Metrics for Evaluating Network Alignment*

Joel Douglas
Systems & Technology Research
joel.douglas@stresearch.com

Ben Zimmerman
Systems & Technology Research
benjamin.zimmerman@stresearch.com

Alexei Kopylov
HRL Laboratories, LLC
akopylov@hrl.com

Jiejun Xu
HRL Laboratories, LLC
jxu@hrl.com

Daniel Sussman
Boston University
sussman@bu.edu

Vince Lyzinski
University of Massachusetts Amherst
vlyzinski@umass.edu

ABSTRACT

We present the metrics for evaluation of network alignment, which refers to the process of identifying node (and edge) correspondence across multiple networks. These metrics are defined in the context of the DARPA Modeling Adversarial Activity program, where a key technical area is to develop novel mathematical techniques to merge networks from various sources. Specifically, two metrics are presented for evaluation: vertex-based metric and edge-based metric. The first metric is defined based on the assumption that ground-truth alignment is given between vertices across the channels. The idea is to evaluate the approximation of the output from network alignment algorithms with respect to the ground truth. The second metric is defined based on matching edges, which also provides and alternate view on assessing the alignment confidence in scenarios where ground truth alignment is not available. Examples on evaluating different network alignment outputs based on our metrics will be presented in the paper.

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**; **Graph algorithms analysis**; *Data structures design and analysis*;

KEYWORDS

Network Alignment, Evaluation Metric

1 INTRODUCTION

Network Alignment refers to the process of finding node (and edge) correspondence across two or more networks. The basic idea is to align networks (graphs or channels [1]) based on topological consistency (affiliated nodes should have a consistent connectivity structure across the different networks) or other criteria. It is a critical technology which can be applied in a wide variety of context and applications. For instance, network alignment has been applied to connecting identical users across different social media platforms based on the friendship networks [11, 12]. Similarly, network alignment has been applied to the text mining domain for cross-lingual translation using lexicon co-occurence networks [4, 6]. In Bioinformatics, network alignment has been widely studied for aligning protein-protein interaction (PPI) networks from different species in order to determine their common functional structures [1, 7]. More recently, network alignment has become an important tool for cyber-physical system in determining correspondence and dependency among critical nodes from different layers [1, 2].

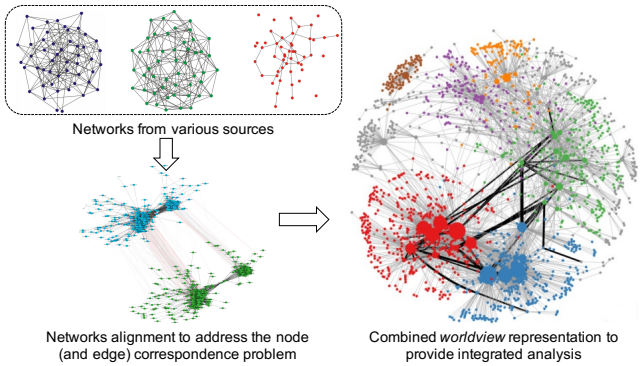

Figure 1: Illustrative example of Network Alignment in the context of the DARPA Modeling Adversarial Activity program [8]. Networks derived from different channels are merged together to provided integrated analysis of adversarial activities.

One of the latest developments in network alignment is on the DARPA-led Modeling Adversarial Activity (MAA) program [8]. The goal of the MAA program is to "develop mathematical and computational techniques for modeling adversarial activity for the purpose of producing high-confidence indications and warnings of efforts to acquire, fabricate, proliferate, and/or deploy weapons of mass terror (WMTs)." MAA assumes that an adversary's WMT activities will result in observable transactions, which may very naturally be modeled using graphs. While the probability is low that any one of the individual graphs will reveal a WMT threat, taken together the probability of detecting a WMT threat will be increased. Network alignment is a key element to enable the generation of a *worldview* network for integrated analysis. Figure 1 shows the high-level idea of the MAA program.

*This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-17-C-0156, FA8750-17-C-0153. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA.

[1]The terms network, graph, and channel are used interchangeably in the paper.

GTA³, February 2018, Marina Del Rey, CA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Given a set of input graphs from different intelligence channels $G_c = (V_c, E_c)$, where V_c is a set of vertexes and E_c is a set of edges, the objective is to create a unified graph $G = (V, E)$ with alignment $m_c : V_c \rightarrow V$ that maps individual ones to the combined network. Each of the input networks are assumed to contain information that are noisy and incomplete, which include the following:

- Nodes and/or edges may be missing
- Nodes and/or edges may be duplicated
- Nodes and/or edges may be mislabeled.

