Latent Associations in Signed Graphs



John Ngan POUGUÉ BIYONG
Lincoln College
University of Oxford

A thesis submitted for
the degree of Doctor of Philosophy in Mathematics

2023

Abstract

Signed graphs model attractive and divisive relations with positive and negative links. While graph connections have historically represented similarity, the presence of negative edges in signed graphs questions the conventional paradigm in network science, and introduces the challenging concept of *antagonistic proximity*: two nodes close in the graph may be functionally distant. A growing body of work has studied the adaptation of conventional methods to understand structural and diffusive patterns in signed graphs.

This thesis investigates the connections explicitly present in signed graphs to infer implicit ones. We seek to uncover latent associations, not only locally but also at the macro-scale, between regions of the graph. We find that the nature of the interactions (positive and negative) and the contexts in which these interactions occur are both important to unveil latent connections.

Our contributions are four-fold. First, we design a novel graph embeddings model informed by edge-attribute and sign-aware random walks. The model is able to infer node positions with respect to contexts they have never engaged in. Second, we show that triadic closure, combined with multi-modal cues, is effective to predict edge signs. Third, we generalise the Louvain algorithm, a popular and efficient method for modularity optimisation in unsigned graphs. Our method is able to increase modularity with a competitive computational speed on real-world signed graphs. Fourth, alongside our methods, we build and open-source three large real-world signed graphs to foster future research. Overall, we propose notable contributions using intuitive principles that can be engineered for further improvements.

Dédié à Tony. PFM.

Acknowledgements

In Spring 2019, having earned a Distinction in the M.Sc. in Mathematical Modelling and Scientific Computing (MMSC), Dr Kathryn Gillow, Course Director, encouraged me to apply for the DPhil.

In an emotional moment, I called my mother to tell her that I was offered the opportunity to do the DPhil, but would not accept. At the time, I felt a duty to relieve our family's financial difficulties. I had been interviewing for big tech companies with the feeling that it was not where I wanted to be but also the firm belief that it was where I had to go. She told me not to worry about her. She would do her part: tighten the belt, and protect me with her prayers, and I should do my part: open the doors to something much bigger for our family.

As part of my Masters degree, I had already been working on a Networks project with Professor Renaud Lambiotte. Renaud had made me feel comfortable from the start and I knew that if I were to do a doctoral program, it had to be with him. After noticing the departmental deadline the night prior, Professors Renaud Lambiotte and J. Doyne Farmer interviewed me on a Monday at 10pm. For some reason, this confirmed to me that I was in the right place. Renaud and Doyne promised to help me secure funding and shared some relevant literature. We finished the call minutes to midnight, right before the department's deadline.

The department granted me an EPSRC scholarship and assigned me the supervisors I had requested. It also approved my request to defer the start of the DPhil to 2020: the school year had been mentally exhausting and I had not had a break since the passing of my uncle Papa during the summer of 2018, just before my MSc course began. In particular, I thank Sandhya Patel, Graduate Studies Administrator, whose support and unspoken understanding were crucial at key moments during my time at the Mathematical Institute. My experience in the department has never been the same since she left.

Renaud understood from the start that I was committed to making the most out of the program. While he could have made his mentorship easier by assigning me a research topic to work on, Renaud let me discover my research path and gave me the space and time to experiment and iterate. In my chaotic, ever-changing quest for sound research directions, he took the time to discover concepts outside his primary research interests in order to support me. He also made possible my collaborations with researchers at Spotify, Pometry, my research visits at the University of Tokyo and my remote work arrangements. The peak of Renaud's mentorship is our generalisation of the Louvain algorithm to signed graphs, presented in the Chapter 5 of this thesis. It has been a privilege to be supervised by Renaud, and I cannot wait for this piece of work to be published. I thank him for making me a better scientist, and for his support and constructive feedback. I will not forget our countless scientific and non-scientific chats over a drink across London and Oxford, and hope for more collaborations in the future.

As a senior academic, Doyne helped me considerably by providing me with collaboration opportunities and assistance navigating administrative matters. Recognising my skills in natural language processing, he identified for me a suitable first DPhil project with researchers in Portland, USA. After its cancellation due to the COVID-19 pandemic, Doyne put me in touch with Valentina Semenova, whose joint work led to the construction and publication of the dataset presented in Chapter 4 of this thesis. I would like to thank Doyne for his lessons on collaboration management, leadership, and his impact on the directions of my thesis.

This thesis and the work published during my time in Oxford are the timeless evidence of an accomplished research excursion. To all my co-authors and collaborators, it was a privilege for me to think and write alongside you. I would particularly like to mention the entire Networks research group, my former collaborators at Twitter (X) who inspired the work presented in Chapter 3, and Professor Ryota Kobayashi whose feedback during my time at the University of Tokyo improved my work on temporal signed graphs (Chapter 5).

I am deeply thankful to Lincoln College for providing me with a positive learning environment and multiple research, well-being, sports, and

financial support programmes.

The DPhil program offered me other exploration opportunities. I had the chance to become a runway and photo model, and an entrepreneur. And a Doctor. This is what a young Black man that has been through it can achieve.

* * *

As my time in Oxford comes to an end, I specifically want to thank Jelani Munroe. What I initially felt to be an unfamiliar, sometimes hostile, environment became a space to invest in personal growth and fulfilment when I met Jelani. I am glad that I started engaging with you while you were joyfully talking to yourself in the middle of that hall in October 2018. You have been a selfless, honest and forward-thinking *consigliere*, my friend.

* * *

I am forever grateful to my family for always looking after me. I think especially about my older sister Clémence Néfertiti who engineered several key decisions in my academic path. Mum, thank you for being the most dedicated person I have ever known, and for teaching me to never aim below the sky. Eventually, I want to express my gratitude to all the friends and mentors who have supported me on this journey we call *life*.

* * *

Contents

1 Introduction **2**

1.1 Background . . . 2

1.2 Plan and Contributions . . . 3

2 Preliminaries **6**

2.1 Graphs . . . 6

2.2 Natural Language Processing . . . 11

3 Learning stance embeddings from signed graphs **13**

3.1 Introduction . . . 13

3.2 Related work . . . 14

3.2.1 Unsigned graph embeddings . . . 14

3.2.2 Signed graph embeddings . . . 15

3.2.3 Skip-Gram with Negative Sampling . . . 16

3.3 Stance Embeddings Model . . . 18

3.3.1 Objective function: learning node & topic embeddings . . . 18

3.3.2 Training data creation . . . 19

3.4 Theory of `SEM` . . . 22

3.4.1 Interpretation of the learnt embeddings . . . 22

3.4.2 Random walks on signed graphs: analytical expression . . . 24

3.4.3 Implicit matrix factorisation . . . 25

3.5 Building signed topical graphs . . . 29

3.5.1 `TwitterSG` . . . 29

3.5.2 `BirdwatchSG` . . . 31

3.5.3 Existing datasets . . . 31

3.6 Learning in real-world graphs . . . 33

3.6.1 Embedding Models . . . 33

3.6.2 Training setup . . . 34

3.6.3 Analysis . . . 35
3.7 Conclusion . . . 40

4 Modeling signed graphs with text **42**
4.1 Introduction . . . 42
4.2 Existing datasets for (dis)agreement detection . . . 43
4.3 `DEBAGREEMENT` . . . 44
4.3.1 Collection . . . 45
4.3.2 Annotation . . . 48
4.3.3 Analysis . . . 50
4.4 Triadic closure and language modeling . . . 53
4.4.1 Evaluating pre-trained language models . . . 54
4.4.2 The power of triadic closure: `siBERT` . . . 55
4.5 Conclusion . . . 56

5 Louvain method for signed graphs **62**
5.1 Introduction . . . 62
5.2 Louvain on signed graphs . . . 63
5.2.1 Approach . . . 63
5.2.2 Caveats . . . 64
5.3 `Signed Louvain`: Louvain for signed graphs . . . 67
5.3.1 Method . . . 67
5.3.2 Experiments . . . 69
5.4 Community evolution in signed graphs . . . 71
5.4.1 Related work . . . 71
5.4.2 Temporal signed graph and modularity . . . 73
5.4.3 Inference . . . 74
5.4.4 Community events . . . 75
5.4.5 Opinion dynamics during the 2020 BLM protests . . . 80
5.5 Conclusion . . . 82

6 Learnings and future directions **83**

Bibliography **86**

List of Figures

2.1 **Triads**. Four triads can form in signed graphs. T_3 and T_1 are strongly balanced, T_0 is weakly balanced, and T_2 is not balanced. 7

2.2 **A random walk on a graph**. A walker randomly walks on the edges of the graph, generating a sequence of nodes it has passed through. . 9

3.1 **Original skip-gram architecture (left) v `SEM` architecture (right)**. `SEM` takes a source node and a topic as input and predicts signed contexts. 19

3.2 **A random walk sampling contexts on a signed graph**. The corresponding sign-informed contexts for source node $u_o = C$ are shown in bold (assuming a window of size 2). 21

3.3 **Equivalence between $\mathcal{G}$ (left) and $\bar{\mathcal{G}}$ (right)**. A signed random walk on $\mathcal{G}$ generates two unsigned random walks on $\bar{\mathcal{G}}$. 26

3.4 **Steps involved in building `TwitterSG`**. 30

3.5 **A sample of `TwitterSG`**. A positive (negative) edge represents a positive (negative) stance of the source node towards the target node for the given topic. 31

3.6 **A sample of `BirdwatchSG`**. A positive (negative) edge represents a positive (negative) stance of the source node towards the target node for the given topic. 32

3.7 **Top 15 topics in `TwitterSG` (top) and `BirdwatchSG` (bottom)** respectively, ordered by decreasing frequency. 33

3.8 **Logistic regression classifier for edge sign prediction** to investigate learning attribute embeddings separately from node embeddings. 37

3.9 **`TwitterSG` topic embeddings learnt by `SEM`**. Related topics are embedded close to each other. Only certain representative topics are labelled for readability. 39

3.10 **`BirdwatchSG` node embeddings learnt by `SEM`**. Two opinion communities are observed in Birdwatch, in keeping with polarisation in the US political public debate. 40

3.11 `BirdwatchSG` **edge embeddings derived from** `SEM` **node embeddings**. Negative (positive) edges are depicted in black (grey). Edges with different signs form distinctive clusters. 41

4.1 **A post and its comments on Reddit**. 45
4.2 **Interface for annotators**. For each labeling task, the subreddit, the initial post (topic), the parent (initial) and child (response) comments are displayed to the annotator. 49
4.3 **Monthly evolution of interactions per subreddit**. + / - / o represent the number of positive, negative and neutral interactions, respectively. 52
4.4 **The role of triadic closure in edge sign predictions**. Full-line edges belong to triangles involving A and B. Dotted-line edges are not used for predictions. 55
4.5 **Instructions for annotators** - summary of task. 59
4.6 **Instructions for annotators** - workflow and rules. 60
4.7 **Instructions for annotators** - key information to look for. 61

5.1 **Performance of Louvain v Relaxed Louvain v** `Signed Louvain` **on a Signed Stochastic Block Model**. The colour indicates the NMI between the final partition and the one used to build the SSBM. As can be seen, the quality of the original Louvain remains low in situations when the density of negative edges between communities is large. . . 66
5.2 **Example of graph where Relaxed Louvain may produce inconsistent communities**, for instance $\{0, 5, 6\}$, due to to its lack of locality during the optimisation. 67
5.3 **Influence of resolution parameters $\gamma^{\pm}$ on community sizes**. The higher γ^{+}, the smaller the communities. The higher γ^{-}, the larger the communities. 70
5.4 **Modularity vs Duration (in seconds)**. L stands for Louvain, SL_d for default `Signed Louvain`, SL_e for extended `Signed Louvain`, and RL for Relaxed Louvain. 72
5.5 **Three scenarios of community dynamics**: a community merger, a community split, and the growth of the graph. 76
5.6 **Influence of tie-decay**. The performance in terms of NMI varies with respect to the choice of the tie-decay parameter η for each of three scenarios. 78

5.7 **Time resolution limit**. Temporal `Signed Louvain` is not able to detect community events which take place faster than its characteristic time $\tau = \frac{1}{\eta}$. The community merger which occurs around $t = 1000$ is undetected for $\eta = 10^{-4}$, and detected for $\eta = 10^{-3}$. 79
5.8 **Temporal trade-off v Instant optimal approach**. The instant optimal version of temporal `Signed Louvain` is unstable compared to its temporal trade-off version, which shows the benefits of smoothing the community evolution through carefully chosen optimisation initialisation. 80
5.9 **Opinion dynamics in r/BLM**. The alluvial diagram depicts the state of opinion communities (gray blocks) at different times. The flows between communities represent the amount of nodes going from one community to the other at the next time step. A non-significant portion of nodes change of opinion (community) between 15 and 30 July 2020. 81

