A story of non-probabilistic representation across modalities and settings

George Giannakopoulos[1]

[1]NCSR Demokritos
ggianna@iit.demokritos.gr

07/04/2021 - Complex Networks Think Tank

Outline

We are here to...

- Offer a different perspective of graph-based representation for learning tasks (and beyond)
- Outline a story of proximity-based graph representation
- Understand what non-probabilistic representation brings
- Describe which interesting and challenging research problems arise

What we should expect to see

- How proximity can model relation in a setting-agnostic manner
- How we can move from probability-based graphs to a possibility-based graphs
- How these graphs are used in learning tasks
- What these graphs have to offer in different, multimodal settings
- Discuss the implications

Outline

Why we need graphs (or what is wrong with vector space)

Indicative vector space limitations:

- Implied independence of features
- Implied (partial/full) ordering
- Difficulty in expressing relations
- Difficulty in expressing transitivity
- At times, very verbose (e.g graph as a neighborhood matrix)

How we use graphs

- We measure their structural parameters, e.g.
  - node traits (number, degree, centrality, ...)
  - edge traits (number, weights, ...)
  - path traits (length, ...)
  - flow traits (maximum, minimum, ...)
  - aggregate traits (density, average node degree, ...)
- We compare graphs
- We index graphs and search for them
- We map them to a vector (!) space, a.k.a. embedding
- We cluster graphs
- We classify graphs
- We index graphs

Graphs across modalities and settings

- Social network analysis [Epasto and Perozzi, 2019]
- Image recognition
  [Auwatanamongkol, 2007, Chen et al., 2019]
- Recommender systems [Ostuni et al., 2014, Nizamkari, 2017]
- Biomedical informatics [Hawe et al., 2019, Tepeli et al., 2020]
- Multi-modal fusion for sentiment analysis [Zadeh et al., 2018]
- Nano-material modeling

The usual (meaning of) graphs

- Nodes are items
- Edges declare relations between nodes
- Weights indicate the "strength" of each relation (oftentimes a probability)

Families of graphs

- Undirected/directed

Families of graphs

- Undirected/directed
- Weighted/unweighted

Families of graphs

- Undirected/directed
- Weighted/unweighted
- Single-partite/multi-partite

Families of graphs

- Undirected/directed
- Weighted/unweighted
- Single-partite/multi-partite
- Single-relation/multi-relational

Families of graphs

- Undirected/directed
- Weighted/unweighted
- Single-partite/multi-partite
- Single-relation/multi-relational
- Graph/hyper-graph

Families of graphs

- Undirected/directed
- Weighted/unweighted
- Single-partite/multi-partite
- Single-relation/multi-relational
- Graph/hyper-graph
- ...and more

To remember

- Graphs offer rich information
- Their meaning can vary a lot, based on the encoding of info in the graph
- We use them a lot, in many settings and across modalities
- We combine them with vector spaces

Outline

A brief intro: a day of 2001 in the bus

- The challenge of efficient retrieval...
- ...regardless of spelling errors...
- ...and time to think, while on the bus.

Intuition — Perception

- People can read even when words are spelled *wnorg*

Intuition — Perception

- People can read even when words are spelled *wnorg*
- But order *does* play some role: *not it does?*

Intuition — Perception

- People can read even when words are spelled *wnorg*
- But order *does* play some role: *not it does?*
- The same stands for images...

Intuition — Perception

- People can read even when words are spelled *wnorg*
- But order *does* play some role: *not it does?*
- The same stands for images...
- ... and audio.

Intuition — Assumptions

- *Neighborhood* (or *Proximity*) can indicate relation (e.g. causality)

Intuition — Assumptions

- *Neighborhood* (or *Proximity*) can indicate relation (e.g. causality)
- An object can be analyzed into its constituent parts

Intuition — Assumptions

- *Neighborhood* (or *Proximity*) can indicate relation (e.g. causality)
- An object can be analyzed into its constituent parts
- You can compute *similarity* or *distance* for the domain of these parts.

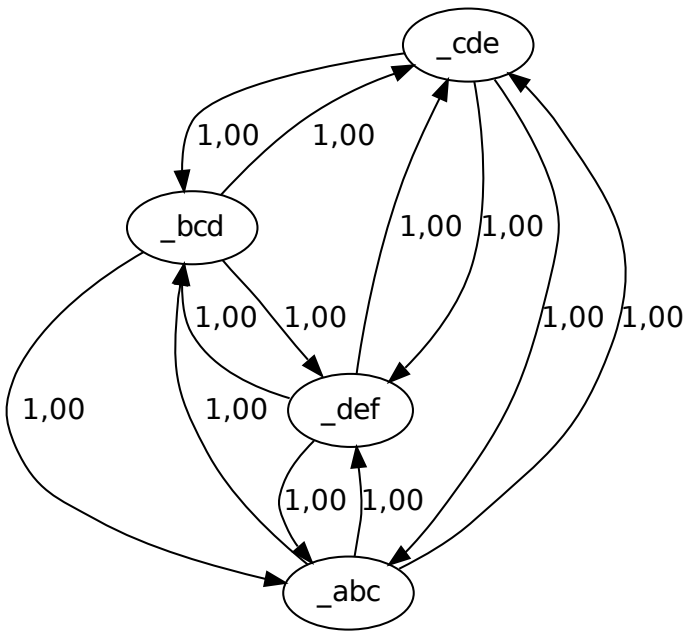
Intuition — Assumptions

- *Neighborhood* (or *Proximity*) can indicate relation (e.g. causality)
- An object can be analyzed into its constituent parts
- You can compute *similarity* or *distance* for the domain of these parts.
- You can define the size of a neighborhood of parts

Intuition — Assumptions

- *Neighborhood* (or *Proximity*) can indicate relation (e.g. causality)
- An object can be analyzed into its constituent parts
- You can compute *similarity* or *distance* for the domain of these parts.
- You can define the size of a neighborhood of parts
- You can see how parts co-occur within a neighborhood

An N-gram Graph [Giannakopoulos et al., 2008]



What does an n-gram graph do?

- Indicates neighborhood
- **Edges** are most important
- It provides an overview (i.e the statistics) of local phenomena in an observation

Extraction Process (NLP example)

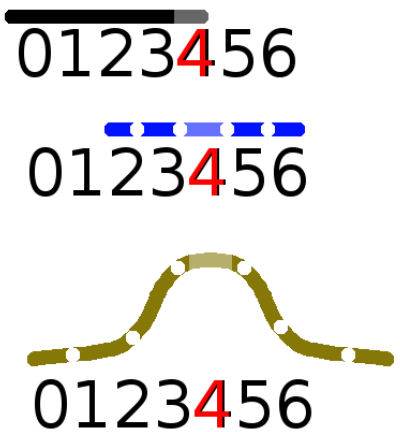
- Extract n-grams of ranks $[L_{\text{min}}, L_{\text{MAX}}]$. One graph per rank.
- Determine neighborhood (window size D_{win}).
- Assign weights to edges.