In addition, multiple nodes from one network maybe mapped to a single node in another network. This means network alignment algorithms developed in our context do not necessarily rely on solving the well-known graph isomorphism problem. As a first step toward the MAA program, we aim to develop a uniform method for network alignment evaluation by taking into considerations the aforementioned factors. Specifically, we propose two metrics for evaluation: vertex-based metric and edge-based metric. The first metric is defined based on the assumption that ground-truth alignment is given between vertices across the channels. The idea is to evaluate the approximation of the output from network alignment algorithms with respect to the ground truth. The second metric is defined based on edges, which also provides and alternate view on assessing the alignment confidence in scenarios where ground truth alignment is not available. Detailed description of the two metrics will be discussed in the next section.

2 METRICS FOR NETWORK ALIGNMENT

2.1 Vertex-Based Metric

The vertex-based metric evaluates the aligned vertices across channels (domains) to a defined truth alignment. In this section, the vertex based metric problem is formulated and examples are given which demonstrate the metric.

2.1.1 Problem formulation. For network alignment problems, a set vertex labels are given in seperate channels. The network alignment process produces a new set of labels, where labels may be common between channels, indicating alignment. Figure 2 illustrates a two channel example problem with three nodes in one channel (defined as channel A) and two nodes in a second channel (channel B). The truth shows that two of the entities are aligned across the channels, however these alignments can be unknown and masked by data labels (in the figure, 21 should be aligned with 11 and 22 should be aligned with 12). The data label view is fed as input to a network alignment algorithm which assigns new labels to nodes. The following vertex matrix (V) represents the network alignment output from Figure 2.

$$V = \begin{bmatrix} 1 & 31 & 31 \\ 2 & 33 & 32 \\ 3 & 32 & 0 \end{bmatrix}$$

In the V matrix, the first column is the truth label across channels, and the remaining columns indicate the labels assigned by the alignment system across the channels. A zero in an entry (i, j) indicates that vertex i does not exist in channel $j - 1$. When scoring the network alignment output, the data labels can be used as a mapping from the output labels to the truth labels. The mapping to truth is completed in the following manner: To build the first row of V, find the data labels in channels A and B which map to the true node label (1). In this example, 21 and 11 are the data labels which map to entity 1 in the truth. Now, proceed to map the data labels to the results, which is label 31 in both channels. Hence, we obtain the first row of the V matrix, [1 31 31]. In this paper, all examples are manually formulated outputs from a notional network alignment.

2.1.2 Mathematical Formulation. We define the following notation for the mathematical formulation:

- C: total number of channels
- N_c: number of nodes in channel c
- N_M: number of total matches made by the system
- N_T: number of true alignments between channels
- i, j: indexes into channel; since our metrics are directional, our convention is to calculate the errors from i to j.
- n, m: indexes into nodes. For notational convenience, we refer to node n in channel i as node n^i.
- T_{in}: the truth label of node n^i.
- $\mathcal{T}_j(T_{in}) = \{m : \mathbb{I}(T_{jm} = T_{in}) = 1\}$. In other words, the set of all nodes in channel j that have the same truth label as n^i.
- V_{in}: the label assigned to node n^i by the graph alignment algorithm
- $\mathcal{V}_j(V_{in}) = \{m : \mathbb{I}(V_{jm} = V_{in}) = 1\}$. In other words, the set of all nodes in channel j that have the same network alignment label as n^i.
- $|X|$ is the number of elements in the set X.
- $\mathbb{I}(x)$: indicator function which is 1 if $x \neq 0$, and 0 if $x = 0$.

There are two types of errors that can be made: creating matches that are incorrect and not creating matches we should have made. These metrics can be written in compact mathematical formulas. The incorrectly matched (I_M) metric is the percent of matches made that are not correct, and is written as the following:

$$I_M = \frac{1}{N_M} \sum_n \sum_m \mathbb{I}\left(T_{jm} \neq T_{in} \ \&\& \ V_{jm} = V_{in}\right) \tag{1}$$

The metric is normalized by the number of total matches made from alignment, as we are counting the fraction of matches that are wrong. The metric needs to be computed for each channel combination.