List of Tables

3.1 **Statistics of signed graph datasets**. The bottom datasets `BirdwatchSG`, `TwitterSG` are our contributions. 34

3.2 **Operations (Φ) to produce edge embeddings** from node embeddings for evaluation. 35

3.3 **Mean AUC from 5-fold CV on stance detection using nearest neighbors (kNN) and logistic regression (LR)** on edge embbedings to predict stance (edge sign). 36

3.4 **Mean AUC from 5-fold CV on cold-start stance detection using nearest neighbors (kNN)**. Results remain comparable to Table 3.3 for `SEM`, demonstrating that we can still maintain high accuracy without prior observations of a user's engagement on a topic. 37

3.5 **Mean AUC from 5-fold CV on stance detection** where we learn topic embeddings learnt during link prediction, separately from node embeddings. 38

4.1 **Subreddits**. The description of each subreddit is provided as written on Reddit, along with their member counts as of August 2021. 46

4.2 **Filtering rates at each step of processing the posts**. r/Dems and r/Reps are abbreviations for r/Democrats and r/Republican, respectively. 47

4.3 **Filtering rates at each step of processing the comments**. r/Dems and r/Reps are abbreviations for r/Democrats and r/Republican, respectively. 47

4.4 **Samples of `DEBAGREEMENT`**: an agreement in r/Brexit (top row), a disagreement in r/Brexit (middle row) and a neutral interaction in r/Climate (bottom row). 50

4.5 **Graph characteristics**, split per subreddit. + / - / o represent the number of positive, negative and neutral pairs, respectively. The start date is the first month of interactions in each subreddit dataset. The full agreement rate is the proportion of pairs whose labels are identical for all annotators. 51
4.6 **Triads in** `DEBAGREEMENT`. The (S) indicator stands for the graph resulting from only keeping the signed interactions (i.e. removing the neutral edges). 53
4.7 **Accuracy of large language models (LLMs) for (dis)agreement detection**: standard deviations are in brackets. We use the base, uncased versions of each model. 54
4.8 **Confusion matrix of BERT**. The model is under-performing on neutral interactions. 55
4.9 **Accuracy of BERT v `siBERT`, its combination with triadic closure**. The introduction of triadic closure significantly improves the performance and highlights the benefits of combining structural signals with language understanding approaches. 57

5.1 **List of empirical graphs used for the numerical experiments**, with some of their key structural properties. The graphs were collected from `DEBAGREEMENT`, `TwitterSG` and `BirdwatchSG`. 70

Statement of Originality

The work presented in this thesis is my own unless specified otherwise below. External sources are referenced appropriately.

Chapter 3 I produced all the work presented in this chapter: the literature review, the model design and implementation, and the experiments. Ahmed El-Kishky and Akshay Gupta supervised and helped me to contextualise this project. This work is published and publicly available [116]. Renaud Lambiotte supervised and helped me substantially to approach the theoretical exploration.

Chapter 4 Parts of this chapter are joint work with Valentina Semenova and collaborators at ScaleAI. This work is published and publicly available [117]. A major contribution of this chapter is a dataset, which was created through joint efforts from Valentina and me. We designed the data collection, processing, and annotation processes together. I implemented the NLP experiments jointly with our collaborators at ScaleAI. I designed and produced the analytical results presented on the annotated dataset, and also designed and produced the bi-modal model proposed at the end of the chapter. That being said, the parts of this work that I produced on my own were influenced by fruitful discussions with Valentina Semenova and Renaud Lambiotte.

Chapter 5 Renaud Lambiotte supervised this work and helped me to conceptualise the model proposed. The experiments on the temporal graphs were partly inspired by the analysis performed on the dataset discussed in the previous chapter.

Chapter 1

Introduction

1.1 Background

A *graph* - or *network* - is, in its simplest form, a collection of points joined together in pairs by lines [102]. The points are referred to as *nodes* and the lines are referred to as *edges*. *Network science* is the science of developing mathematical, computational, and statistical tools to model, analyse, and understand networks.

Many objects in the physical, biological, and social sciences can be thought of as networks. Biological networks represent complex systems of interactions between biological entities, such as protein-protein interactions [113, 123], DNA-protein interactions [152], or neuronal interactions in the brain [18]. Ecological networks are representations of biotic interactions in an ecosystem in which species are connected by pairwise interactions [67], capturing interactions between flowering plants and animal pollinators [94] for instance. Other examples include telecommunications, electrical, supply, social, transport, urban and economic networks among others [86].

Network scientists have been studying major questions such as network structures, diffusion in networks, and dynamics of networks. Understanding the complex shapes formed by connected nodes is crucial to understand the micro- and macro-structure and function of networks [104]. Diffusion (e.g. virus epidemics) is concerned with the propagation of quantities over the network via its edges. Diffusion processes are naturally dependent on the network structure [77]. Network dynamics, on the other hand, relate to the temporal evolution of a network [55].

Historically, connections and proximity in a network have been considered as indicators of similarity, attachment which fosters diffusion [40, 93, 106]: a highway increases mobility between two large cities (nodes); the diffusion of ideas among a group of friends is more likely than among a group of strangers; a nerve conveys information from one region of the body to the other.

However, signed graphs (or networks), at the center of this thesis, are graphs which model attractive (positive edge) and antagonistic (negative edge) relations [49]. The presence of negative edges in signed graphs questions the conventional paradigm in network science, and introduces the challenging concept of *antagonistic proximity*: two close nodes in the network may be functionally very distant. As a consequence, structures, diffusion and dynamics must be studied under a different lens, and different principles which account both for cohesive and divisive connections must be considered [145, 57, 36, 74, 75]. Signed graphs are used to model social systems where relationships can be cooperative or adversarial [140, 83], as well as ecological systems where species may exhibit symbiotic and competitive interactions [139], or synchrony and anti-synchrony between the activity patterns of different regions of the brain [129].

This thesis studies the node interactions explicitly present in signed graphs to infer other implicit connections. We seek to uncover latent structures not only at the local scale (between nodes or edges in the same region of the graph), but also latent associations (or divisions) at the macro-scale, between regions of the graph. We will see that the keys to unveil the latent patterns are not only the nature of the connections (positive or negative), but also the importance of the contexts in which these connections occur.

1.2 Plan and Contributions

Chapter 2: Preliminaries

In the preliminary chapter, we present foundational concepts used in this thesis, including structural balance, triads, signed stochastic block models, normalised mutual information, random walks, graph embeddings, stance detection and large language models.

Chapter 3: Learning stance embeddings from signed graphs

In this chapter, we study the question of contextual interactions. Graph interactions are complex and can change depending on underlying context. A relevant example is two individuals supporting each other on a topic, but disagreeing on another topic. We model such configurations as edge-attributed signed graphs, where the edge attributes are interchangeably called topics. One important challenge is to exploit the position of a node in observed contexts to infer the position of this node in contexts they have not engaged in. To address this challenge, we propose the Stance Embeddings

Model (SEM), a scalable, efficient, and interpretable embeddings method to learn node and topic representations and similarities. Besides outperforming existing methods, SEM has the unique advantage of being auditable conceptually: we show that the model implicitly factorises a matrix capturing the structural and spectral properties of the graph. We test our approach on TwitterSG and BirdwatchSG, two novel context-aware signed graphs that we open-source to foster future research in network science. TwitterSG is 6x larger than the largest publicly available signed social graph.

Parts of this chapter have been published and received the Best Paper Award of the 16th ACM International Conference on Web Search and Data Mining (WSDM), held in Singapore in February 2023: *Pougué-Biyong, John, et al. "Learning stance embeddings from signed social graphs." Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. 2023* [116].

Chapter 4: Modeling signed graphs with text

After recognising the importance of developing context-aware methods in the previous chapter, this chapter addresses a second issue in signed graph modeling: the shortage of open-source graphs containing context-aware edges. This chapter aims to supply researchers with signed graphs that connect directly to the real-world mechanisms they represent. We do this by collecting multi-modal data to create a context-aware graph dataset. We design, build, and analyse DEBAGREEMENT, a signed graph with text data collected from Reddit discussions. In DEBAGREEMENT, the edges contain the textual exchanges of the discussion. A positive (resp. negative) edge represents an agreement (resp. disagreement) between the two nodes. This text can be consolidated with the graph structure for a multitude of downstream tasks such as predicting the sign of unobserved edges. This chapter bridges the gap between two fields of research which can considerably benefit from each other: edge sign prediction in network science, and (dis)agreement detection in natural language processing. Indeed, beyond the originality of its design, the dataset unlocks the unique opportunity to combine graph structural concepts with recent advances in natural language processing in order to better capture unobserved interactions. Through structural principles, large language models, extensive analysis and experiments, this chapter demonstrates how powerful it is to combine structural knowledge and language understanding to unveil latent associations between nodes - the signs of the edges in this case. As a result, DEBAGREEMENT's impact transcends network science - it becomes a tool to also investigate social theories, online polarisation, and study how people express opinions and change their stance on social media.

Parts of this chapter have been published at the 35th Conference on Neural Information Processing Systems (NeurIPS) held online in December 2021: *Pougué-Biyong, John, Valentina Semenova, et al. "DEBAGREEMENT: A comment-reply dataset for (dis) agreement detection in online debates." Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). 2021* [117].

Chapter 5: Louvain method for signed graphs

In this method chapter, we complete our research by studying unobserved associations at the macro scale, through community detection in signed graphs. We study generalisations of the Louvain method [15], a popular and efficient method for modularity optimisation in unsigned graphs. We propose a new heuristics-based method, `Signed Louvain`, and expand the heuristics to temporal signed graphs. The heuristics benefit from the scalability and speed advantages of the Louvain method, and overcomes its shortcomings on signed graphs. Through extensive synthetic experiments, we show the advantages and limitations of our proposed method. As a use case, we apply our model to unveil opinion dynamics in a real-world debate graph from `DEBAGREEMENT`.

This thesis traces the journey of an early-career network scientist, starting from systematic literature reviews, through the formulation of research questions, the design of methods, data collections, experimental implementations, and up to critical analysis.

Chapter 2

Preliminaries

2.1 Graphs

A graph $\mathcal{G}$, or network, is a set of nodes connected by edges [102]. The weight of an edge represents the strength of the link between the two nodes it connects. While traditional graphs are usually assumed to only contain positive edges, a *signed graph* is a graph whose edges are either positive or negative[1]. In that case, a positive edge weight represents *attachment*, *attraction*, and a negative edge weight represents *antagonism*, *opposition*. A signed graph is said to be *unweighted* if the weights of its edges are either -1 (if the edge is negative) or $+1$ (if the edge is positive). Otherwise, it is weighted and the absolute value of the weight describes the intensity of the attachment or the opposition. A signed graph is characterised by an *adjacency matrix* $A \in \mathbb{R}^{n \times n}$, where $n = |\mathcal{G}|$ is the number of nodes in $\mathcal{G}$. For nodes i and $j \in \mathcal{G}$, $A_{ij} > 0$ (< 0) indicates the presence of a positive (negative) edge between i and j. A graph is said to be *undirected* if $\forall i, j,\ A_{ij} = A_{ji}$, i.e. an edge from i to j exists if and only if the same edge exists from j to i. We assume that all graphs studied in this thesis are undirected. We decompose $A = A^+ - A^-$, where A^+ (A^-) is the adjacency matrix of A's positive (negative) part [58, 75]. We also note $|A| = A^+ + A^-$ such that $|A|_{ij} = |A_{ij}|$.

Structural properties

In unsigned graphs, connections are often driven by the principle of homophily, the sociological theory that connections occur between similar nodes (e.g. individuals) [93, 111]. In signed graphs, however, the presence of negative edges challenges this principle as two negatively connected nodes have an antagonistic relationship.

[1] By contrast, an unsigned graph refers to a graph whose all edge weights are positive.

Such divisive relations play a pivotal role on the structural properties of signed graphs. While unsigned graphs are governed by homophily, signed graphs are studied through the lens of other social theories. Social balance theory and triadic closure have their origins in cognitive dissonance theory from the early 20th century [133, 52]. They speculate that (A) *the friend of my friend is my friend*, (B) *the enemy of my friend is my enemy*, and (C) *the enemy of my enemy is my friend*. These claims translate into network science principles which fall under the term *structural balance*. To provide an intuition, we consider the three-node graph structures displayed in Figure 2.1, and called *triads* or *triangles*. We denote T_k the triad with k positive edges. The structural balance principle states that T_3 and T_1 are strongly balanced, T_0 is weakly balanced, and T_2 is not balanced [83]. Drawing a parallel with social balance theories, T_2 is not balanced because it represents a state in which an individual is a friend of two individuals who are enemies. By contrast, triads T_3 and T_1, which contain an even number of negative edges, are strongly balanced: they represent the claims (A) and (B) respectively. T_0 is weakly balanced - it depicts a three-side situation where *the enemy of my enemy is also my enemy*.