Example

String:	*abcde*
Character N-grams (Rank 3):	*abc, bcd, cde*
Edges (Window Size 1):	*abc-bcd, bcd-cde*
Weights (Occurrences):	*abc-bcd (1.0) , bcd-cde (1.0)*

Window-based Extraction of Neighborhood — Examples

Figure: N-gram Window Types (top to bottom): non-symmetric, symmetric and gauss-normalized symmetric. Each number represents either a word or a character n-gram

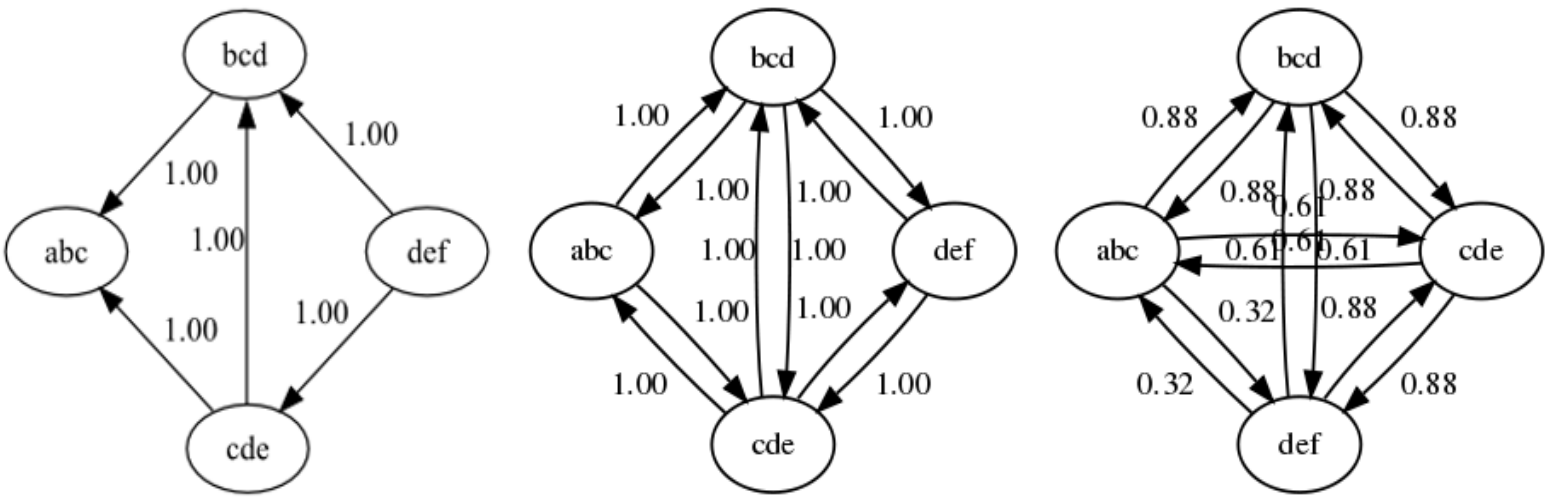


See

[Giannakopoulos et al., 2008, Giannakopoulos, 2009]

N-gram Graph — Representation Examples

Figure: Graphs Representing the String *abcdef* (from left to right): non-symmetric, symmetric and gauss-normalized symmetric. N-Grams of Rank 3. D_{win} value 2.



But why is an NGG useful?

- It represents *relations*
- It contains *generalization* inherently (i.e. is a lossy compression)
- It has some interesting features algorithmically: *unique nodes*
- The edge weights can have different semantics
- It can generalize easily to other settings (hold yer horses... we'll get back to this)
- It has some unique features: e.g. does centrality make a node important?

Interesting questions: the beginning

- What set of strings does a given NGG represent? (Hint: Compression)
- What possible ways do we have for edge weighting? (Hint: Too many, each offering different info)
- Do the typical algorithms for node importance keep their semantics here? (Hint: Not really...)

Outline

Indicative applications of n-gram graphs (NGGs)

- Classification [Rentoumi et al., 2009, Giannakopoulos et al., 2012, Polychronopoulos et al., 2014]
- Clustering [Kim et al., 2016, George Giannakopoulos et al., 2014, Pittaras et al., 2018, Tsekouras et al., 2020]
- Summarization [George Giannakopoulos et al., 2014]
- Summary evaluation [Giannakopoulos et al., 2008, Giannakopoulos and Karkaletsis, 2009, Giannakopoulos and Karkaletsis, 2010]

Classifying with NGGs: first approach

- Represent each instance as a graph
- Define a *metric-based* similarity (or kernel)
- Use a lazy (e.g. nearest neighbor) approach

Classifying with NGGs: second approach

- Represent each instance as a graph
- Represent each class as a graph (by combining instance graphs)
- Examine the similarity of test instances to each of the class graphs
- Assign the label connected to the max similarity

Classifying with NGGs: similarity-based learning approach

- Represent each instance as a graph
- Represent each class as a graph (by combining instance graphs)
- Examine the similarity of test instances to each of the class graphs
- Create a *feature vector*, containing the similarities as dimensions
- Learn a classficiation model

Clustering with NGGs

- Represent each instance as a graph
- Define a *metric-based* similarity (or kernel)
- Utilize distance-based approaches to form clusters

NOTE: Of course, we can always revert to embeddings, but *we need to think why*

What we need so far

- Process for representing instances
- A merging operator (e.g. for class graphs)
- A similarity operator

Outline

Representing Sets of Graphs

A representative graph for a set

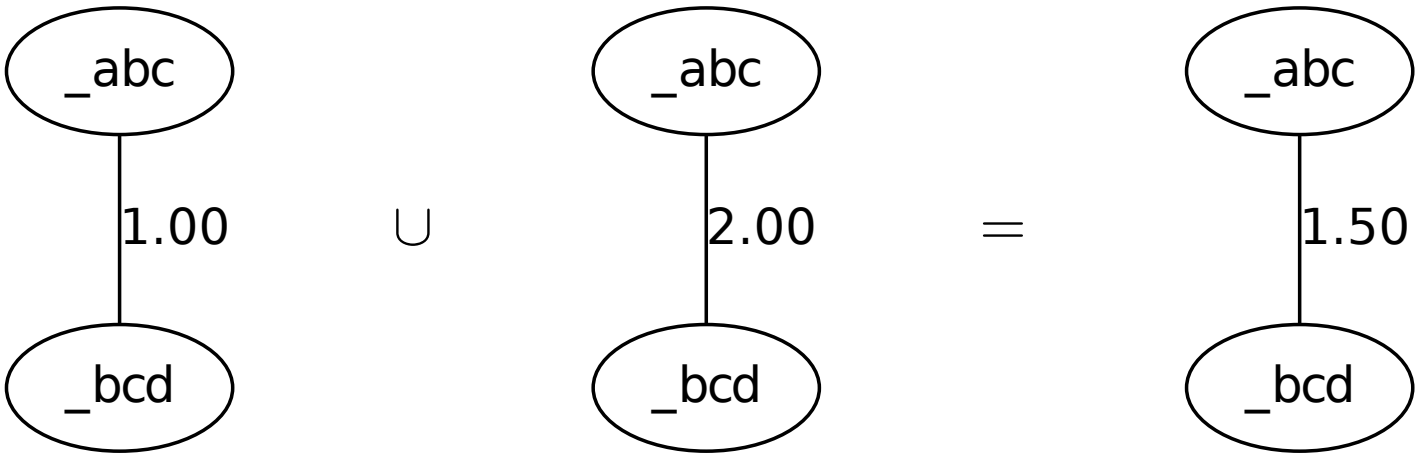
- is similar to functionality to the centroid of vectors
- for edges the effect of merging is **non-linear**