The second metric, incorrectly not matched (I_{NM}), is the percentage of the total number of matches that were missed by the algorithm, and is written as the following:

$$I_{NM} = \frac{1}{N_T} \sum_n \max\left[|\mathcal{T}_j(T_{in})| - |\mathcal{V}_j(V_{in})|, 0\right] \tag{2}$$

The incorrectly not matched metric counts the number of matches in the truth for each node in channel i, $|\mathcal{T}_j(T_{in})|$, and subtracts the number of matches made from the alignment system, $|\mathcal{V}_j(V_{in})|$. If too many matches were made ($|\mathcal{T}_j(T_{in})| - |\mathcal{V}_j(V_{in})| < 0$) the matches will be penalized in the I_M metric. The metric is normalized by the number of matches in the truth, as we are counting the fraction of matches missed. Like the I_M metric, the I_{NM} metric needs to be computed for each channel combination.

The metrics can be aggregated by computing the root mean squared (RMS) accuracy across the different pairs of channels with the following equations:

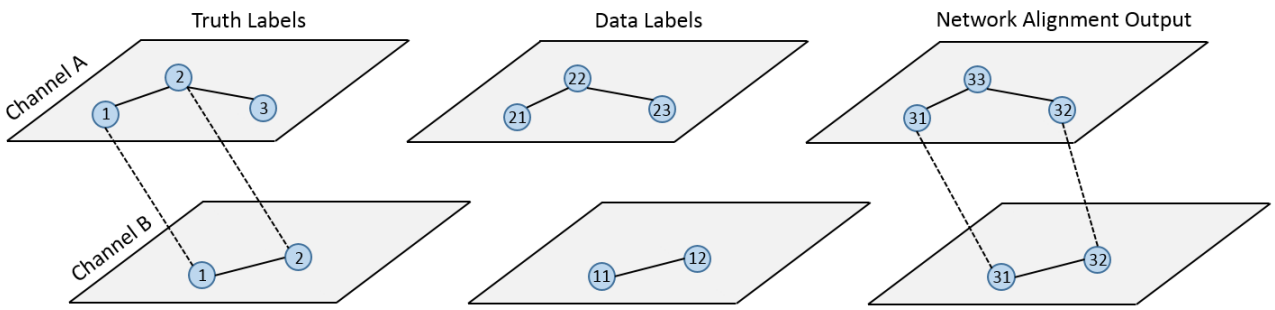

Figure 2: The truth labels, data labels, and network alignment results for a two channel problem. The truth labels indicate two entities should be aligned across channels. The alignments are hidden by the data labels. A network alignment algorithm uses the data labels as input and assigns new labels to the entities. In this example, the alignment between labels 31 is correct, while the second alignment of entity 32 is incorrect.

$$RMS_M = 1 - \sqrt{\frac{\sum(|\boldsymbol{I_M}|^2)}{C^2 - C}} \tag{3}$$

$$RMS_{NM} = 1 - \sqrt{\frac{\sum(|\boldsymbol{I_{NM}}|^2)}{C^2 - C}} \tag{4}$$

Where RMS_M is the matched accuracy, RMS_{NM} is the not matched accuracy, and $|\boldsymbol{x}|^2 = x_i \cdot x_i$.

2.1.3 Example 1. For the example defined in Figure 2, the truth is defined as the following: node 1 is aligned to node 1 and node 2 is aligned to node 2 across the channels, while node 3 exists in channel A and is not aligned to any node in channel B. The output illustrates a labeling error (the true node 2 in channel B is labeled as 32, which aligns to the true node 3 in channel A).

Table 1: Metrics representing Example 1

Channel Direction	I_M	I_{NM}
A to B	0.500	0.500
B to A	0.500	0.000

Table 2: *RMS* metrics representing Example 1

RMS_M	RMS_{NM}
0.500	0.646

The incorrectly matched and incorrectly not matched metrics are reflected in Table 1. When evaluating from channel A to channel B, one node is aligned properly, one node is incorrectly not aligned, and one node is not aligned properly. From the channel B perspective, one node is properly aligned and one node is incorrectly aligned. These labeling errors lead to the I_M and I_{NM} scores reported in Table 1. In addition, the *RMS* accuracies are shown in Table 2. As a reminder, lower scores is better for the I_M and I_{NM} metrics, while scores closer to 1.0 are better for the *RMS* accuracies.