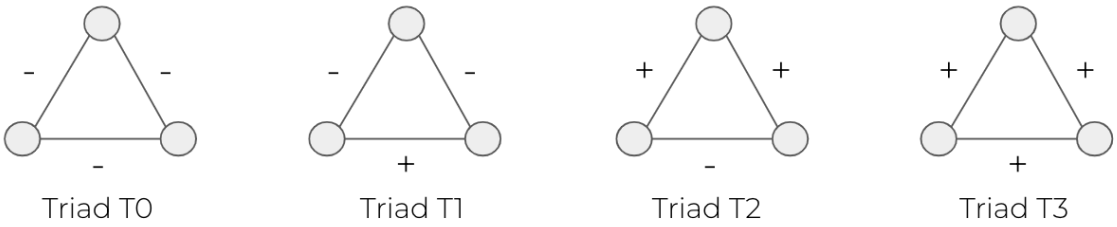


Figure 2.1: **Triads**. Four triads can form in signed graphs. T_3 and T_1 are strongly balanced, T_0 is weakly balanced, and T_2 is not balanced.

Structural balance is a principle that applies not only to local structures like triads, but also to the global structure of a graph. A graph is defined as k-balanced (or k-clusterable) if it can be decomposed into k *clusters* - or *communities* - where positive edges are inside the clusters and negative edges are across the clusters. K-clusterable graphs are known to exhibit the weak version of structural balance [146, 143] (when $k \in \{1, 2\}$, they exhibit the strong version). For instance in Figure 2.1, T_3 is 1-clusterable (the three nodes are friends so part of the same group), T_1 is 2-clusterable (two nodes are friends, and enemies to the same node), T_0 is 3-clusterable (each node is on its own), and T_2 is not clusterable. In real settings, graphs such as social networks are rarely fully k-balanced and there are some edges that make the graph

unbalanced. The minimum number of such edges (negative edges within communities and positive edges across communities) is called *frustration* [6].

Generative model

A standard model to generate near-balanced signed, unweighted graphs is the Signed Stochastic Block Model (SSBM) [54, 134, 87]. Consider a number n of nodes and a partition of the nodes into k clusters $\{C_1, ..., C_k\}$ of potentially different sizes. In a simplified form [62], a SSBM randomly generates edges inside and between clusters according to four parameters p_{in}, p_{out}, $p_{fr,in}$ and $p_{fr,out}$. p_{in} (resp. p_{out}) is the density of edges inside (resp. between) communities. For instance, if $p_{in} = 1$, all pairs of nodes inside the same community are connected with an edge. Conversely, if $p_{out} = 1$, any node is connected to all nodes outside its community. In keeping with the definition of k-clusterable structures above, in-cluster edges are positive and edges between clusters are negative by default. Recognising the fact that real graphs are rarely k-balanced, $p_{fr,in}$ and $p_{fr,out}$ are parameters controlling the amount of frustration (relatively to the planted partition) inside and between communities. In particular, $p_{fr,in}$ is the share of negative edges among the existing in-community edges, and $p_{fr,out}$ is the share of positive edges among the existing between-community edges. For example, in a community of 10 nodes, $\frac{11\times10}{2} = 55$ edges can be formed in total (each of the 11 nodes can connect to the 10 other nodes, but each formed edge is bidirectional so is counted twice). With $p_{in} = 80\%$, $80\% \times 55 = 44$ edges are randomly selected and created as positive edges among the pool of 55 edge candidates. Then, with $p_{fr,in} = 25\%$, $25\% \times 44 = 11$ edges are sampled among the 44 created edges, and their sign is swapped to being negative. Overall, p_{in} and p_{out} control the density of the generated graph, defined as the ratio between the number of existing edges and the total number of possible edges:

$$D = \frac{m}{\frac{n(n-1)}{2}} = \frac{2m}{n(n-1)}$$

where m is the number of edges in the graph, and n is the number of nodes.

Signed Stochastic Block Models are particularly useful in generating random graphs on which community detection methods can be evaluated. Community detection is the task of finding clusters in a graph, and is critical for understanding structures [43, 131] and diffusion [77] in complex systems. The standard metric to evaluate the quality of the clusters unveiled by a community detection method is the

Normalised Mutual Information (NMI) [136, 43]. It quantifies how close the communities found by a community detection algorithm are to the ground-truth communities, or clusters[2]. The NMI ranges between zero and one, with a score of one signifying a perfect match of partitions. .

Diffusion on graphs

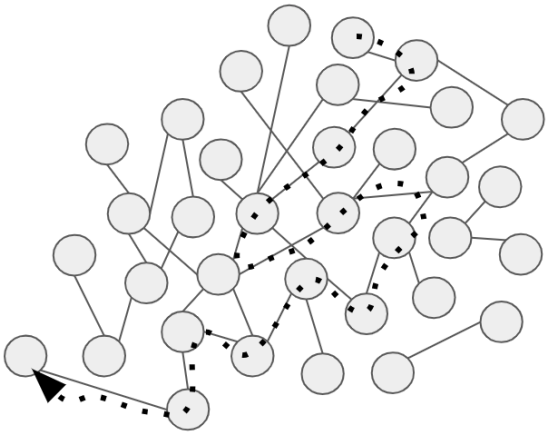


Figure 2.2: **A random walk on a graph**. A walker randomly walks on the edges of the graph, generating a sequence of nodes it has passed through.

Diffusion processes on graphs find applications into the modelling of epidemics, urban planning, survey respondent-driven sampling or interactions between animals, among others. In particular, *Markovian* - or *memoryless* - random walks are among the most fundamental types of stochastic processes [105].

For the sake of simplicity, we assume that the graph is unsigned in this section. Several random walkers start at different nodes on the graph. At each time step, they randomly select a connecting edge to walk on to reach a neighbouring node. The likelihood of walking on a given edge is proportional to the weight of this edge, relatively to the other edges originating at the departure node. We denote $n_{i,0}$ the initial number of walkers at node i. At any step r, $n_{j,r}$ the number of walkers standing on node j is given recursively by:

$$n_{j,r+1} = \sum_i \frac{1}{k_i} A_{ij} n_{i,r}$$

[2]Note that the *ground truth* is a ill-defined concept as the ground truth is not always an optimal partition, especially in real-world graphs [112].

where the *(weighted) degree* $k_i = \sum_l A_{il}$ is the sum of the weights of edges incident to node i. In the vector form with $\mathbf{n_r} = (n_{1,r}, ..., n_{|\mathcal{G}|,r})$:

$$\mathbf{n_{r+1}} = \mathbf{n_r} D^{-1} A = \mathbf{n_r} T$$

where D is the diagonal matrix whose diagonal entries are the weighted degrees k_i of the nodes in the graph. $T = D^{-1}A$ is called the *transition matrix* of $\mathcal{G}$ and its entry T_{ij} is the probability for a walker to move from node i to node j. Because T is row-normalised, 1 is one of its eigenvalues, and we can show that the process described in the above equation converges towards a stationary state $\mathbf{n}$ which satisfies [92]:

$$\mathbf{n}(I - T) = \mathbf{n}(I - D^{-1}A) = 0$$

$I - D^{-1}A$ is the *random-walk laplacian* of graph $\mathcal{G}$ and governs the convergence speed and diffusion patterns of the walks on the graph. The stationary state is the normalised vector of the weighted degrees of the nodes $(k_1, ..., k_{|\mathcal{G}|})$. The spectral analysis of the random-walk laplacian has been the subject of a substantial body of literature (see, e.g., [77, 92]). The process presented above is memoryless, in the sense that the walkers make decisions about their future only based on the nodes they are currently standing on.

Other modeling choices include biased walks where the walker makes decision based on the two (or more) last nodes it has passed through [63], or random walks with restart in which the walker has a non-zero probability to teleport back to its initial node [144]. A popular application of the random walk with restart is the Google's PageRank algorithm [108]. In this thesis, we will present a random walk process on signed graphs.

Community detection

Community detection is crucial for understanding structures [43, 131] and diffusion [77] in complex systems, and has been extensively studied for unsigned graphs [44, 2, 84]. In unsigned graphs, it involves the identification of densely connected groups of nodes, often called *communities* or *clusters*. In signed graphs, however, communities are dense groups of nodes both positively connected to each other and negatively connected to the rest of the network. The presence of negative edges challenges conventional community detection methods as they must not only discern cohesive groups of nodes but also account for the divisive relations at play [145]. We will see in this thesis that the divisive relations play a pivotal role in the comprehensive understanding of signed community structures, and their evolution.

Graph embeddings

Graph embeddings denote the set of methods that learn vector representations of graph components such as nodes or edges [48, 20]. The adoption of such methods has increased in recent years as they are notably more scalable than traditional graph approaches. Additionally, the learnt vectors - or embeddings - can be used to perform an abundance of downstream tasks, such as node classification, edge, or sign prediction—in contrast to traditional graph approaches which are often designed specifically for a given task [20]. In the literature, a distinction is made between shallow embeddings which learn representations from the adjacency matrix of the graph, and graph representation learning methods that leverage node or edge features to learn in an inductive manner. Graph embedding methods are particularly hard to audit and their main limitation remains their lack of interpretability [19]. In this thesis, we overcome this limitation and reconcile signed graph embeddings and matrix factorisation approaches.

2.2 Natural Language Processing

A variety of real-world graphs like social networks are driven by textual exchanges. This section provides a brief introduction to some of the tools developed in the field of natural language processing (NLP) for language understanding. Natural language processing is a subfield of linguistics, computer science, information engineering and artificial intelligence, concerned with the processing and analysis of large amounts of natural language data [25]. Common challenges in NLP involve building language models that can translate, analyse sentiments, classify documents, understand or generate text. In 2013, word embedding methods such as Word2Vec [98] or GLoVe [114] were able to capture word associations by learning equivalences between words that would appear in the same contexts. For instance, such models could understand that the relation between the words *king* and *man*, is equivalent to the relation between the words *queen* and *woman*. Recent state-of-the-art advancements like the Transformer [151] have proven significantly more efficient on such tasks, by employing *attention mechanisms*. Introduced in 2015 [9], *attention* is a mechanism that was initially designed to enable a model to focus on specific words when translating a sentence. In the sentence *The animal did not cross the street because it was too tired*, a good reader would realise that the word *it* refers to *the animal* and not to *the street*. When attempting to translate the word *it*, the Transformer learns to pay more attention to *the animal* to translate accurately. A notable attention-based model

is BERT [37] published in 2018, and whose influence has sparked further research leading to the development of systems like ChatGPT [107].

An NLP task of particular interest in signed graph research is *(dis)agreement detection*. (Dis)agreement detection falls under the field of *stance detection* [153], the automatic classification of the position of a targeted piece of text into one of three classes: in favor, against, or neutral. (Dis)agreement detection is fundamental in the study of societal polarisation and the spread of ideas online [125, 156, 141, 126], but more generally to understand attachment and opposition in text-based interactions occurring in graphs.

Chapter 3

Learning stance embeddings from signed graphs

3.1 Introduction

As discussed in Chapter 2), traditional signed graph modelling reduces the interaction between any two nodes to a binary value of attachment (+) or antogonism (−). However, this approach is limiting, since interactions may be much more complex and change depending on underlying context. Most interaction groups will have several different aspects of discourse that have rich structure and dynamics. We use the term *topics* to refer to these different aspects and contexts of discourse represented as graph interactions. For example, two senators may agree on bills related to climate change, but differ on taxation policy bills. Or, in a sports community, two Cameroonian football fans may support rival local clubs, but will generally both support the national team at the World Cup. These examples highlight the value of modeling *relative* stances across a range of topics [99] in order to determine latent or implicit associations between nodes without supervision, which becomes even more important at scale.

Edge-attributed signed graphs are the graph model adapted to represent attachment and opposition across several contexts [102]. We call them *signed topical graphs* as they represent signed interactions across topics of discourse in a graph: each edge represents a binary agreement value ($\{+, -\}$) with respect to a single topic t; the inventory of topics $\mathcal{T}$ is assumed to be fixed and finite, but varies across applications.

In this chapter, we consider learning latent associations between nodes and topics in signed topical graphs. For this purpose, we employ a graph embeddings approach (see Preliminaries, section 2.1). Graph embeddings are particularly interesting because of their scalability [48, 20], and their ability to be used to solve to a plethora

of downstream tasks [20]. However, they are criticised for their lack interpretability since it is most often impossible to determine an analytical expression of the learnt embeddings [19]. This stems from the fact that graph embeddings methods generally optimise untractable cost functions.