Representing Sets of Graphs

A representative graph for a set

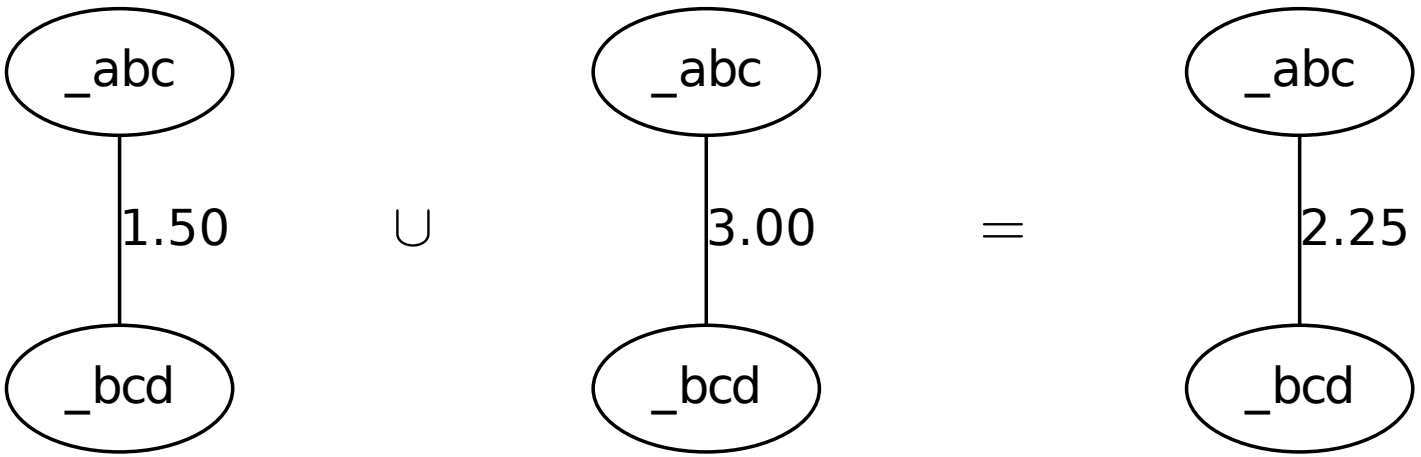
- is similar to functionality to the centroid of vectors
- can be represented using the **update** operator (non-trivial merging)
- for non-common edges in two graphs the effect is **non-linear**

Merge vs. Update (1)



$$\frac{1+2}{2} = 1.5$$

Merge vs. Update (2)



$$\frac{1.5+3}{2} = 2.25$$

But what if we want $\frac{1+2+3}{3} = 2$?

Merge vs. Update (3)

$$\text{updatedValue} = \text{oldValue} + l \times (\text{newValue} - \text{oldValue}) \quad (1)$$

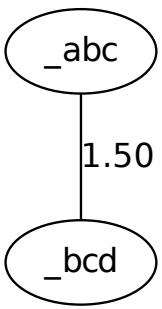
where $0 \leq l \leq 1$ is the learning factor

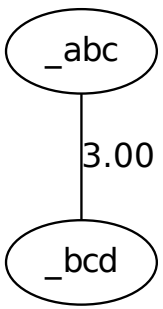
Representative (or *Centroid*) Graph

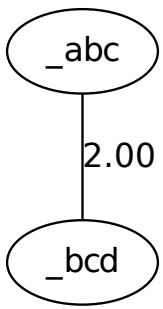
Use update operator with learning factor: $\frac{1}{\text{instanceCount}}$, where *instanceCount* is the number of instances that will be described by the graph *after* the update.

Merge vs. Update (4)

$$I = \frac{1}{3}$$

update(

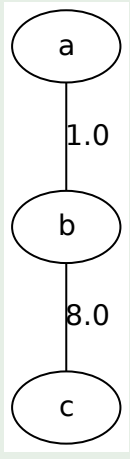
, 

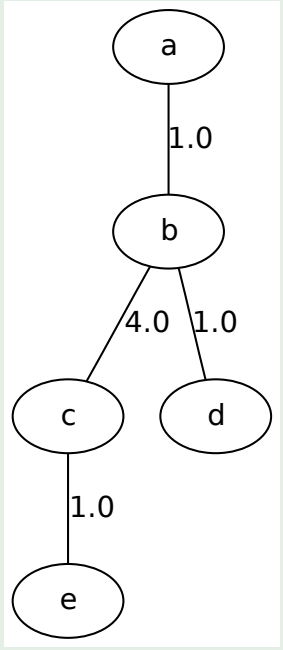
, I) = 

$$1.5 + \frac{1}{3} \times (3 - 1.5) = \frac{3}{2} + \frac{1}{3} \times \frac{3}{2} = \frac{4}{2} = 2$$

N-gram Graph – Size Similarity

Example



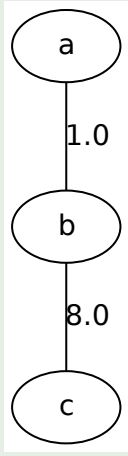


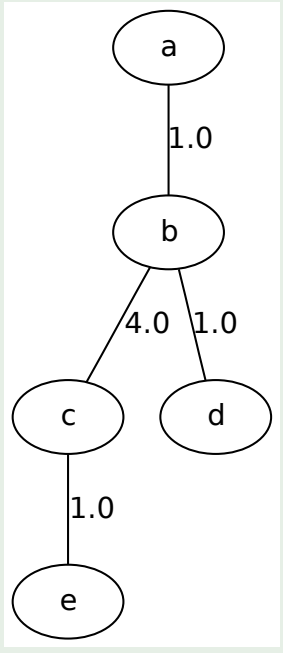
$|G_1| = 2$ $|G_2| = 4$

Result: $\frac{2}{4} = 0.5$

N-gram Graph – Containment Similarity

Example

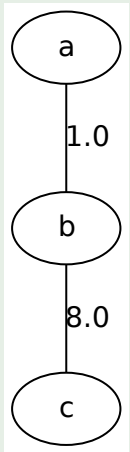


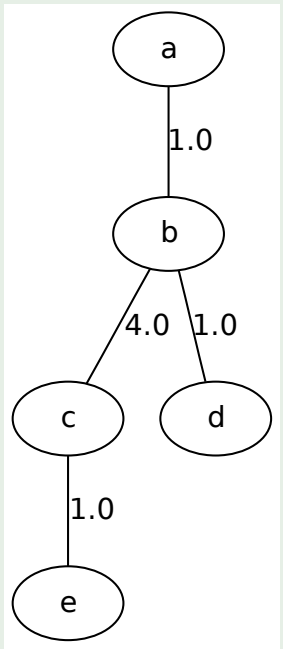


Result: $\frac{1}{2} + \frac{1}{2} = 1.0$

N-gram Graph – Value Similarity

Example

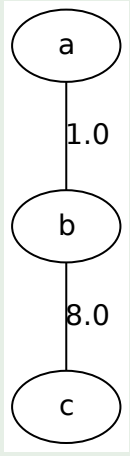


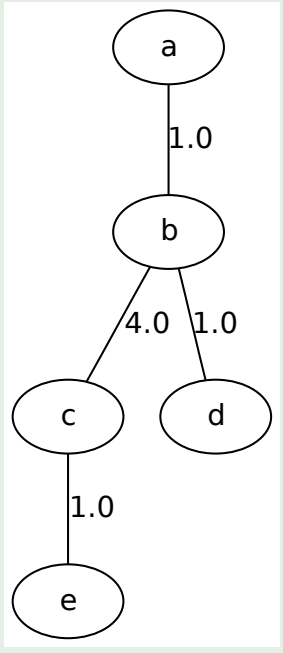


Result: $\frac{\frac{1.0}{1.0}}{4} + \frac{\frac{4.0}{8.0}}{4} = \frac{1}{4} + \frac{1}{8} = 0.375$

N-gram Graph – Normalized Value Similarity

Example





Result: $\frac{VS}{SS} = \frac{0.375}{0.5} = 0.75$

Outline

Is an n-gram graph probabilistic?