Table 3: Metrics representing Example 2

Channel Direction	I_M	I_{NM}
A to B	0.250	0.250
B to A	0.250	0.000

Table 4: *RMS* metrics representing Example 2

RMS_M	RMS_{NM}
0.750	0.823

2.1.4 Example 2. The following example increases the number of nodes to compare across two channels and introduces the "many to one" problem. In Figure 3, the channel A entity labeled 3 in the truth should be aligned to two entities in the other channel. The defined vertex metrics appropriately handle this issue. We construct the following vertex matrix as example output from network alignment.

$$V = \begin{bmatrix} 1 & 31 & 0 \\ 2 & 32 & 0 \\ 3 & 33 & 33 \\ 3 & 0 & 31 \\ 4 & 34 & 0 \\ 5 & 35 & 35 \\ 6 & 36 & 36 \end{bmatrix}$$

The metrics and *RMS* accuracies are shown in Tables 3 and 4. The I_M metric for both channel comparisons demonstrates that from the four alignments made, only one was incorrect, while the I_{NM} metric score reflects the match we didn't make, but should have (node 33 was only matched once in channel A).

2.1.5 Example 3. The final example demonstrates a problem with three channels. Note that for some cross channel comparisons (channel A and channel B in this example) no errors are present, which is reflected by the zero scores for I_M and I_{NM}. Other channel comparisons in this example exhibit both incorrectly matched and incorrectly not matched errors.

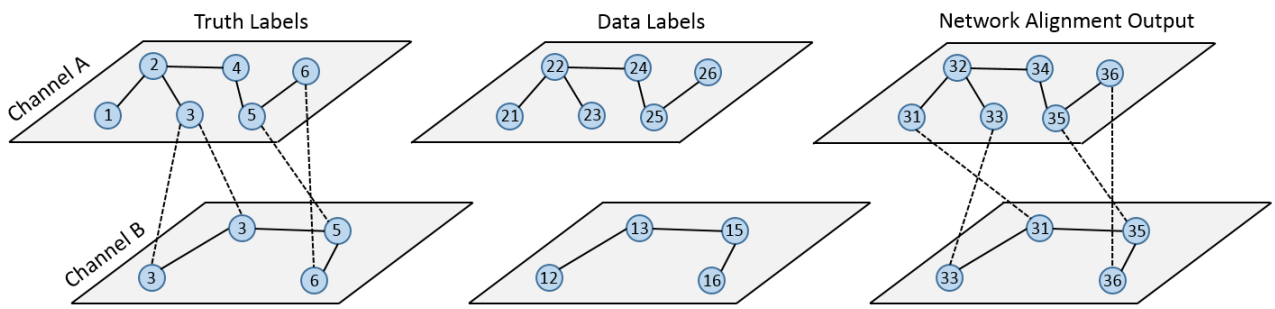

Figure 3: Truth labels, data labels, and network alignment labels for a two channel problem. Note the truth has one entity in channel A (labeled 3) which should not be aligned to two entities in channel B. The results shows three correct alignments (nodes 33, 35 and 36) and one incorrect alignment (node 31). The incorrect association is reflected in the vertex metrics.

Table 5: Metrics representing Example 3

Channel Direction	I_M	I_{NM}
A to B	0.000	0.000
A to C	0.800	0.000
B to A	0.000	0.000
B to C	0.600	0.000
C to A	0.800	0.000
C to B	0.600	0.200

Table 6: *RMS* metrics representing Example 3

RMS_M	RMS_{NM}
0.423	0.918

$$V = \begin{bmatrix} 1 & 0 & 31 & 31 \\ 2 & 32 & 32 & 34 \\ 3 & 33 & 33 & 36 \\ 4 & 34 & 34 & 32 \\ 5 & 35 & 35 & 35 \\ 6 & 36 & 30 & 33 \end{bmatrix}$$

2.2 Edge-based

When the ground truth vertex correspondences are unknown across networks, it is helpful to have a metric that considers the alignment of *edges* across networks. Indeed, this is often the approach in traditional graph matching regimes; see, for example, the graph matching review papers [3, 5] for a multitude of algorithms that adopt an edge-based metric to align graphs. Not only is such a metric useful for giving an objective function to optimize without known across-graph vertex correspondences, it additionally provides an objective function to calibrate accordingly when truth is known.