Our proposed method, the Stance Embeddings Model (`SEM`), is scalable, efficient, and interpretable. It leverages the skipgram with negative sampling (SGNS) model [98] on signed topical graphs to learn embeddings for nodes as well as for topics. Learning member (node) and topic embeddings jointly enable us to represent topic-informed stance embeddings for each member. The name of the model draws inspiration from the information captured in the embeddings it learns: in essence, the model not only learns relative positions of nodes between each other but also in relation to a topic; as a consequence, the model represents the position - or stance - of a node conditioned on a given topic, and is able to infer the position of a node on a topic they have never engaged with. We audit `SEM` theoretically and show that the model implicitly factorises a matrix which captures the structural properties of the signed graph, and that it also learns topic similarities in an unsupervised fashion. In order to evaluate our approach, we build two Twitter-based signed topical graphs, `TwitterSG` and `BirdwatchSG`, that we open-source alongside our model. `TwitterSG` is $\sim$6x larger than the Epinions graph which, to the best of our knowledge, is the largest publicly available signed social graph. In our experiments, we show that `SEM` is able to learn accurate embeddings compared to competitive topic-agnostic baselines. We display and provide comprehensive analysis of the learnt user and topic embeddings.

3.2 Related work

This chapter falls in the domain of shallow, signed and edge-attributed graph embeddings. We review existing unsigned and signed graph embeddings methods in sections 3.2.1 and 3.2.2 respectively. In section 3.2.3, we review the skip-gram with negative sampling [98] as a preliminary step to introduce our method.

3.2.1 Unsigned graph embeddings

Unsigned graph embeddings methods consist of learning embeddings in unsigned graphs [48]. The most popular methods include node2vec [46], deepwalk [115], and LINE [142]. Node2vec and deepwalk build on top of word2vec [97], a word embedding technique in natural language processing. Node2vec generates second-order random walks on unsigned graphs, and learns node embeddings by training a skip-gram with

negative sampling (SGNS) [98] to predict the surroundings of the input node. The learnt embeddings are such that nodes close in the graph are close in the embedding space. Deepwalk is a specific case of node2vec as it generates unbiaised random walks and follows the same subsequent process. LINE is implicitly a special case of deepwalk [120]. Such methods are not adapted for signed graphs because they are based on the homophily assumption (that connected nodes should lie close in the embedding space), whereas in signed graphs positively connected nodes should be closer while opposing nodes farther apart—even if these nodes are connected.

3.2.2 Signed graph embeddings

We report three types of signed graph embeddings approaches in the literature: spectral, random walk, and deep learning methods. The earliest approaches to learn signed graph embeddings are spectral [75, 73, 164, 165]. Spectral methods consist of computing some low-rank matrix factorisation of the signed graph laplacian and/or studying its spectrum.

Random walk methods overcome the homophily limitation encountered in signed graphs by defining random walks on signed graphs. SNE [162] generates uniform random walks by ignoring the edge signs, and replaces the skip-gram model by a log-bilinear model. The model predicts the representation of a target node given its predecessors along a path. To capture the signed relationships between nodes, two signed-type vectors are incorporated into the log-bilinear model. SIDE [65] generates first-order random walks and defines a likelihood function composed of a signed proximity term to model social balance, and two bias terms to mimic preferential attachment [103]. Similarly, SIGNet [60] designs a new sampling technique for directed signed graphs, and optimises an objective function that learns two representations for each node. BESIDE [23] mathematically models *bridge* edges based on balance and status theory.

Among deep learning approaches, SiNE [154] is a deep neural network-based model guided by social theories. It maximises the margin between the embedding similarity of *friends* and the embedding similarity of *foes*. StEM [122] is a deep learning method aiming at learning not only representations of nodes of different classes (e.g. friends and foes) but also decision boundaries between opposing groups. Hence, unlike other methods, e.g. SiNE, which are distance-based (thus, use only local information), StEM attempts to incorporate global information. Instead of relying on social theories, ROSE [61] learns role similarities between nodes.

Overall, no previous work on signed graph embeddings has investigated the importance of edge attributes. In order to do so, our approach extends the skip-gram model of node2vec to signed topical graphs. We facilitate this by ensuring that each training example is constructed via a topic- and sign-informed random walk. Leveraging the skip-gram architecture not only provides scalability advantages, but also the flexibility to adapt the embedding process to effectively utilise edge attributes. This approach also enables us to provide a theoretical interpretation to our results. While all aforementioned related works are edge-attribute agnostic, our proposed method can leverage edge attributes such as topics in the form of contextual topic embeddings. We argue that incorporating edge attributes - such as topics in social graph interactions - in the embedding process can improve understanding of latent associations.

3.2.3 Skip-Gram with Negative Sampling

The Skip-Gram with Negative Sampling model (SGNS) is a model introduced in *word2vec* [98] to efficiently learn word embeddings from a corpus of text. In this model, a corpus of words w and their contexts c (i.e. their surrounding words in the text) is given. B denotes the set of all word and context pairs extracted from the text. For any pair (w, c) of word and context, $\mathbb{P}(B = 1 \mid w, c)$ denotes the probability that (w, c) came from the corpus data B. We assume that there are parameters θ controlling the distribution: $\mathbb{P}(B = 1 \mid w, c; \theta)$. Given a set B' of (w, c) pairs which do not appear in the observed data B, also called *negative samples*, the goal is to set the parameters θ so as to maximize the objective:

$$\begin{aligned}
\mathcal{L} &= \arg\max_{\theta} \prod_{(w,c)\in B} \mathbb{P}(B = 1 \mid w, c; \theta) \prod_{(w,c)\in B'} \mathbb{P}(B = 0 \mid w, c; \theta) \\
&= \arg\max_{\theta} \prod_{(w,c)\in B} \mathbb{P}(B = 1 \mid w, c; \theta) \prod_{(w,c)\in B'} (1 - \mathbb{P}(B = 1 \mid w, c; \theta)) \\
&= \arg\max_{\theta} \sum_{(w,c)\in B} \log \mathbb{P}(B = 1 \mid w, c; \theta) + \sum_{(w,c)\in B'} \log(1 - \mathbb{P}(B = 1 \mid w, c; \theta))
\end{aligned}$$

The skip-gram with negative sampling model is parametrised by modelling the conditional probability using a softmax function. The probability that a given pair (w, c) came from B is modelled as:

$$\mathbb{P}(B = 1 \mid w, c; \theta) = \frac{1}{1 + e^{-\vec{w} \cdot \vec{c}}},$$

where $\overrightarrow{w}$ and $\overrightarrow{c}$ are the vector representations of the word w and the context c. This leads to the objective:

$$\arg\max_{\theta} \sum_{(w,c)\in B} \log \sigma(\overrightarrow{c} \cdot \overrightarrow{w}) + \sum_{(w,c)\in B'} \log \sigma(-\overrightarrow{c} \cdot \overrightarrow{w}) \tag{3.1}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$, $\overrightarrow{w}$ and $\overrightarrow{c}$ are vector representations for w and c respectively. The parameters to optimise are the embeddings, i.e. the entries of the vectors $\overrightarrow{w} = ((v_w)_1, ..., (v_w)_d)$ and $\overrightarrow{c} = ((v_c)_1, ..., (v_c)_d)$ where d is the dimension of the embedding space. In practice, for each positive example (w, c), b negative examples $(w, c_j) \in B'$ are sampled[1] [45, 98].

Figure 3.1 (left) depicts the original architecture of the skip-gram model. Given an input, it predicts the surrounding words, which we refer to as the *input's context*. The embedding weights W_u and W_c are optimised to maximise the objective function in Equation (3.1). The original authors first showed that words which appear often in the same context are embedded close in the vector space [97, 98]. Following this observation, in an attempt to provide a theoretical understanding to the efficiency of the skip-gram model with negative sampling, it has been shown [85] that the optimal parameters $\overrightarrow{u}$ and $\overrightarrow{c}$ in Equation (3.1) are such that:

$$\overrightarrow{u} \cdot \overrightarrow{c} = \log\left(\frac{\#(u,c)|B|}{\#(u)\#(c)}\right) - \log b \tag{3.2}$$

where b is the number of negative samples; $\#(w, c)$, $\#(w)$, and $\#(c)$ denote the number of times (w, c), w, and c appear in B respectively. The expression $\log\left(\frac{\#(w,c)|B|}{\#(w)\#(c)}\right)$ is the pointwise mutual information (PMI) [26] of (w, c)[2]. In other words, the SNGS implicitly factorises the matrix $M = W \cdot C^\top$ [120]:

$$M_{ij} = W_i \cdot C_j = \overrightarrow{w_i} \cdot \overrightarrow{c_j} = PMI(w_i, c_j) - \log b \tag{3.3}$$

[1] Negative sampling approximates the set B' by sampling a finite set of (potentially) false targets for word w. See [98], Section 2.2.

[2] The PMI of a pair of outcomes x and y belonging to discrete random variables X and Y quantifies the discrepancy between the probability of their coincidence and their individual distributions, assuming independence [26]:

$$PMI(x; y) = \log \frac{p(x, y)}{p(x)p(y)}$$

The measure can take positive or negative values. The PMI equals zero if X and Y are independent. The lower bound is reached when the outcomes are mutually exclusive and the upper bound is achieved when the outcomes are perfectly associated (i.e. $p(x|y)$ or $p(y|x) = 1$:

$$-\infty \leq PMI(x; y) \leq \min[-\log p(x), -\log p(y)]$$

Deepwalk [115] subsequently used the skip-gram with negative sampling model to learn node embeddings in unsigned graphs. The node pairs in the training data are generated using markovian random walks on the graphs. A node is in the context of another node if they surround each other in the random walk sequence. Two nodes which are close in the graph appear to be close in the vector space. Node2vec [46] further showed that using second-order random walks enables to capture homophily and structural equivalence too.

3.3 Stance Embeddings Model

We present `SEM` in this section. We denote $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a signed topical graph. Each edge has a topic t, and a sign $-$ or $+$. We use $\mathcal{T}$ to denote the finite set of topics t. There can be multiple edges corresponding to different topical interactions between two nodes. We define $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ as the sub-graph of $\mathcal{G}$ which contains all the nodes that engage with topic t ($\mathcal{V}_t \subseteq \mathcal{V}$) and all the edges with topic t ($\mathcal{E}_t \subset \mathcal{E}$). We aim to learn a node mapping function $f_{\mathcal{V}} : \mathcal{V} \to \mathbb{R}^d$, and a topic embedding function $f_{\mathcal{T}} : \mathcal{T} \to \mathbb{R}^d$, which map the nodes and the topics in the same embedding space of dimension d. To do so, we build upon previous research on SGNS and node embeddings. `SEM` differs from previous random walk-based work (see section 3.2.1) by feeding the data samples (u, v, s, t) informed by an interaction topic t and a sign s into the skip-gram with negative sampling model (see section 3.3.1). The sign between two nodes u and v is inferred by social balance theory, and represents agreements $(+)$ or disagreements $(-)$ between the two nodes on a given topic (see section 3.3.2). Ideally, our model should embed agreeing nodes close to each other, and push apart nodes who disagree, even though they are neighbours in the graph.