Well: not necessarily.
However, we can move from counts (co-occurrences) to normalized counts. :)
But why do it? As they are, n-gram graphs, may offer different semantics:

possibility vs probability

Outline

Proximity graphs: Prerequisites

- We have a whole...
- ...than can be broken down into (identifiable) constituents
- and we have a way to talk about proximity between constituents

Time for a game!

How can we represent the following?

- Text
- Images
- Video
- Audio

What do you think?

Let's listen to some ideas!

Possible answers: Text

- Text / hypertext as a whole
  - Words as constituents, character distance as proximity indicator
  - Characters as constituents, character distance as proximity
  - Entities as constituents, word distance as proximity
  - Markup and words as constituents, sequence distance as proximity

Possible answers: Image

- Image as a whole
  - Pixels as constituents, euclidean distance as proximity indicator
  - Patches as constituents, color histogram distance as proximity indicator
  - Points of Interest as constituents, word distance as proximity indicator
  - Recognized objects as constituents, euclidean distance as proximity indicator
  - Image (as an optical flow tensor) as constituent, euclidean distance as proximity indicator

Possible answers: Video

- Image as a whole
  - Frame (as a color matrix) as constituent, frame number distance as proximity indicator
  - Short term behaviors as constituents, frame number distance as proximity indicator
  - Frame (as an color matrix) as constituent, color histogram distance as proximity indicator

Possible answers: Audio

- Image as a whole
  - Wave position as constituent, time distance as proximity indicator
  - Identified short-term event as constituent, time distance as proximity indicator
  - Audio segment as constituent, wave dissimilarity as proximity indicator

Challenges

- How can we map non-categorical observations to symbols?
  - Quantization
  - Distributions as symbols
- How can we efficiently search whether we have already created a corresponding symbol?
  - Need for indexing

A recap example

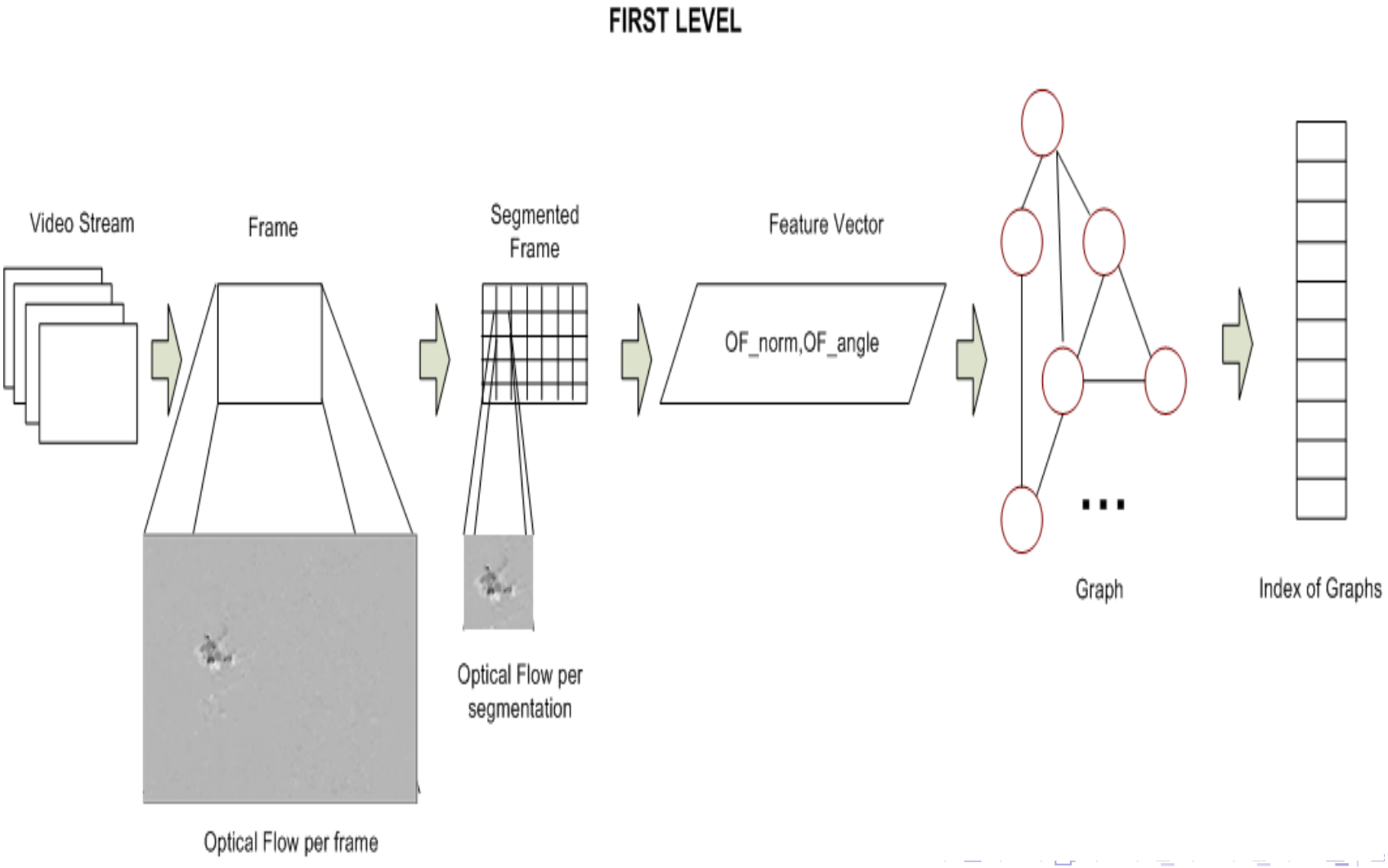
Behavior recognition from video

We have a video with frames. We want to classify the behavior that appears in the sequence.

