOMASS: Structural-based approaches towards handling historical documents. In: International Conference on Pattern Recognition. (2010) 946–949
33. Stauffer, M., Fischer, A., Riesen, K.: A Novel Graph Database for Handwritten Word Images. In: International Workshop on Structural, Syntactic, and Statistical Pattern Recognition. (2016)
34. Arakala, A., Davis, S.A., Horadam, K.J.: Retina features based on vessel graph substructures. In: International Joint Conference on Biometrics. (2011) 1–6
35. Lajevardi, S.M., Arakala, A., Davis, S.A., Horadam, K.J.: Retina Verification System Based on Biometric Graph Matching. *IEEE Transactions on Image Processing* **22**(9) (2013) 3625–3635
36. Horadam, K.J., Arakala, A., Davis, S., Lajevardi, S.M.: Hand vein authentication using biometric graph matching. *IET Biometrics* **3**(4) (2014) 302–313
37. Arakala, A., Hao, H., Davis, S., Horadam, K.J.: The Palm Vein Graph for Biometric Authentication. In: International Conference on Information Systems Security and Privacy. (2015) 199–218
38. Choi, Y., Kim, G.: Graph-based fingerprint classification using orientation field in core area. *IEICE Electronics Express* **7**(17) (2010) 1303–1309
39. Brun, L., Conte, D., Foggia, P., Vento, M.: A Graph-Kernel Method for Re-identification. In: International Conference on Image Analysis and Recognition. (2011) 173–182
40. Wang, K., Wang, Y., Zhang, Z.: On-line Signature Verification Using Segment-to-Segment Graph Matching. In: International Conference on Document Analysis and Recognition. (2011) 804–808
41. Ozdemir, E., Gunduz-Demir, C.: A Hybrid Classification Model for Digital Pathology Using Structural and Statistical Pattern Recognition. *IEEE Transactions on Medical Imaging* **32**(2) (2013) 474–483
42. Núñez, J.M., Bernal, J., Ferrer, M., Vilariño, F.: Impact of Keypoint Detection on Graph-Based Characterization of Blood Vessels in Colonoscopy Videos. In: Computer-Assisted and Robotic Endoscopy. (2014) 22–33

43. Brun, L., Conte, D., Foggia, P., Vento, M., Vilemin, D.: Symbolic learning vs. graph kernels: An experimental comparison in a chemical application. In: East-European Conference on Advances in Databases and Information Systems. (2010)
44. Gaüzère, B., Brun, L., Vilemin, D.: Two New Graph Kernels and Applications to Chemoinformatics. In: Graph-Based Representations in Pattern Recognition. (2011) 112–121
45. Gaüzère, B., Hasegawa, M., Brun, L., Tabbone, S.: Implicit and Explicit Graph Embedding: Comparison of Both Approaches on Chemoinformatics Applications. In: International Workshop on Structural, Syntactic, and Statistical Pattern Recognition. (2012) 510–518
46. Koop, D., Freire, J., Silva, C.T.: Visual summaries for graph collections. In: IEEE Pacific Visualization Symposium. (2013) 57–64
47. Cuzzocrea, A., Coi, J.L.D., Fisichella, M., Skoutas, D.: Graph-based matching of composite OWL-S services. In: International Conference on Database Systems for Advanced Applications. (2011) 28–39
48. Niedermann, F.: Deep Business Optimization : concepts and architecture for an analytical business process optimization platform. PhD thesis, University of Stuttgart (2015)
49. Schuhmacher, M., Ponzetto, S.P.: Knowledge-based graph document modeling. In: ACM international conference on Web search and data mining, New York, New York, USA (2014) 543–552
50. Hasan, R., Gandon, F.: A Machine Learning Approach to SPARQL Query Performance Prediction. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). (2014) 266–273
51. Bourquin, M., King, A., Robbins, E.: BinSlayer: accurate comparison of binary executables. In: ACM SIGPLAN on Program Protection and Reverse Engineering, New York, New York, USA (2013) 1–10
52. Elhadi, A.A.E., Maarof, M.A., Osman, A.H.: Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph. American Journal of Applied Sciences **9**(3) (2012) 283–288
53. Kostakis, O., Kinable, J., Mahmoudi, H., Mustonen, K.: Improved call graph comparison using simulated annealing. In: ACM Symposium on Applied Computing, New York, New York, USA (2011) 1516–1523
54. Zhang, M., Duan, Y., Yin, H., Zhao, Z.: Semantics-Aware Android Malware Classification Using Weighted Contextual API Dependency Graphs. In: ACM SIGSAC Conference on Computer and Communications Security, New York, New York, USA (2014) 1105–1116
55. Paul, S.: Exploring story similarities using graph edit distance algorithms. (2013)
56. Flórez-Puga, G., González-Calero, P.A., Jiménez-Díaz, G., Díaz-Agudo, B.: Supporting sketch-based retrieval from a library of reusable behaviours. Expert Systems with Applications **40**(2) (2013) 531–542
57. Kammer, M., Bodlaender, H., Hage, J.: Plagiarism detection in Haskell programs using call graph matching. (2011)
58. Røkenes, H.D.: Graph-based Natural Language Processing: Graph edit distance applied to the task of detecting plagiarism. (2012)