3.3.1 Objective function: learning node & topic embeddings

The training examples are composed of a source node u, a topic t, and a set of contexts $C_t(u)$ where contexts consist of *(node, sign)* pairs. We associate embedding vectors W_u, W_c, and W_t for the source, context (node-sign pair), and topics respectively; these vectors are parameters to be learned. In Figure 3.1 (right), we visualize our proposed architecture: a topic-aware skip-gram model. The goal is to condition the skip-gram predictions with the topic of the interactions. In other words: for a given input node, we want to predict its surrounding nodes and the nature (sign) of the relationship with these nodes. Because the nature of an interaction depends on its context, the topic is provided as input in order to condition the predictions to the topic. On top

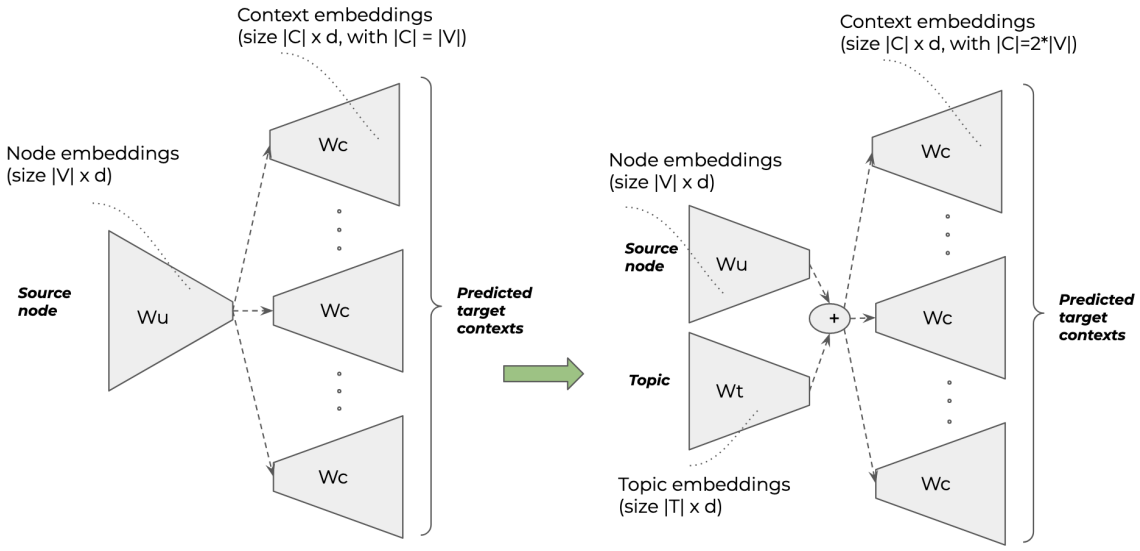


Figure 3.1: **Original skip-gram architecture (left) v `SEM` architecture (right).** `SEM` takes a source node and a topic as input and predicts signed contexts.

of enabling the model to learn both node and topic embeddings simultaneously, we will see theoretically and experimentally that this design permits it to capture topic associations in an unsupervised fashion. Compared to Equation (3.1), our modified skip-gram objective reads as follows:

$$\arg\max_{\theta} \sum_{\substack{(u,c,t)\in B \\ c=(v,s)}} \log \sigma(\overrightarrow{c} \cdot (\overrightarrow{u} + \overrightarrow{t})) + \sum_{\substack{(u,c,t)\in B' \\ c=(v,s)}} \log \sigma(-\overrightarrow{c} \cdot (\overrightarrow{u} + \overrightarrow{t})) \tag{3.4}$$

In B, the context $c = (v, s)$ is the combination of a node v from the graph, and a sign s derived from Equations (3.5) and (3.6) below.

3.3.2 Training data creation

To create the training data, we follow the random walk sampling procedure proposed in node2vec [46] and presented in Equation (3.5). However, while node2vec operates on unsigned graphs with no edge attribute, our embedding approach naturally incorporates signed edges as well as edge attributes such as topics. Given a signed topical graph, we outline how we create training examples to learn node and topic embeddings using our skip-gram objective.

Random walks on edge-attributed graphs

We consider second-order random walks. We first iterate through each topic-specific subgraph G_t and mask the edge signs, yielding a topic-graph $G'_t = (V_t, E'_t)$ where all

edges are unsigned. We define a second-order random walk with two parameters p and q that guide the walker on G'_t. In particular, let us consider a walker that just traversed edge (r, u) and now resides at node u. The walker next decides to walk on an existing edge (u, v) with the unnormalised transition probability π_{uv}:

$$\pi_{uv} = \begin{cases} \frac{1}{p} & \text{if } v = r \\ 1 & \text{if } d_{rv} = 1 \\ \frac{1}{q} & \text{if } d_{rv} = 2 \end{cases} \tag{3.5}$$

where d_{rv} is the shortest path distance between nodes r and v. p and q are return and in-out parameters respectively, and control how fast the walk explores and leaves the neighborhood of starting node s. For example, $q < 1$ means the walker is more inclined to visit nodes which are further away from node s. For $p = q = 1$, we recover the unbiaised (and more computationally efficient) random walk proposed in deepwalk [115]. We simulate random walks of fixed length, starting from each node. At every step of the walk, the sampling is done based on transition probabilities defined in Equation (3.5). Note that a given random walk only spans one edge attribute but carries this edge attribute during the training of the skip-gram model - see Figure 3.1 (right).

Creation of signed contexts

In signed graphs, positively connected nodes should be embedded in close proximity while negatively connected nodes should be farther away. Unlike with deepwalk and node2vec whereby a source node predicts context node, we propose to predict *sign and node* as contexts. In other words, we predict not only the context node, but also whether the source node connects positively or negatively with them on a given topic. While the context node is determined by the random walk, there may not be a signed edge between a source node and context node for that topic. In these cases, we apply strong balance theory [52] to infer whether or not a source and a non-connecting context node *agree* on some topic. Let t be an arbitrary topic, and consider the graph G_t depicted in Figure 3.2. Assuming a random walk sampled via the procedure described above in Equation (3.5), we have a sequence of nodes. Using a window of size L around a source node u_0, $2L$ context nodes are produced from the walk, L before u_0 and L after: $(u_{-L} \dots u_0 \dots u_{+L})$. In addition we compute the inferred sign $S(u_0, u_i)$ between the source node and the i_{th} context node as follows:

$$S(u_0, u_i) = \begin{cases} \prod_{m=i+1}^{0} w_{u_{m-1}u_m} & i < 0 \\ \prod_{m=1}^{i} w_{u_{m-1}u_m} & i > 0 \end{cases} \tag{3.6}$$

where w_{uv} is the weight, $+1$ or -1, between nodes u and v.

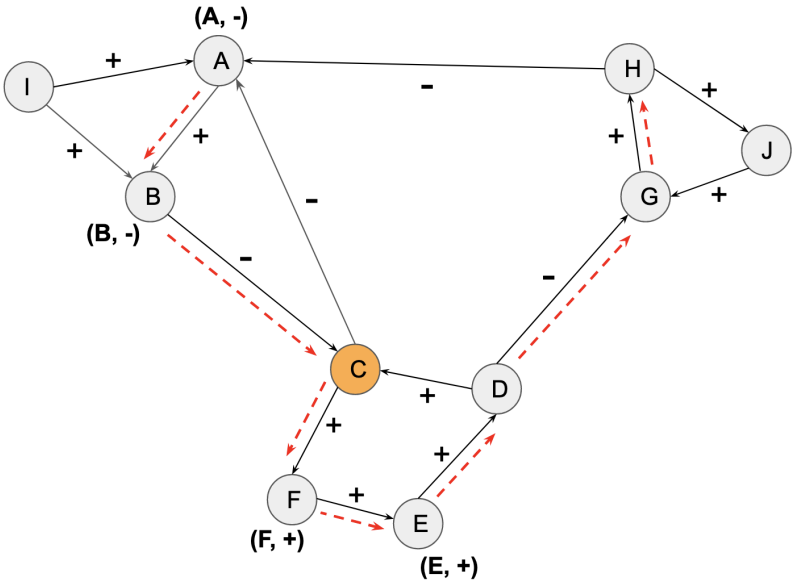


Figure 3.2: **A random walk sampling contexts on a signed graph**. The corresponding sign-informed contexts for source node $u_o = C$ are shown in bold (assuming a window of size 2).

Equation (3.6) mimics strong balance theory to assign each context node a sign with respect to the source node. Indeed, $S(u_0, u_i)$ is equal to $+1$ (resp. -1) if there is an even (resp. odd) number of negative edges between nodes u_0 and u_i. These patterns are in keeping with the beliefs that *the friend (+) of my friend (+) is my friend (+)*, *the friend (+) of my enemy (−) is my enemy (−)*, and *the enemy (−) of my enemy (−) is my friend (+)*. Equation (3.6) applies this to (dis)agreements over topics and as such, we can compute the (dis)agreement sign between the source node and a context node simply by multiplying the edge signs between the source and context as defined by the random walk between them. Although random walk generations and balance theory are applied on each topical graph independently, we show in sections 3.4.1 and 3.6.3 that our model is able to learn associations between topics during training. By incorporating these signs with the source node alongside each context node, our skip-gram objective needs to not only predict the context node, but also whether or not the context node agrees with the source node on a topic. As such, node proximity and stance both influence a node's and a topic's embedding. That is why we call our model `SEM`, for *Stance Embeddings Model*.

3.4 Theory of SEM

In this section, we perform a theoretical analysis of SEM. We first provide some intuition and interpretation of the embeddings learnt by SEM (section 3.4.1). We then present an analytical expression of a random walk process on signed graphs in section 3.4.2. We use this process to show that SEM is equivalent to a matrix factorisation approach.

3.4.1 Interpretation of the learnt embeddings

By generalising to signed topical graphs, we can show that SEM learns embedding matrices $W_u = (\overrightarrow{u_1} \ ... \ \overrightarrow{u_{|\mathcal{G}|}})$, $W_c = (\overrightarrow{c_1} \ ... \ \overrightarrow{c_{2|\mathcal{G}|}})$, and $W_t = (\overrightarrow{t_1} \ ... \ \overrightarrow{t_{|\mathcal{T}|}})$ that satisfies the following:

$$(\overrightarrow{u_i} + \overrightarrow{t_m}) \cdot \overrightarrow{c_j} = \log\left(\frac{\#(u_i, c_j, t_m)|B|}{\#(u_i, t_m)\#(c_j)}\right) - \log b \tag{3.7}$$

where b is the number of negative samples; $\#(u_i, c_j, t_m)$, $\#(u_i, t_m)$, and $\#(c_j)$ denote the number of times (u_i, c_j, t_m), (u_i, t_m), and c_j appear in B respectively[3] the mathematical proof of Equation (3.2) provided in [85].

Topic correlations We now show that SEM can learn topic correlations in spite of the fact that random walk generations and balance theory are applied on each topical graph independently. We take two topics t_m and $t_{m'}$, and any node u_i and context c_j. Through Equation (3.7), we can write:

$$\begin{aligned}(\overrightarrow{t_m} - \overrightarrow{t_{m'}}) \cdot \overrightarrow{c_j} &= (\overrightarrow{u_i} + \overrightarrow{t_m} - \overrightarrow{u_i} - \overrightarrow{t_{m'}}) \cdot \overrightarrow{c_j} \\ &= (\overrightarrow{u_i} + \overrightarrow{t_m}) \cdot \overrightarrow{c_j} - (\overrightarrow{u_i} + \overrightarrow{t_{m'}}) \cdot \overrightarrow{c_j} \\ &= PMI\left((u_i, t_m), c_j\right) - PMI\left((u_i, t_{m'}), c_j\right)\end{aligned}$$

where PMI stands for Pointwise Mutual Information. In the cases where G_{t_m} and $G_{t_{m'}}$ have the same signed adjacency matrix, the same training data is generated from both graphs asymptotically, i.e. in the limit of infinite number of walks. As a consequence, their respective topic embeddings are identical:

$$\forall u_i, \forall c_j, PMI\left((u_i, t_m), c_j\right) = PMI\left((u_i, t_{m'}), c_j\right) \implies \forall c_j, \overrightarrow{t_m} \cdot \overrightarrow{c_j} = \overrightarrow{t_{m'}} \cdot \overrightarrow{c_j}$$

This condition is equivalent to $\overrightarrow{t_m} = \overrightarrow{t_{m'}}$ as long as the set of vectors c_j is large and diverse enough to constitute a generating set of the embedding space [53]. In

[3] It suffices to apply the proof (see [85], section 3.1) to each topic-specific sub-graph individually.

other words, SEM closely embed topics on which nodes show similar connectivity behaviours. Conversely, we take two nodes u_i and v_j that are diametrically distant in G_{t_m} but positively connected neighbours in $G_{t_{m'}}$. In graph G_{t_m}, $\#(u_i, v_j^+) = o(\#(u_i))$ because u_i and v_j are unlikely to appear in the same context window. In contrast, $\#(u_i, v_j^+) = \Theta(\#(u_i))$ in graph $G_{t_{m'}}$ since u_i and v_j are positive neighbours. Using the expression of $(\overrightarrow{t_m} - \overrightarrow{t_{m'}}) \cdot \overrightarrow{c_j}$ above, this results, into:

$$\begin{aligned}
\forall v_j, \|\overrightarrow{t_m} - \overrightarrow{t_{m'}}\| \|\overrightarrow{v_j^+}\| &\geq |(\overrightarrow{t_m} - \overrightarrow{t_{m'}}) \cdot \overrightarrow{v_j^+}| \\
&= |(\overrightarrow{u_i} + \overrightarrow{t_m}) \cdot \overrightarrow{v_j^+} - (\overrightarrow{u_i} + \overrightarrow{t_{m'}}) \cdot \overrightarrow{v_j^+}| \\
&= \| \log \frac{\#(u_i, v_j^+, t_m)}{\#(u_i, t_m)} \frac{\#(u_i, t_{m'})}{\#(u_i, v_j^+, t_{m'})} \| \\
&\longrightarrow \infty
\end{aligned}$$

where we have used the Cauchy–Schwarz inequality [13] in the first row. In other words, topics which display no similarity in node connectivity patterns are pushed far apart.