The edge-metric we propose can be described as follows. Letting $\mathcal{N} = \max_c N_c$, we consider the $\mathcal{N} \times \mathcal{N}$ adjacency matrices $A^{(1)}, \ldots, A^{(C)}$ for the C channel graphs, where (letting "$\sim$" denote the presence of an edge between two vertices)

$$A^{(k)}_{nm} = \begin{cases} 1, & \text{if } n^k \sim m^k \\ -1, & \text{if } n^k \nsim m^k \\ 0, & \text{if either } m \text{ or } n \text{ are not} \\ & \text{present in channel } k \end{cases}$$

We will assume that these adjacency matrices have already been aligned by some algorithm (or correspond to the true alignment). To ease notation, we assume that node n^k in the k-th graph $A^{(k)}$ has been aligned to node n^j in the j-th graph $A^{(j)}$ for each $k, j \in [C]$. Note that if node n is not present in channel k, then we will still include it in $A^{(k)}$, but we will set $A^{(k)}_{n\cdot} = A^{(k)}_{\cdot n} = 0$.

Let $Q = \{-1, 0, 1\}^C$ and let α be a function from Q to $\mathbb{R}$. For any such function α, we define the metric

$$d_\alpha(A^{(1)}, \ldots, A^{(C)}) = \sum_{q \in Q} \alpha(q) \cdot \Big\{\text{number of } (n, m) \text{ pairs such that } A^{(k)}_{n,m} = q_k, \forall k \in [C]\Big\}.$$

By appropriately choosing α this metric can differentially reward and penalize matches and mismatches across the different graphs being matched. This framework is quite general, and can easily be modified to incorporate possible additional pathologies (such as heterogeneity, multiple instantiations of the same vertex, etc.), within each graph.

2.2.1 Choosing α. As the number of graphs being matched grows, the number of parameters in α grows exponentially as $3^C - 1$. This presents a practical burden to both the computation of d_α for general α, and potentially tuning the parameters to account for known ground truth correspondences. To aid the practitioner, we will present simple choices of α that lead to sensible, tractable edge metrics d_α:

- We could consider α to be 0 except for $\alpha(-1, -1, -1) = \alpha(1, 1, 1) = 1$. This would be the natural extension of the standard two-graph graph matching objective function of [9, 10], rewarding full agreements across all networks. We will denote this choice of α as α_1.
- A simple, intuitive option for α is to impose certain symmetry constraints. For example, we can impose that $\alpha(q) = \alpha(q')$

for all q, q' such that the counts of 1's, 0's, and -1's are the same in both q and q'; i.e., q and q' are permutations of each other. This reduces the number of parameters in α to be cubic in m. This particular choice has the advantage of being agnostic to the ordering of the channels.

- An alternative criterion for edge metrics is to impose they are additive across pairs of channels. In particular, we can suppose that for all $\alpha(q) = \sum_{i \neq j} \alpha_{ij}(q_i, q_j)$ where $\alpha_{ij} : \{-1, 1, 0\}^2 \mapsto \mathbb{R}$, and q_i is the i-th element of q. If this property holds then it reduces the effective dimension of α from $3^C - 1$ to $\Theta(C^2)$.

One of the major challenges in adopting an edge-based metric is adopting the right cross-channel matching objective function (i.e., choosing the appropriate α) so that the optimal matching from an optimization perspective is the true matching. As such, one potential role for the vertex metrics with ground truth available is to help tune an edge metric for use when ground truth is absent, guiding the edge metric towards a more suitable optimum. This will allow the practitioner to adapt to different channel topologies, among other data features/eccentricities. This heuristic provides a second option for constructing α will be to do this adaptively based on an observed correspondence, appropriately rewarding graphs which are more similar or more different as desired. However, this approach likely requires a significant research effort, and a detailed implementation is beyond the scope of this manuscript.

It is not always the case that the matching of the nodes implicitly reflects the matching of the edges and vice versa. This would ideally be true, but we have many examples of real data alignment problems where the "true" node matching is inferior to the optimal matching (according to the alignment objective function).

2.2.2 Example computation. Consider a simple setting with $C = 3$. After alignment, the counts of the 27 possible tuples in Q are as follows (where q's with 0 count are omitted in the table):

count	q		
591	-1	-1	-1
26	-1	-1	0
192	-1	-1	1
34	-1	0	-1
11	-1	0	1
217	-1	1	-1
17	-1	1	0
232	-1	1	1
47	0	-1	-1
17	0	-1	1
14	0	1	-1
40	0	1	1
279	1	-1	-1
9	1	-1	0
199	1	-1	1
13	1	0	-1
33	1	0	1
186	1	1	-1
28	1	1	0
590	1	1	1

We can then easily compute $d_{\alpha_1} = 591 + 590 = 1181$. We could consider also a second choice of α that rewards partial agreements as follows (letting $q(i)$ be the number of elements in q equal to i for $i \in \{-1, 0, 1\}$):

$$\alpha(q) = \begin{cases} 1 & \text{if } |q(1) - q(-1)| = 3 \text{ in } q \\ c_1 & \text{if } |q(1) - q(-1)| = 2 \text{ in } q \\ c_2 & \text{if } |q(1) - q(-1)| = 1 \text{ in } q \end{cases}$$

In this example, $d_\alpha = 1181 + c_1 \cdot 208 + c_2 \cdot 1305$. Appropriately choosing c_1 and c_2 would allow the user to reward/penalize partial matches as appropriate.