Kudos to (Semveillance dataset, NCSR “Demokritos”, CIL)

Segmented Frame Representation — Hierarchy (1)

Training Step

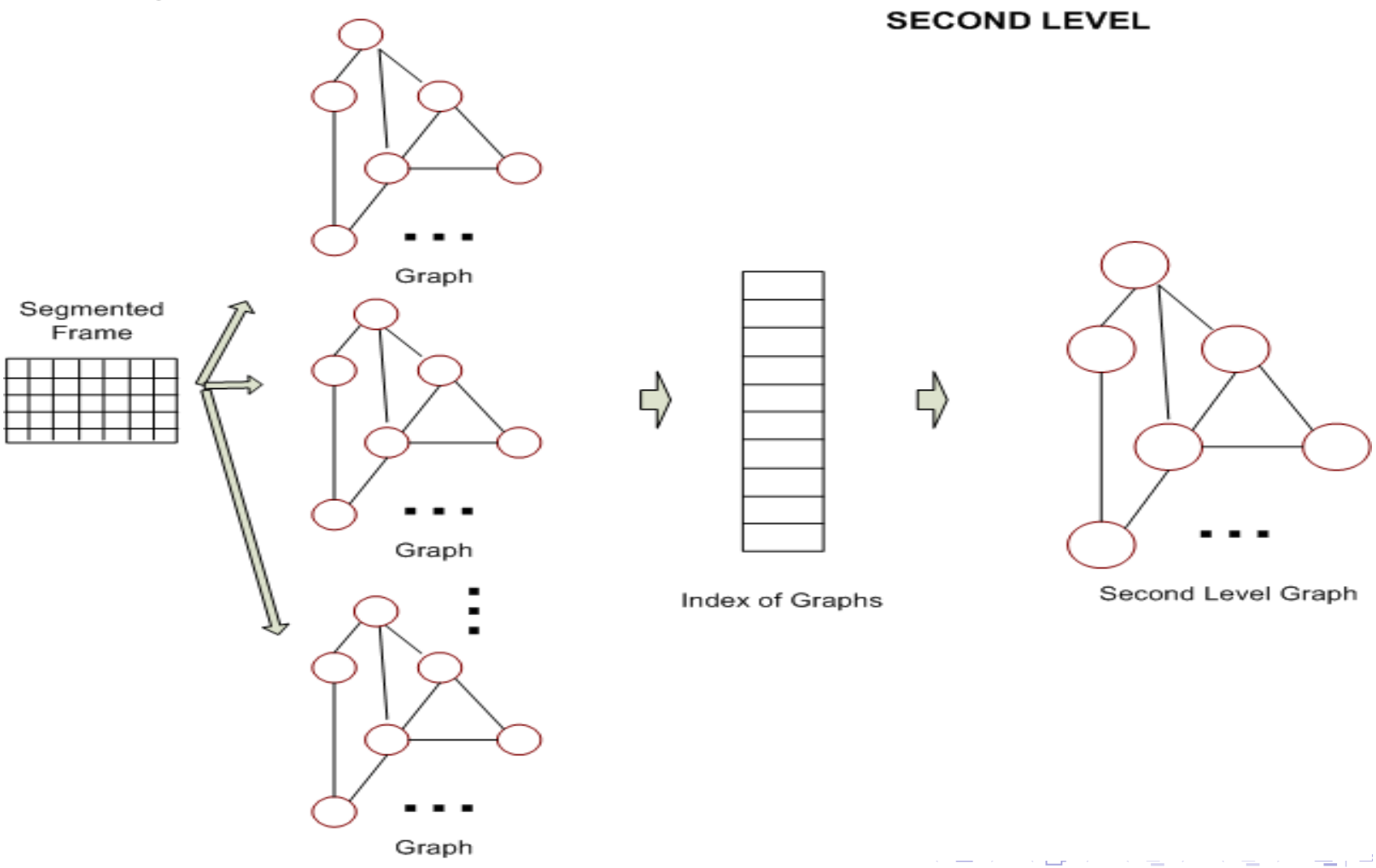


Searching and Updating the Index

```
Data: CurrentGraph, IndexOfGraphs
Result: informed IndexOfGraphs
dgName = null;
for each Graph in set of graphs of IndexOfGraphs do
    compute similarity between CurrentGraph and Graph;
    if similarity ≥ maxForMerging then
        dgName = name of the Graph;
    else if similarity ≥ minForMerging then
        dgName = name of the Graph;
        Graph = result of merging CurrentGraph and Graph;
    else
        CurrentGraph = result of removal of Graph from CurrentGraph
    end
end
if dgName = null then
    assign a new name to the CurrentGraph and add CurrentGraph
    and name in the index;
end
```

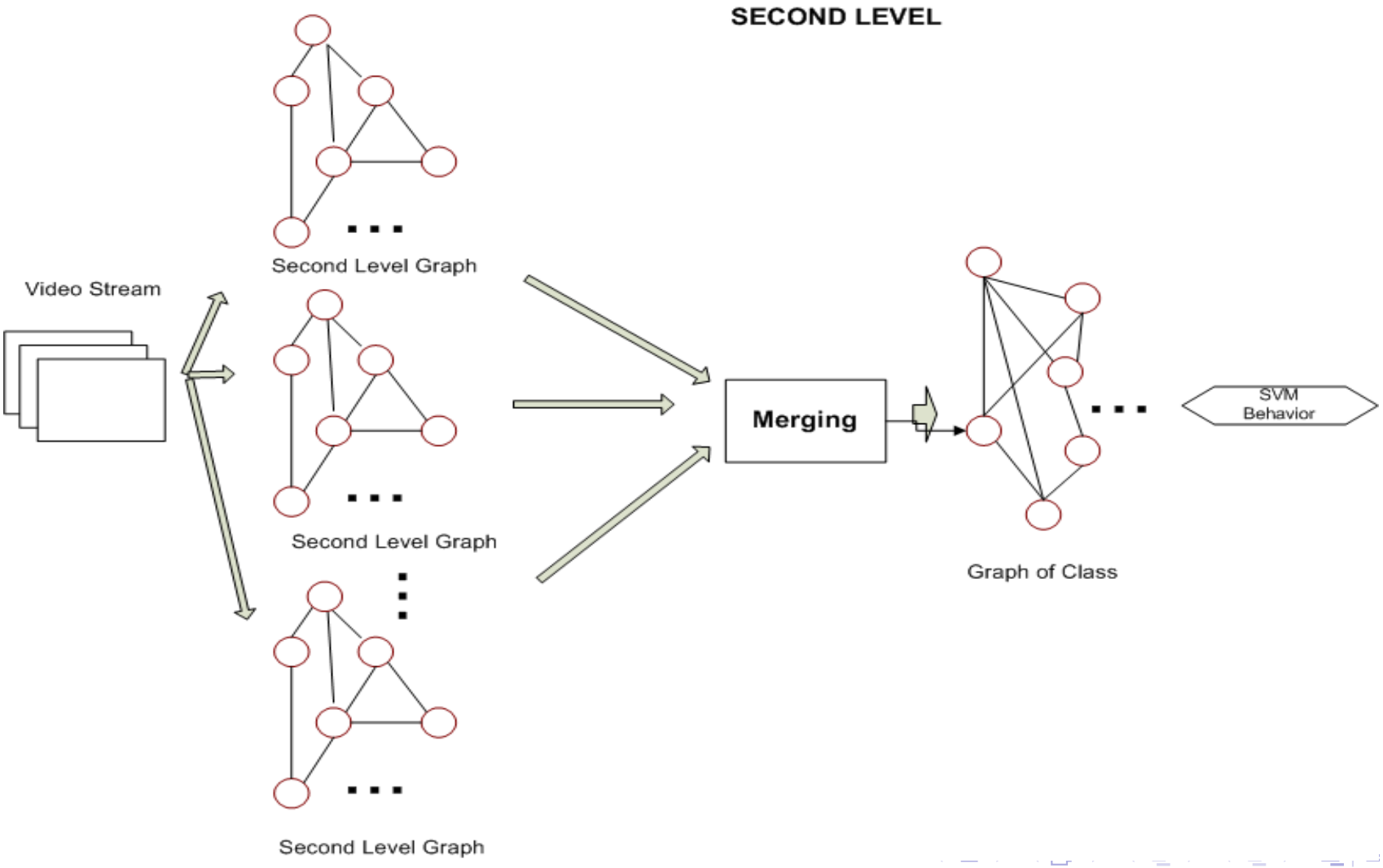
Segmented Frame Representation — Hierarchy (2)