Target node sign disambiguation If a graph is unbalanced, as it is often the case in real-world signed graphs [83], the data creation strategy may assign different signs from a source node to target node by following different walks between the two nodes. We quantify how this ambiguity affects the embedding training process. For a topic t_m, a source node u_i and a target node v_j:

$$\begin{aligned}
(\overrightarrow{u_i} + \overrightarrow{t_m}) \cdot (\overrightarrow{v_j^+} - \overrightarrow{v_j^-}) &= (\overrightarrow{u_i} + \overrightarrow{t_m}) \cdot \overrightarrow{v_j^+} - (\overrightarrow{u_i} + \overrightarrow{t_m}) \cdot \overrightarrow{v_j^-} \\
&= \log \frac{\#(u_i, v_j^+, t_m)}{\#(u_i, v_j^-, t_m)} \frac{\#(v_j^-)}{\#(v_j^+)}
\end{aligned}$$

We denote $\#(v_j) = \#(v_j^-) + \#(v_j^+)$ the number of times v_j appear in the training data, and $\#(u_i, v_j, t_m) = \#(u_i, v_j^+, t_m) + \#(u_i, v_j^-, t_m)$ the number of times (u_i, v_j) appear in the topical graph G_{t_m}, irrespective of the context sign. With this notation, the above expression reduces to:

$$(\overrightarrow{u_i} + \overrightarrow{t_m}) \cdot (\overrightarrow{v_j^+} - \overrightarrow{v_j^-}) = -\log \left(\frac{\frac{\#(u_i, v_j, t_m)}{\#(u_i, v_j^+, t_m)} - 1}{\frac{\#(v_j)}{\#(v_j^+)} - 1} \right)$$

In cases where most paths between node u_i and v_i are positive (resp. negative) in topical graph G_{t_m}, the expression on the right-hand side tends to infinity, which results in embeddings v_j^+ and v_j^- pushed further apart. In other words, the node's positive and negative positions are clearly distinguishable from each other.

3.4.2 Random walks on signed graphs: analytical expression

We show in section 3.4.3 that optimising `SEM` is equivalent to performing spectral matrix factorisation for carefully chosen hyperparameters. In order to do so, we define a random walk process which mimics the way `SEM` samples node-contexts pairs in the graph. For the sake of simplicity, and because random walks pass over one topical graph at a time, we consider a graph $\mathcal{G}$ spanning one topic only. We assume the graph is connected, undirected and signed. Several walkers move in the graph, starting from different nodes. We consider two types of walkers, positive and negative. We denote $n^+_{i,0}$ ($n^-_{i,0}$) the initial number of positive (negative) walkers at node i. Both walker types perform random walks by walking on an edge independently of its sign. If a walker takes a positive edge, it keeps its own sign; otherwise it changes sign. At any step r:

$$\begin{aligned} n^+_{j,r+1} &= \sum_i \frac{1}{k_i}(A^+_{ij} n^+_{i,r} + A^-_{ij} n^-_{i,r}) \\ n^-_{j,r+1} &= \sum_i \frac{1}{k_i}(A^-_{ij} n^+_{i,r} + A^+_{ij} n^-_{i,r}). \end{aligned} \tag{3.8}$$

We denote $N_i = n^+_i + n^-_i$ is the total number of walkers at node i (irrespective of their sign) and $\Delta_i = n^+_i - n^-_i$ denotes the *charge* of walkers at node i. The charge can be positive or negative, and represents the relative preponderance of positive or negative walkers sitting at a given node. With this notation,

$$\begin{aligned} n^+_i &= \frac{N_i + \Delta_i}{2} \\ n^-_i &= \frac{N_i - \Delta_i}{2}. \end{aligned} \tag{3.9}$$

Summing and subtracting Equations (3.8), the dynamics of N and Δ are given by:

$$\begin{aligned} N_{j,r+1} &= \sum_i \frac{1}{k_i} |A|_{ij} N_{i,r} \\ \Delta_{j,r+1} &= \sum_i \frac{1}{k_i} A_{ij} \Delta_{i,r} \end{aligned} \tag{3.10}$$

where $|A| = A^+ + A^-$. In the vector form with $\mathbf{N_r} = (N_{1,r}, ..., N_{|\mathcal{G}|,r})$ and $\mathbf{\Delta_r} = (\Delta_{1,r}, ..., \Delta_{|\mathcal{G}|,r})$:

$$\begin{aligned} \mathbf{N_{r+1}} &= \mathbf{N_r} D^{-1} |A| \\ \mathbf{\Delta_{r+1}} &= \mathbf{\Delta_r} D^{-1} A. \end{aligned} \tag{3.11}$$

where D is a diagonal matrix whose diagonal entries are the degrees k_u of the nodes $u \in \mathcal{G}$. Eventually, for any step r and chosen initial conditions N_0 and Δ_0:

$$\begin{aligned} \mathbf{N_r} &= \mathbf{N_0} P^r \\ \mathbf{\Delta_r} &= \mathbf{\Delta_0} Q^r \end{aligned} \tag{3.12}$$

with $P = D^{-1}|A|$ and $Q = D^{-1}A$. P_{ij} is the probability for a walker to go from node i to j, irrespective of the edge sign A_{ij}. Q_{ij} is the relative charge that transfers from node i to j. The transition matrix P describes the connectivity of graph while the charge matrix Q informs about the charge flux in the graph. Altogether, the two matrices carry the complementary information necessary to describe a signed graph. P is row-normalised so 1 is one of its eigenvalues [102]. As a consequence, the dynamics of $\mathbf{N_r}$ converges towards a stationary distribution π such that $\pi = \pi P$ in Equation (3.11) [76]. Previous literature has shown that 1 is also an eigenvalue for Q if and only if $\mathcal{G}$ is a balanced graph [58]. However, real-world signed social graphs are rarely balanced [83, 82]. We also note that P and Q are related to the unsigned $I - P$ and signed $I - Q$ random-walk laplacians [77, 57].

3.4.3 Implicit matrix factorisation

We now show that `SEM` is asymptotically equivalent to factorising a matrix whose expression is provided in Equation (3.21). For one topic, Equation (3.7) rewrites as:

$$\overrightarrow{u_i} \cdot \overrightarrow{c_j} = \log\left(\frac{\#(u_i, c_j)|B|}{\#(u_i)\#(c_j)}\right) - \log b \tag{3.13}$$

We assume that the walks are of first-order (Markovian process) for the sake of simplicity. We recall that L is the context window size in our signed context generation process (section 3.3.2). We take any training sample $(u, c) \in B$. Without loss of generality, we assume that $c = v^+$. According to Equation (3.6), the sample (u, v^+) is in B only if there exists a signed random walk that joins u and v through a path with an even number of negative edges. The vertex u can either appear before or after its context v^+, at a distance $r \in [1, L]$ in the corresponding random walk sequence. We denote $\#(u, v^+)_{\overrightarrow{r}}$ and $\#(u, v^+)_{\overleftarrow{r}}$ the number of times u appear r steps before and after its context v^+ respectively in a random walk. The total number of times u and v^+ appear within the same context window is therefore:

$$\#(u, v^+) = \sum_{r=1}^{L} \#(u, v^+)_{\overrightarrow{r}} + \#(u, v^+)_{\overleftarrow{r}} \tag{3.14}$$

We link the quantity $\#(u, v^+)$ to the structure of the signed graph by defining $T^+ = D^{-1}A^+$ and $T^- = D^{-1}A^-$ so that Equation (3.8) rewrites as:

$$\begin{aligned} n^+_{j,r+1} &= \sum_i T^+_{ij} n^+_{i,r} + T^-_{ij} n^-_{i,r} \\ n^-_{j,r+1} &= \sum_i T^-_{ij} n^+_{i,r} + T^+_{ij} n^-_{i,r} \end{aligned} \tag{3.15}$$

Equations (3.15) are defining a random walk process on an unsigned graph $\bar{\mathcal{G}}$ whose nodes are u^+ and u^- for all nodes $u \in \mathcal{G}$. A positive edge between nodes u and v in $\mathcal{G}$ translates into one unsigned edge between u^+ and v^+, and one unsigned edge between u^- and v^- in $\bar{\mathcal{G}}$. Similarly, a negative edge between nodes u and v in $\mathcal{G}$ translates into one unsigned edge between u^+ and v^-, and one unsigned edge between u^- and v^+ in $\bar{\mathcal{G}}$. An example is depicted in Figure 3.3. A node u with degree k in $\mathcal{G}$ gives rise to two nodes $u+$ and $u-$ with degree k in $\bar{\mathcal{G}}$. Similarly, a random walk starting at node u_0 in $\mathcal{G}$ defines two random walks starting at nodes u_0^+ and u_0^- in $\bar{\mathcal{G}}$ respectively.

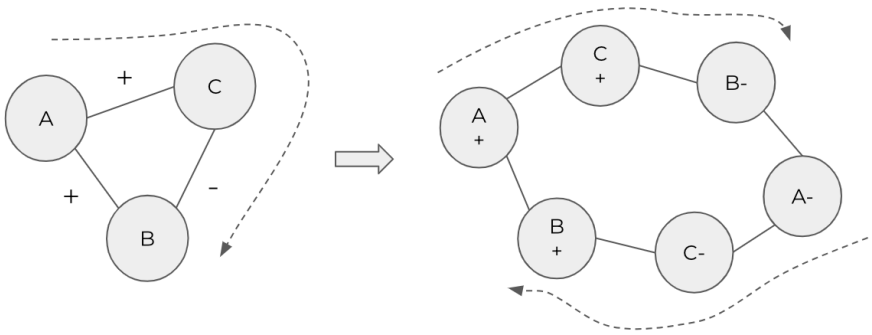


Figure 3.3: **Equivalence between $\mathcal{G}$ (left) and $\bar{\mathcal{G}}$ (right)**. A signed random walk on $\mathcal{G}$ generates two unsigned random walks on $\bar{\mathcal{G}}$.

Based on this observation, we define:

$$\bar{A} = \left[\begin{array}{c|c} A^+ & A^- \\ \hline A^- & A^+ \end{array}\right], \; \bar{D} = \left[\begin{array}{c|c} D & 0 \\ \hline 0 & D \end{array}\right] \text{ and } \bar{T} = \left[\begin{array}{c|c} T^+ & T^- \\ \hline T^- & T^+ \end{array}\right] \tag{3.16}$$

such that $\bar{T} = \bar{D}^{-1}\bar{A}$. With this notation, we simplify Equation (3.8):

$$(\mathbf{n}^+ \;\; \mathbf{n}^-)_{\mathbf{r+1}} = (\mathbf{n}^+ \;\; \mathbf{n}^-)_{\mathbf{r}} \bar{T}, \text{or} \tag{3.17}$$

$$(\mathbf{n}^+ \;\; \mathbf{n}^-)_{\mathbf{r}} = (\mathbf{n}^+ \;\; \mathbf{n}^-)_{\mathbf{0}} \bar{T}^r \tag{3.18}$$

where $\mathbf{n}^+_{\mathbf{r}}$ (resp. $\mathbf{n}^-_{\mathbf{r}}$) is the row vector whose entries are the number of positive (resp. negative) walkers at each node at step r.

For any $k < L$, a walker going from node u_k to u_{k+1} on a negative edge in $\mathcal{G}$ corresponds to the two walkers in $\bar{\mathcal{G}}$ walking from u_k^+ to u_{k+1}^- and u_k^- to u_{k+1}^+ respectively. As a result, $\#(u, v^+)_{\overrightarrow{r}}$ the number of times u appears r steps before its context v^+ in a random walk in $\mathcal{G}$ is equal to $\#(u^+, v^+)_{\overrightarrow{r}}$ the number of times u^+ appears r steps before node v^+ in a random walk in $\bar{\mathcal{G}}$, and equal to $\#(u^-, v^-)_{\overrightarrow{r}}$ the number of times u^- appears r steps before node v^- in a random walk in $\bar{\mathcal{G}}$: $\#(u, v^+)_{\overrightarrow{r}} = \#(u^+, v^+)_{\overrightarrow{r}} = \#(u^-, v^-)_{\overrightarrow{r}}$. We denote $B_{\overrightarrow{r}}$ ($B_{\overleftarrow{r}}$) the subset of B for which the context appear r steps after (before) the node. The strong advantage of reasoning in $\bar{\mathcal{G}}$ is that it is an unsigned graph so we can use all the known spectral properties of unsigned matrices such as $\bar{A}$ the adjacency matrix of $\bar{\mathcal{G}}$ and its transition matrix $\bar{T}$ in Equations (3.16) [77]. In particular, Equation (3.17) converges towards a stationary state $\pi = \pi\bar{T}$, where the stationary vector $\pi = (\pi^+ \ \ \pi^-)$ is normalised such that $\pi_{u^+} = \pi_{u^-} = \frac{k_u}{\text{vol}(\bar{\mathcal{G}})} = \frac{k_u}{2\text{vol}(\mathcal{G})}$ and where the volume of a graph is defined as its total number of edges. By choosing the initial number of walkers at node u to be π_u, the proportion of walkers that start at node u^+ (resp. v^+) and reach v^+ (resp. u^+) in r steps in $\bar{\mathcal{G}}$, is given in stationary regime by:

$$\frac{\#(u^+, v^+)_{\overrightarrow{r}}}{|B_{\overrightarrow{r}}|} = \frac{k_u}{2\text{vol}(\mathcal{G})}(\bar{T}^r)_{u^+,v^+} \tag{3.19}$$

$$\frac{\#(u^+, v^+)_{\overleftarrow{r}}}{|B_{\overleftarrow{r}}|} = \frac{k_v}{2\text{vol}(\mathcal{G}))}(\bar{T}^r)_{v^+,u^+} \tag{3.20}$$