2.3 Summary

The vertex metrics defined address errors from creating matches that are incorrect and not creating matches that should have been created. For a given problem with C channels, the two vertex metrics will yield $(C^2 - C)$ metrics to encompass all channel combinations. The metrics are capable of handling nodes which shouldn't be aligned and nodes which are aligned more than once. Both vertex metrics can be compressed into a single metric, the *RMS* accuracy, to determine the overall accuracy of a network alignment algorithm.

The edge-based metric provides the means to score the alignment of edges across channels in the absence of vertex truth. The metric depends on the definition of an α function which changes how matches are rewarded and mismatches are penalized in the cross channel alignment. We have presented several choices for α which reward agreements and impose constraints. In addition, the metric provides an objective function to calibrate when truth is available.

3 CONCLUSIONS

We have introduced two metrics for evaluating the output of network alignment in the context of DARPA's MAA program. The first metric is based on measuring the "goodness" of alignment with respect to nodes, and the second metric is defined with respect to edges. These metrics will be used for guiding the design of network alignment algorithms in achieving the objectives for the MAA program. In addition, we expect to develop new metrics as we discover additional alignment criteria and scenarios throughout the program.

REFERENCES

[1] Chen Chen, Hanghang Tong, Lei Xie, Lei Ying, and Qing He. 2016. FASCINATE: Fast Cross-Layer Dependency Inference on Multi-layered Networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 765–774. https://doi.org/10.1145/2939672.2939784

[2] Chen Chen, Hanghang Tong, Lei Xie, Lei Ying, and Qing He. 2017. Cross-Dependency Inference in Multi-Layered Networks: A Collaborative Filtering Perspective. *ACM Trans. Knowl. Discov. Data* 11, 4, Article 42 (June 2017), 26 pages. https://doi.org/10.1145/3056562

[3] D. Conte, P. Foggia, C. Sansone, and M. Vento. 2004. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence* 18, 03 (2004), 265–298.

[4] Dipanjan Das and Slav Petrov. 2011. Unsupervised Part-of-speech Tagging with Bilingual Graph-based Projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 600–609. http://dl.acm.org/citation.cfm?id=2002472.2002549

[5] P. Foggia, G. Percannella, and M. Vento. 2014. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence* 28, 01 (2014), 1450001.

[6] Dehong Gao, Furu Wei, Wenjie Li, Xiaohua Liu, and Ming Zhou. 2015. Cross-lingual Sentiment Lexicon Learning with Bilingual Word Graph Label Propagation. *Comput. Linguist.* 41, 1 (March 2015), 21–40. https://doi.org/10.1162/COLI_a_00207

[7] Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. 2009. IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics* 25, 12 (2009). https://doi.org/10.1093/bioinformatics/btp203

[8] Carey Schwartz. 2017. Modeling Adversarial Activity (MAA). https://www.darpa.mil/program/modeling-adversarial-activity. (2017). Accessed: 2017-11.

[9] J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind, R. T. Vogelstein, and C. E. Priebe. 2014. Fast Approximate Quadratic Programming for Graph Matching. *PLoS ONE* 10, 04 (2014).

[10] M. Zaslavskiy, F. Bach, and J.P. Vert. 2009. A Path Following Algorithm for the Graph Matching Problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31, 12 (2009), 2227–2242.

[11] Jiawei Zhang, Qianyi Zhan, and Philip S. Yu. 2016. *Concurrent Alignment of Multiple Anonymized Social Networks with Generic Stable Matching*. Springer International Publishing, Cham, 173–196. https://doi.org/10.1007/978-3-319-31311-5_8

[12] Xiaoping Zhou, Xun Liang, Haiyan Zhang, and Yuefeng Ma. 2016. Cross-Platform Identification of Anonymous Identical Users in Multiple Social Media Networks. *IEEE Trans. on Knowl. and Data Eng.* 28, 2 (Feb. 2016), 411–424. https://doi.org/10.1109/TKDE.2015.2485222