Training Step



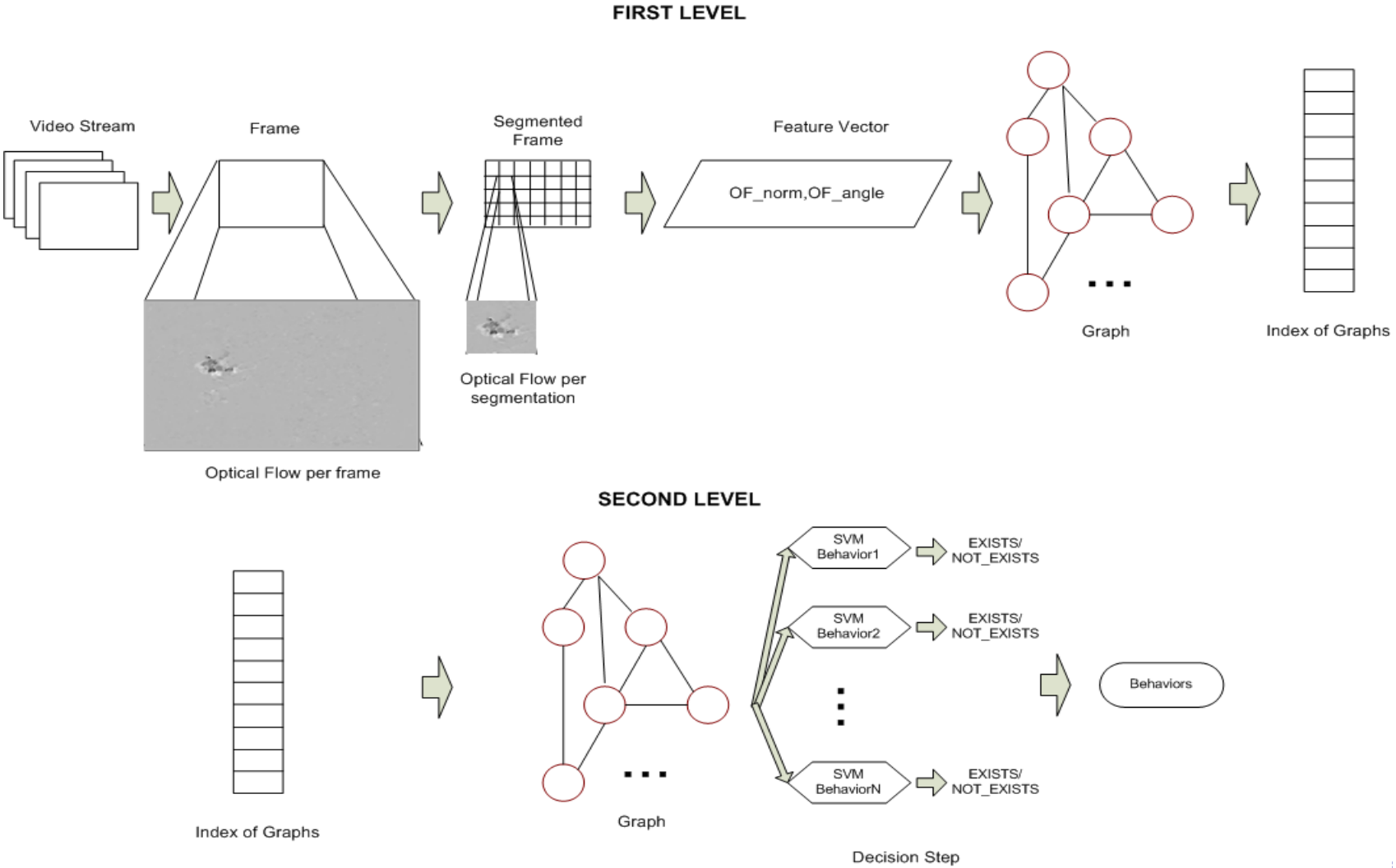
Segmented Frame Representation — Representing a Class