Using the notation from Equation (3.14):

$$\begin{aligned}
\frac{\#(u, v^+)}{|B|} &= \sum_{r=1}^{L} \frac{\#(u, v^+)_{\overrightarrow{r}}}{|B|} + \frac{\#(u, v^+)_{\overleftarrow{r}}}{|B|} \\
&= \frac{1}{2L}\sum_{r=1}^{L} \frac{\#(u, v^+)_{\overrightarrow{r}}}{|B_{\overrightarrow{r}}|} + \frac{\#(u, v^+)_{\overleftarrow{r}}}{|B_{\overleftarrow{r}}|} \\
&= \frac{1}{2L}\sum_{r=1}^{L} \frac{\#(u^+, v^+)_{\overrightarrow{r}} + \#(u^-, v^-)_{\overrightarrow{r}}}{2|B_{\overrightarrow{r}}|} + \frac{\#(u^+, v^+)_{\overleftarrow{r}} + \#(u^-, v^-)_{\overleftarrow{r}}}{2|B_{\overleftarrow{r}}|} \\
&= \frac{1}{4L}\sum_{r=1}^{L} \frac{k_u}{2\text{vol}(\mathcal{G})}\left[(\bar{T}^r)_{u^+,v^+} + (\bar{T}^r)_{u^-,v^-}\right] + \frac{k_v}{2\text{vol}(\mathcal{G})}\left[(\bar{T}^r)_{v^+,u^+} + (\bar{T}^r)_{v^-,u^-}\right] \\
&= \frac{1}{4L}\sum_{r=1}^{L} \frac{k_u}{\text{vol}(\mathcal{G})}(\bar{T}^r)_{u^+,v^+} + \frac{k_v}{\text{vol}(\mathcal{G})}(\bar{T}^r)_{v^+,u^+} \\
&= \frac{1}{4L\text{vol}(\mathcal{G})}\sum_{r=1}^{L} k_u(\bar{T}^r)_{u^+,v^+} + k_v(\bar{T}^r)_{v^+,u^+}
\end{aligned}$$

where we used the fact that $|B_{\overrightarrow{r}}| = |B_{\overleftarrow{r}}| = \frac{|B|}{2L}$ by construction of the node-context pairs. We also notice that nodes u^+ and u^- play equivalent structural roles in $\bar{\mathcal{G}}$ due to their similar relation with node u in $\mathcal{G}$. For this reason, $(\bar{T}^r)_{u^+,v^+} = (\bar{T}^r)_{u^-,v^-}$, and $(\bar{T}^r)_{v^+,u^+} = (\bar{T}^r)_{v^-,u^-}$. We can prove similarly that:

$$\frac{\#(u,v^-)}{|B|} = \frac{1}{4L\text{vol}(\mathcal{G})}\sum_{r=1}^{L} k_u(\bar{T}^r)_{u^+,v^-} + k_v(\bar{T}^r)_{v^-,u^+}$$

Using these two expressions and conditioning over node v, we compute:

$$\begin{aligned}
\frac{\#(u)}{|B|} &= \sum_v \frac{\#(u,v^+)}{|B|} + \frac{\#(u,v^-)}{|B|} \\
&= \frac{1}{4L\text{vol}(\mathcal{G})}\sum_{r=1}^{L} k_u\left[\sum_v (\bar{T}^r)_{u^+,v^+} + (\bar{T}^r)_{u^+,v^-}\right] + \left[\sum_v k_v\left((\bar{T}^r)_{v^+,u^+} + (\bar{T}^r)_{v^-,u^+}\right)\right] \\
&= \frac{1}{4L\text{vol}(\mathcal{G})}\sum_{r=1}^{L} k_u + k_u = \frac{k_u}{2\text{vol}(\mathcal{G})}
\end{aligned}$$

where we used the fact that $\bar{T}^r$ is row-normalised and $D\bar{T}^r$ is column-normalised [102]. With a similar approach, we compute:

$$\frac{\#(v^+)}{|B|} = \frac{\#(v)}{|B|}\frac{\#(v^+)}{\#(v)} = \frac{k_v}{2\text{vol}(\mathcal{G})}\omega_{v^+}$$

where $\omega_{v^+} = \frac{\#(v^+)}{\#(v)}$ denotes the relative propensity of positive contexts of v among all the samples node v appears in. All put together, Equation (3.13) is as:

$$\begin{aligned}
\frac{\#(u,v^+)|B|}{\#(u)\#(v^+)} &= \frac{vol(G)}{L}\sum_{r=1}^{L}\frac{1}{k_v\omega_{v^+}}(\bar{T}^r)_{u^+,v^+} + \frac{1}{k_u\omega_{v^+}}(\bar{T}^r)_{v^+,u^+} \\
\frac{\#(u,v-)|B|}{\#(u)\#(v-)} &= \frac{vol(G)}{L}\sum_{r=1}^{L}\frac{1}{k_v\omega_{v^-}}(\bar{T}^r)_{u^+,v^-} + \frac{1}{k_u\omega_{v^-}}(\bar{T}^r)_{v^-,u^+}
\end{aligned}$$

which means, in the matrix form, that `SEM` implicitly factorises the following matrix:

$$\log\frac{vol(G)}{L}\left(\sum_{r=1}^{L}\bar{T}^r\right)\left[\bar{\Omega}\bar{D}\right]^{-1} + \text{cst} \tag{3.21}$$

where the log is applied element-wise, and $\bar{\Omega}$ is a diagonal matrix whose diagonal entries are the relative contributions $(\omega+ \;\; \omega-)$. The expression in the log corresponds to the superposition of random walk processes of lengths up to L, scaled by the propensity of observing a given context sign in the graph structure.

3.5 Building signed topical graphs

We present `TwitterSG` (section 3.5.1) and `BirdwatchSG` (section 3.5.2) two novel large-scale signed topical graphs that we use in our experiments (section 3.6). We compare `TwitterSG` and `BirdwatchSG` with existing real-world signed graphs in section 3.5.3).

3.5.1 `TwitterSG`

Twitter Signed Graph, or `TwitterSG`, is a signed, directed, edge-attributed graph of users, drawn from interactions on the social platform X, formerly called Twitter until July 2023. A positive edge exists from user A to user B if user A liked a tweet posted by user B. A negative edge exists from user A to user B if user A expressed opposition towards user B's tweet, e.g., by replying *I disagree with you*. The topic of an edge from user A to user B is determined by the topic of user B's tweet, also called the *target tweet*. Tweet topics were inferred with a topic classifier provided and used in production by Twitter (X); we restrict interactions in `TwitterSG` to sports-related topics (e.g., sports teams, players, managers, or events). The tweets related to these interactions were published between 20th May (Ice Hockey World Championships) and 8th August 2021 (closing date of the 2020 Tokyo Olympic Games), and collected via Twitter (X) API[4].

Several challenges arise when attempting to build a large signed graph with interactions on Twitter (X). First, the graph may be extremely sparse due to the large number of active users and the skewed distribution of tweets per user. Second, opposition generally goes silent or is expressed via reply to a tweet, which requires more effort than clicking a *like* button to express support. A user may just keep scrolling if they do not agree with a statement, instead of engaging with it. For this reason, there is a substantial imbalance between the amount of support and opposition signals. And lastly, opposition in a tweet may be implicit.

To overcome these challenges, we adopt a multi-step strategy to create a user-tweet graph (Figure 3.4), that we project onto a user-user graph:

1. We curated a list of high-precision English and French expressions which express clear opposition (e.g. "I disagree" and "you're wrong")[5]. We retained all sports-

[4] https://developer.twitter.com/en/products/twitter-api

[5] Selected expressions are available at https://github.com/lejohnyjohn/learning-stance-embeddings-from-signed-social-graphs.

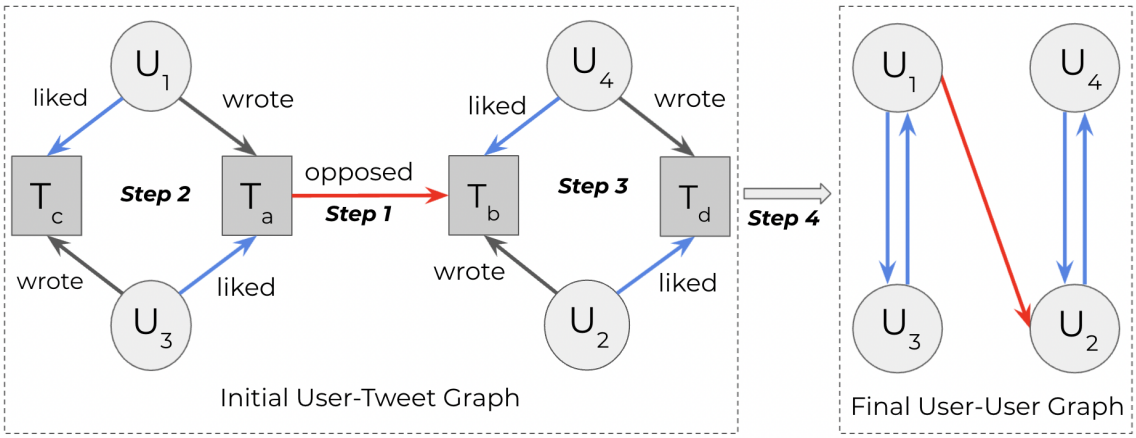


Figure 3.4: **Steps involved in building `TwitterSG`.**

related tweets T_a containing at least one of these expressions, and the tweets T_b they replied to. we denote T_a (T_b) the tweet posted by user U_1 (U_2).

2. To control the graph sparsity, we retained all users $U3$ who both wrote a tweet T_c liked by user $U1$, and liked the opposition tweet T_a written by user $U1$.

3. Similarly, we retained all users $U4$ who both wrote a tweet T_d liked by user $U2$, and liked the tweet T_b written by user $U2$. At this stage, the positive interactions largely outnumbered the negative ones (300k negative interactions for more than 100M positive ones). Filtering out a large portion of positive edges would increase the share of negative edges but would decrease the number of users. Conversely, filtering in a large portion of positive edges would push the share of negative edges close to 0 but increase the number of users. We found a trade-off cut by selecting a share of likes retrieved in steps (2) and (3) so that the share of negative edges in our graph is close to 10%. We ranked the topics by decreasing frequency and filtered out all the tweets not related to the top 200 most frequent topics.

4. We projected the resulting user-tweet graph onto a user-user graph [56, 167], and anonymised all the nodes (users) and edges (tweets).

Eventually, the edge data of the final graph is provided under the format depicted in Figure 3.5. `TwitterSG` contains 753,944 nodes (users), 200 topics and 12,848,093 edges. Among these edges, 9.6% are negative (opposition) and 90.4% are positive. The most frequent topics are depicted in Figure 3.7 (top). There may be several edges between two nodes (several topics, several interactions).

source_node	*target_node*	*topic*	*weight*
1	6	Copa America	+1
1	6	NFL	-1
4	5	Kylian Mbappe	-1

Figure 3.5: **A sample of `TwitterSG`**. A positive (negative) edge represents a positive (negative) stance of the source node towards the target node for the given topic.

3.5.2 `BirdwatchSG`

Birdwatch Signed Graph, or `BirdwatchSG`, is a signed, directed, edge-attributed graph of users, drawn from note ratings on Community Notes, formerly called Birdwatch[6]. Birdwatch is a pilot launched by Twitter (X) in January 2021 in the USA to address misleading information on the platform in a community-driven fashion: the Birdwatch participants can identify information in tweets they believe is misleading and write notes that provide informative context. They can also rate the helpfulness (either *helpful*, *somewhat helpful*, or *not helpful*) of notes added by other contributors. All Birdwatch contributions are publicly available on the Birdwatch website[7] so that anyone can download and access to the data.

Starting with Birdwatch data from January to July 2021, we create a positive (negative) edge from participant U_1 to U_2 if participant U_1 rated a note written by participant U_2 as *helpful* (*not helpful*). We filter out the *somewhat helpful* ratings. The topic associated with an edge is the topic of the tweet the note refers to. We anonymise all the nodes and edges. Eventually, the edge data of the final graph is provided under the format depicted in Fig. 3.6.