Training Step



Segmented Frame Representation — Testing

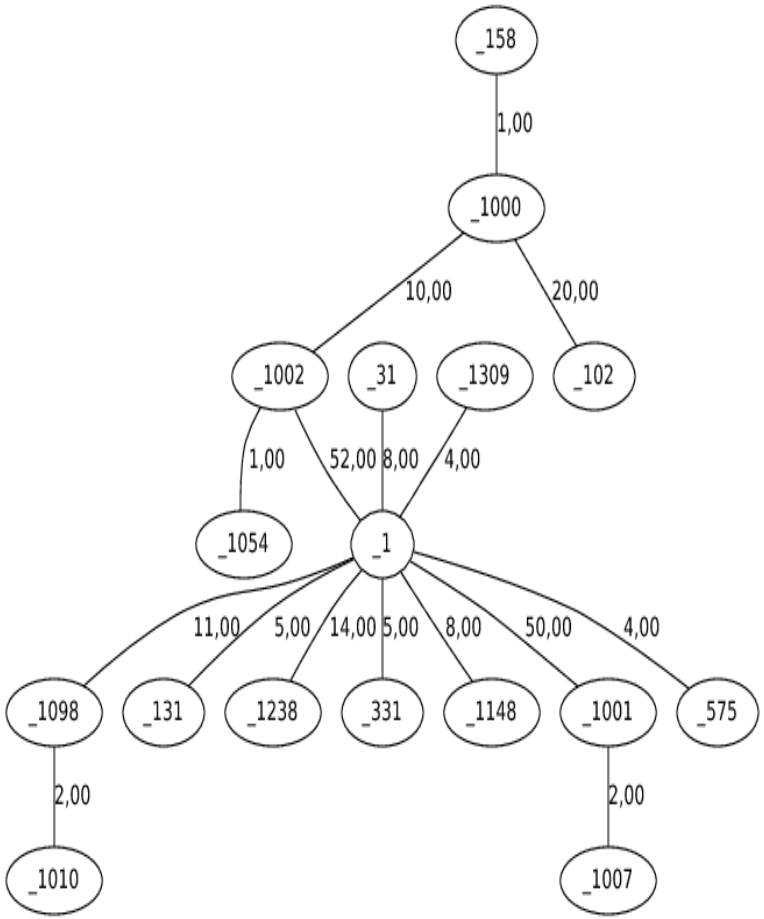
Testing Step



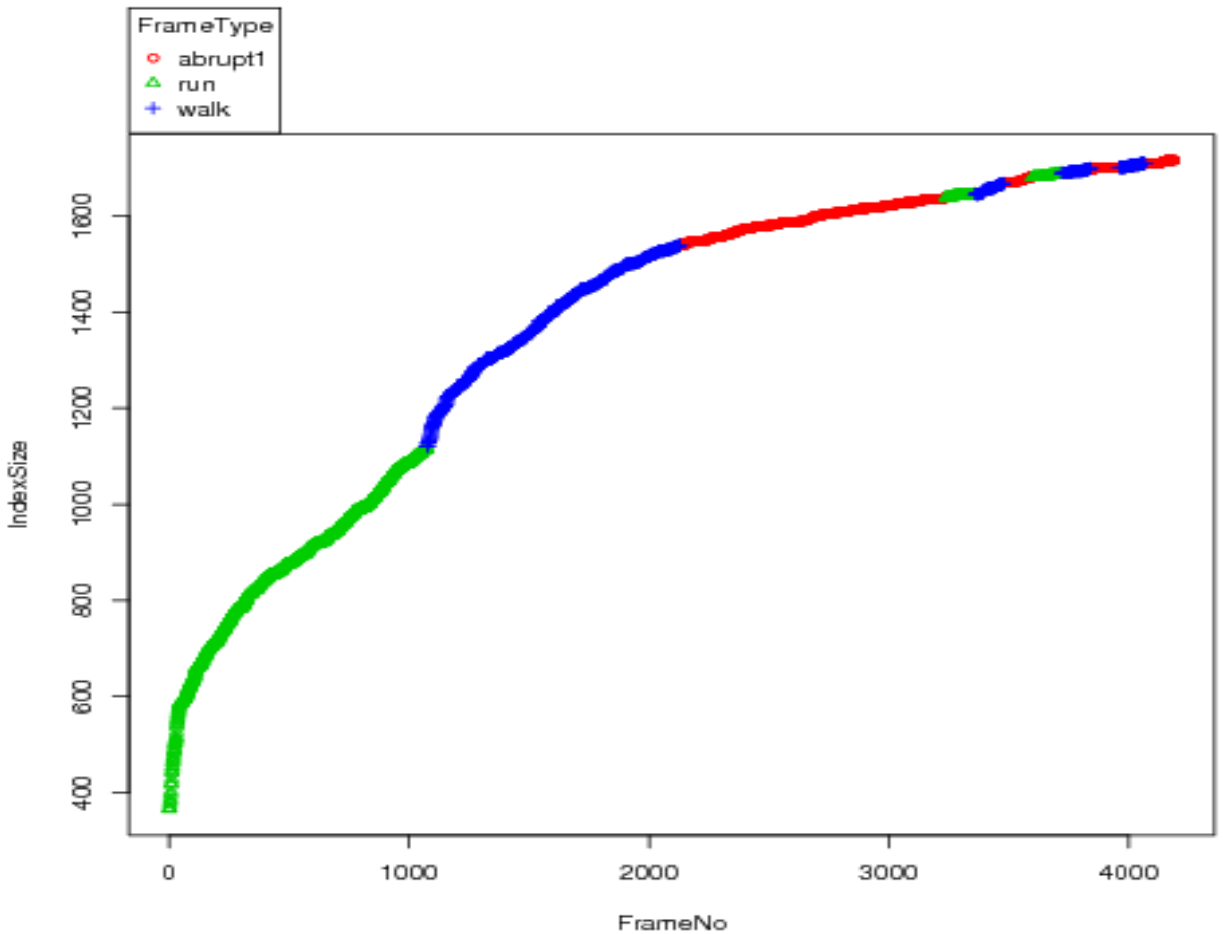
Frame as Symbols Example

1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2	1	1	1	1	1	2	1	1	1	1
1	1	3	3	1	1	1	1	1	1	1	1	3	3	3	1	4	2	1	1	1	1	1	1	2	1	3	1
1	1	3	1	2	1	1	1	1	1	1	1	3	3	1	2	3	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1
1	1	1	1	1	1	2	1	1	2	5	1	1	1	1	6	1	2	1	1	1	1	1	1	6	1	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	3	1	1	1	4	1	1
1	1	2	1	1	1	1	6	1	1	1	3	1	1	1	1	1	1	1	1	1	6	1	1	2	1	1	1
1	1	1	1	1	1	1	1	1	1	1	2	2	5	1	1	2	1	1	1	1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2	6	1	1	1	1	1	1	1	1	1	1	2	1	3	4	1	7	1	1	1
1	1	1	1	1	1	1	1	1	1	1	7	1	5	1	1	1	4	1	1	1	1	7	1	8	1	1	1
1	1	1	1	2	1	1	2	1	1	1	7	1	6	1	1	9	1	1	1	1	1	4	8	4	1	1	2
1	1	1	1	1	1	1	1	1	1	1	1	10	2	10	1	2	11	1	1	1	12	13	14	1	10	1	1
1	1	14	14	1	1	14	2	1	1	1	1	1	1	1	14	2	1	1	1	1	3	14	12	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	12	15	6	1	16	1	1	1	1	1	1	12	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	11	10	10	1	16	10	11	10	1	1	1	1
1	10	1	12	1	1	15	1	10	10	1	1	1	10	10	1	6	17	17	14	14	1	11	11	1	1	1	18
1	1	1	1	10	10	10	1	1	1	1	10	1	1	13	11	13	14	14	1	11	14	13	10	1	1	10	15
1	10	1	14	1	1	1	10	1	1	10	1	1	1	11	10	1	10	1	1	12	11	13	13	1	1	12	1
1	1	10	1	1	10	10	1	10	11	12	12	10	10	15	11	1	11	14	1	1	12	15	19	4	14	18	1
1	1	1	1	1	1	1	10	11	1	1	14	10	11	18	13	14	1	3	20	4	4	15	1	2	2	15	3
1	12	1	1	10	12	11	1	1	1	1	1	1	19	12	20	2	4	13	21	4	12	14	1	1	2	22	15
1	10	1	1	10	1	23	14	1	10	11	1	1	1	1	15	12	14	12	1	1	11	13	1	1	15	20	10
1	1	1	1	1	1	1	17	11	10	1	1	10	1	1	1	10	20	12	15	12	17	24	1	12	25	10	1
1	1	1	1	6	1	1	1	26	1	1	11	1	1	1	1	1	14	1	14	1	10	26	27	28	10	1	1
1	1	1	1	1	1	1	1	1	1	12	1	6	10	1	1	1	1	1	1	1	1	1	26	15	23	1	1
1	1	1	10	12	1	1	1	1	1	14	1	1	1	1	6	1	1	1	1	1	1	1	29	13	1	1	
1	10	1	1	1	1	1	1	1	1	1	1	1	1	1	10	1	1	10	25	1	1	1	12	1	1	1	1
1	1	1	1	1	1	1	1	14	26	26	6	1	10	1	1	1	10	27	10	12	1	1	1	1	10	1	
1	1	1	1	1	1	14	1	1	1	1	14	1	1	10	11	10	1	1	10	12	1	6	1	12	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	14	10	1	6	1	10	1	1	1	10
1	1	1	1	1	1	1	1	1	1	1	1	6	1	1	1	1	12	29	1	1	1	1	10	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	12	12	1	1	10	1	1	1	10	1	10	1	1	1	1	1	1
1	1	1	1	1	1	6	1	1	1	1	1	1	1	1	1	12	1	12	1	1	1	10	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30	12	1	1	1	1	1	1	10	1	1
1	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	12	1	1	1	1	1	10

Symbol Graph Example



Size of Index vs. Frames



Experiments and Inter-class Similarity

Table 1: Before noise reduction inter-class similarity

Category 1	Category 2	Whole Frame Motion Representation Similarity	Symbolic Approach Similarity
run	walk	0,6644	0,4972
run	abrupt	0,6933	0,2001
walk	run	0,6644	0,4972
walk	abrupt	0,6738	0,3333
abrupt	run	0,6933	0,2001
abrupt	walk	0,6738	0,3333

Table 2: After noise reduction inter-class similarity

Category 1	Category 2	Whole Frame Motion Representation Similarity	Symbolic Approach Similarity
run	walk	0,4039	0,4148
run	abrupt	0,4510	0,0522
walk	run	0,2891	0,4148
walk	abrupt	0,5494	0,2073
abrupt	run	0,2829	0,0522
abrupt	walk	0,3709	0,2073

Noise Removal Effect

Table 1: Before noise reduction inter-class similarity

Category 1	Category 2	Whole Frame Motion Representation Similarity	Symbolic Approach Similarity
run	walk	0,6644	0,4972
run	abrupt	0,6933	0,2001
walk	run	0,6644	0,4972
walk	abrupt	0,6738	0,3333
abrupt	run	0,6933	0,2001
abrupt	walk	0,6738	0,3333

Table 2: After noise reduction inter-class similarity

Category 1	Category 2	Whole Frame Motion Representation Similarity	Symbolic Approach Similarity
run	walk	0,4039	0,4148
run	abrupt	0,4510	0,0522
walk	run	0,2891	0,4148
walk	abrupt	0,5494	0,2073
abrupt	run	0,2829	0,0522
abrupt	walk	0,3709	0,2073

Experiments (Semveillance dataset)

Behavior	Precision	Recall	F-measure
run	0.9656	0.7178	0.8231
walk	0.6741	0.9287	0.7746
abrupt	0.9522	0.9298	0.9408

And more...

- Nano-material surfaces: as 2D/3D images
- Genomic sequences: as strings
- Molecular dynamics/chemistry: as 3D shapes
- Other graphs as proximity graphs

To remember

- Many ways to encode information in the graph
- Many existing applications
- Significant challenges regarding graph indexing

Outline

Successes and Failures

- Successful in text analysis (summary evaluation, sentiment analysis, topic detection, ...)
- Successful in user modeling
- Promising in image/video analysis
- Effective in other settings
- Scaling challenges with millions of nodes
- Scaling challenges when combined with other modalities
- Not really useful in as a language model (i.e. sequence generator)

Almost there...

- N-gram Graphs and Operators
- Beyond probabilities
- Domain agnostic
- Multi-modal applicability
- Applicability across settings
- Many interesting research directions

Sneak Peek into Next Steps

- N-gram graph indexing

Sneak Peek into Next Steps

- N-gram graph indexing
- N-gram graphs in multi-type / multi-relational settings / multi-partite settings

Sneak Peek into Next Steps

- N-gram graph indexing
- N-gram graphs in multi-type / multi-relational settings / multi-partite settings
- Non-probabilistic perplexity

A story of non-probabilistic representation across modalities and settings

George Giannakopoulos[1]

[1]NCSR Demokritos
ggianna@iit.demokritos.gr

07/04/2021 - Complex Networks Think Tank

Thank you

References I

Auwatanamongkol, S. (2007).

Inexact graph matching using a genetic algorithm for image recognition.

Pattern Recognition Letters, 28(12):1428–1437.

Chen, Z.-M., Wei, X.-S., Wang, P., and Guo, Y. (2019).

Multi-label image recognition with graph convolutional networks.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5177–5186.

Epasto, A. and Perozzi, B. (2019).

Is a single embedding enough? learning node representations that capture multiple social contexts.

In *The World Wide Web Conference*, pages 394–404.

References II

George Giannakopoulos, George Kiomourtzis, and Vangelis Karkaletsis (2014).

NewSum: "N-Gram Graph"-Based Summarization in the Real World.

In *Innovative Document Summarization Techniques: Revolutionizing Knowledge Understanding*. IGI.

bibtex: george_giannakopoulos_newsum:_2014.

Giannakopoulos, G. (2009).

Automatic Summarization from Multiple Documents.

PhD thesis, Department of Information and Communication Systems Engineering, University of the Aegean, Samos, Greece, http://www.iit.demokritos.gr/~ggianna/thesis.pdf.

Giannakopoulos, G. and Karkaletsis, V. (2009).

N-gram graphs: Representing documents and document sets in summary system evaluation.

In *Proceedings of Text Analysis Conference TAC2009*.

References III

Giannakopoulos, G. and Karkaletsis, V. (2010).

Summarization system evaluation variations based on n-gram graphs.

Gaithersburg, MD, U.S.A.

Giannakopoulos, G., Karkaletsis, V., Vouros, G., and Stamatopoulos, P. (2008).

Summarization system evaluation revisited: N-gram graphs.

ACM Trans. Speech Lang. Process., 5(3):1–39.

Giannakopoulos, G., Mavridi, P., Paliouras, G., Papadakis, G., and Tserpes, K. (2012).

Representation Models for Text Classification: A Comparative Analysis over Three Web Document Types.

In *Proceedings of the 2Nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, pages 13:1–13:12, New York, NY, USA. ACM.

References IV

Hawe, J. S., Theis, F. J., and Heinig, M. (2019).

Inferring interaction networks from multi-omics data.

Frontiers in genetics, 10:535.

Kim, S. M., Pena, M. I., Moll, M., Giannakopoulos, G., Bennett, G. N., and Kavraki, L. E. (2016).

An Evaluation of Different Clustering Methods and Distance Measures Used for Grouping Metabolic Pathways.

In *8th International Conference on Bioinformatics and Computational Biology (BICoB) - To Appear*.

bibtex: kim_evaluation_2016.

Nizamkari, N. S. (2017).

A graph-based trust-enhanced recommender system for service selection in iot.

In *2017 International Conference on Inventive Systems and Control (ICISC)*, pages 1–5. IEEE.

References V

Ostuni, V. C., Di Noia, T., Mirizzi, R., and Di Sciascio, E. (2014).

A linked data recommender system using a neighborhood-based graph kernel.

In *International conference on electronic commerce and web technologies*, pages 89–100. Springer.

Pittaras, N., Giannakopoulos, G., Tsekouras, L., and Varlamis, I. (2018).

Document clustering as a record linkage problem.

In *Proceedings of the ACM Symposium on Document Engineering 2018*, pages 1–4.

References VI

Polychronopoulos, D., Krithara, A., Nikolaou, C., Paliouras, G., Almirantis, Y., and Giannakopoulos, G. (2014).

Analysis and Classification of Constrained DNA Elements with N-gram Graphs and Genomic Signatures.

In Dediu, A.-H., Martín-Vide, C., and Truthe, B., editors, *Algorithms for Computational Biology*, Lecture Notes in Computer Science, pages 220–234. Springer International Publishing.

bibtex: polychronopoulos_analysis_2014.

Rentoumi, V., Giannakopoulos, G., Karkaletsis, V., and Vouros, G. (2009).

Sentiment analysis of figurative language using a word sense disambiguation approach.

Borovets, Bulgaria.

References VII

Tepeli, Y. I., Ünal, A. B., Akdemir, F. M., and Tastan, O. (2020).

Pamogk: A pathway graph kernel based multi-omics approach for patient clustering.

Bioinformatics.

Tsekouras, L., Petasis, G., Giannakopoulos, G., and Kosmopoulos, A. (2020).

Social web observatory: A platform and method for gathering knowledge on entities from different textual sources.

In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2000–2008.

References VIII

Zadeh, A. B., Liang, P. P., Poria, S., Cambria, E., and Morency, L.-P. (2018).

Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph.

In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246.