The graph contains 2,987 nodes (users), 1,200 topics and 441,986 edges. Among these edges, 36.9% are negative (opposition) and 63.1% are positive. The most frequent topics are depicted in Figure 3.7 (bottom). There may be several edges between two nodes (several topics, several interactions).

3.5.3 Existing datasets

Early signed graphs were derived from observations in the physical world such as the relationships among Allied and Axis powers during World War II [7]. The develop-

[6]See Birdwatch (Community Notes) website at https://blog.twitter.com/en_us/topics/product/2021/introducing-birdwatch-a-community-based-approach-to-misinformation

[7]https://twitter.github.io/birdwatch/

source_node	*target_node*	*topic*	*weight*
1	6	US Politics	+1
1	6	Ted Cruz	-1
4	5	Twitter	-1

Figure 3.6: **A sample of `BirdwatchSG`**. A positive (negative) edge represents a positive (negative) stance of the source node towards the target node for the given topic.

ment of social media has enabled the mining of large signed social graphs. Signed graphs in social media often have thousands of users and millions of links, and they are usually sparser than physical world signed graphs. Epinions, Slashdot [83], Wiki-Rfa [155], BitcoinOtc, BitcoinAlpha [72, 71] are the largest and most widely used signed graphs for benchmarking signed graph embeddings methods. Epinions.com[8] was a product review site where users can write reviews for various products with rating scores from 1 to 5. Other users could rate the helpfulness of reviews. Slashdot[9] is a technology news platform on which users can create friend and foe links with other users. For a Wikipedia editor to become an administrator, a request for adminship (RfA) must be submitted, and any Wikipedia member may cast a supporting, neutral, or opposing vote. This induces a directed, signed graph Wiki-Rfa [155] in which nodes represent Wikipedia members and edges represent votes. BitcoinOtc and BitcoinAlpha [72, 71] are who-trusts-whom graphs of users who trade using Bitcoin on online platforms. Since Bitcoin users are anonymous, there is a need to maintain a record of users' reputation to prevent transactions with fraudulent and risky users. Platforms' members can rate each other members positively or negatively.

Table 3.1 provides statistics on the existing graphs, `TwitterSG` and `BirdwatchSG`. `TwitterSG` is $\sim$ 6x larger than the Epinions graph which, to the best of our knowledge, is the largest publicly available signed social graph. `TwitterSG` and `BirdwatchSG` are fully anonymized without personally identifiable information, and have the unique advantage to present attributed edges. We open source the graph for future research[10].

[8] https://en.wikipedia.org/wiki/Epinions

[9] https://slashdot.org/

[10] See Github repository at https://github.com/lejohnyjohn/learning-stance-embeddings-from-signed-social-graphs

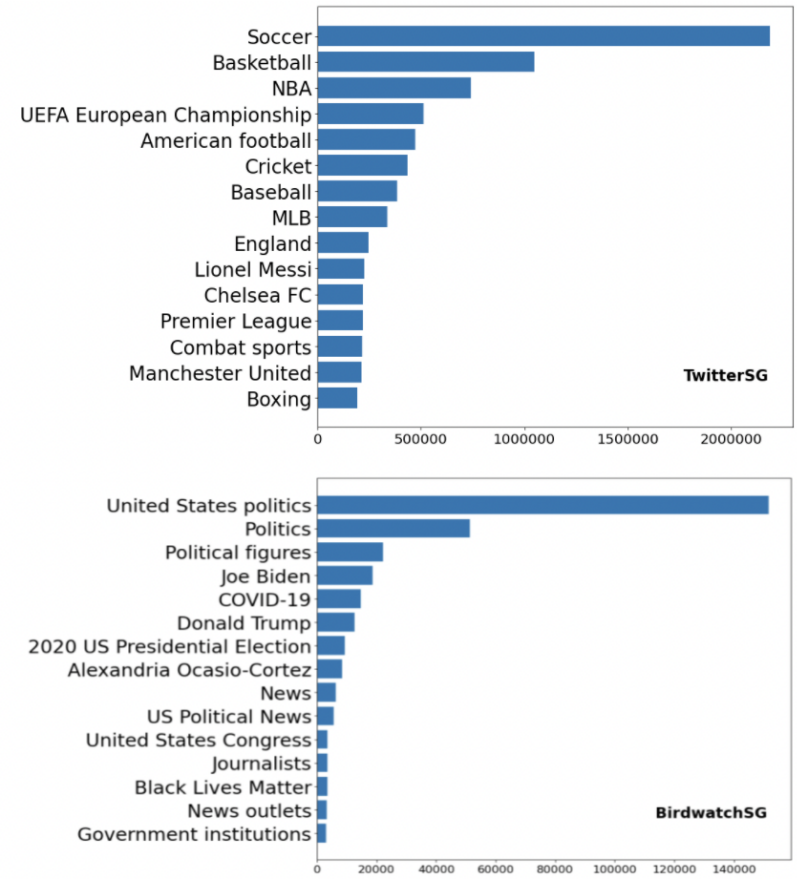


Figure 3.7: **Top 15 topics in `TwitterSG` (top) and `BirdwatchSG` (bottom)** respectively, ordered by decreasing frequency.

3.6 Learning in real-world graphs

In this section, we train and evaluate the embeddings produced by our `SEM` method and compare its performance to three state-of-the art signed graph embedding models on our `TwitterSG` and `BirdwatchSG` datasets.

3.6.1 Embedding Models

We evaluate two variants of our model: `SEM` in which the topic and node embeddings are added in the first layer of the topic-aware skip-gram architecture, as depicted in Figure 3.1, right; `SEM`-m in which the topic information is ignored. This corresponds to masking the topic embedding in the first layer of the topic-aware skip-gram architecture. The two variants only change whether the node and topic embedding are combined during training.

Dataset	$\|V\|$	$\|E\|$	$\%_{\|E_-\|}$
BitcoinAlpha	3,783	24,186	7%
BitcoinOtc	5,881	35,592	9%
Epinions	131,828	841,372	15%
Slashdot	77,357	516,575	23%
Wiki-Rfa	10,835	159,388	22%
`BirdwatchSG`	2,987	441,986	37%
`TwitterSG`	753,944	12,848,093	10%

Table 3.1: **Statistics of signed graph datasets**. The bottom datasets `BirdwatchSG`, `TwitterSG` are our contributions.

We compare `SEM` to three state-of-the-art signed graph embedding methods described in section 3.2.2: StEM [122], SIDE [65], SiNE [154]. We chose these methods based on their competitive performance, code availability and reproducibility. Like `SEM`-m, these three methods are topic agnostic and were only tested on signed graphs lacking topics, or other attributes, on edges.

3.6.2 Training setup

As a common practice, we set the node embedding dimension (d) to 64 for all methods and experiments. For the number of walks starting at each node, we have seen in section 3.4.3 that the ideal choice for implicit matrix factorisation is to use the stationary distribution of the unsigned random walk process, i.e. to start k_u walks from node u, where k_u is the degree of node u. We also choose a walk length $l = 40$, context size $L = 5$, return parameter $p = 1.5$, in-out parameter $q = 0.5$, negative sample size to 20, subsampling threshold to 1e-5, and the optimisation is run for 1 to 5 epochs [11]. For two given nodes and a given topic, edge weights are summed and the overall topical edge weight is set to +1 if the sum is positive, and -1 otherwise. For baseline methods, we use the same parameter settings as those suggested in their respective papers. The edge topic information is masked for baselines and `SEM`-m.

[11] The sentivity of the outcomes with respect to these hyperparameters has been studied for the original skip-gram (see [46], Section 4.4). We expect a similar behaviour in this work, because `SEM` and the original skip-gram are conceptually trained in a similar fashion.

3.6.3 Analysis

Evaluation protocol

We follow previous work by evaluating SEM, and baselines on a link sign prediction task [122, 65, 154]. In link sign prediction, we are given a signed graph where the sign, or agreement value, on several edges is missing and we predict each edge's sign value using the observed edges. In the context of TwitterSG and BirdwatchSG, link sign prediction is equivalent to detecting (dis)agreement - or stance - between two nodes. In particular, we formulate link sign prediction as a binary classification task using embedding learned from each method as follows. For each dataset, we perform 5-fold cross-validation (80/20% training/test set) and evaluate with mean AUC [100] over the 5 folds. For all approaches, we create edge embeddings by combining node embeddings u_1 and u_2 using $\Phi(u_1, u_2)$ using operations from Table 3.2. Note that this means that for topic-aware SEM , we do not explicitly use the topic embedding for evaluation. Using the edge representations in the training set, we fit a binary classifier to predict edge signs on the test set. Due to the sign imbalance sign in the edge data, we downsample the positive signs when fitting the classifier.

Operation	Output
hadamard	$w[i] = u_1[i] \times u_2[i]$
ℓ_1	$w[i] = \lvert u_1[i] - u_2[i] \rvert$
ℓ_2	$w[i] = (u_1[i] - u_2[i])^2$
Average	$w[i] = \frac{1}{2}(u_1[i] + u_2[i])$
Concatenation	$w = u_1 \otimes u_2$

Table 3.2: **Operations (Φ) to produce edge embeddings** from node embeddings for evaluation.

Stance detection

In Table 3.3, we report results for SEM variants and baselines using both nearest neighbors (kNN) and logistic regression (LR) classification on edge embeddings. For each approach, we report the best value over the choices of translation operator $\Phi(\cdot, \cdot)$ from Table 3.2.

On TwitterSG and BirdwatchSG, SEM-m, the topic-agnostic version of SEM, shows better or competitive performance with the three baselines. The topic-aware SEM variant significantly outperforms topic-agnostic baselines on both datasets and across

	TwitterSG			BirdwatchSG		
	kNN		LR	kNN		LR
	$k=5$	$k=10$		$k=5$	$k=10$	
SiNE	86.0	86.6	61.1	86.4	80.6	76.8
StEM	91.1	91.2	84.5	90.7	88.0	87.7
SIDE	91.0	87.5	82.1	92.6	90.0	82.7
SEM-m	91.0	92.7	84.5	92.5	90.5	87.0
SEM	**94.5**	**95.7**	**88.2**	**95.1**	**93.2**	**91.7**

Table 3.3: **Mean AUC from 5-fold CV on stance detection using nearest neighbors (kNN) and logistic regression (LR)** on edge embbedings to predict stance (edge sign).

both edge embedding classifiers. On TwitterSG, SEM improves the AUC by 3.4% and 3.0% for the $k = 5$ and $k = 10$ kNN classifiers respectively, compared to the best performing topic-agnostic method. On BirdwatchSG, SEM improves by 2.5% and 2.7% the AUC at $k = 5$ and $k = 10$ respectively, compared to the best performing topic-agnostic method. These results demonstrate that SEM learns improved node embeddings for signed edge prediction.

Cold-start topical predictions

One important advantage of learning node and topic embeddings jointly is the potential for predicting the stance of a node on topics for which we have not observed their engagement. We investigate the performance of methods on this *cold start* [88] subset of test samples (u_1, u_2, w, t) such that the engagement of user u_1 or u_2 on topic t was not observed during training. In other words, there is no training sample of the form $(u_1, ., ., t)$ or $(., u_2, ., t)$. This represents 28% and 17% of the test data for TwitterSG and BirdwatchSG respectively (average over 5 folds). In Table 3.4, we present the AUC results of edge sign predictions on the cold start setting (using only nearest neighbors classification since this had better performance overall). Only SEM is able to maintain performance across both datasets and edge embedding classifiers (compared to Table 3.3). This hints that, during training, SEM learns topic relationships such that an observed (dis)agreement on one topic affects the likelihood of (dis)agreements on other topics.

	TwitterSG		BirdwatchSG	
	kNN		kNN	
	$k = 5$	$k = 10$	$k = 5$	$k = 10$
SiNE	83.0	84.3	84.2	80.1
StEM	92.8	90.2	89.9	88.4
SIDE	88.7	86.0	91.3	89.0
SEM-m	90.4	91.9	91.8	89.0
SEM	**94.7**	**96.1**	**95.9**	**93.4**

Table 3.4: **Mean AUC from 5-fold CV on cold-start stance detection using nearest neighbors (kNN)**. Results remain comparable to Table 3.3 for SEM, demonstrating that we can still maintain high accuracy without prior observations of a user's engagement on a topic.

Adding topics post-embedding training

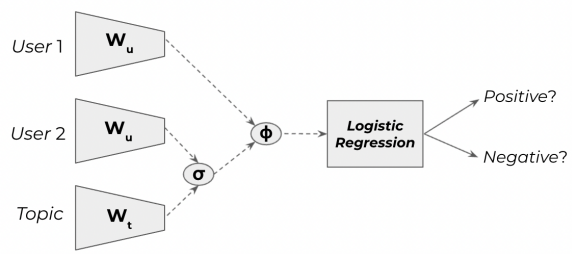


Figure 3.8: **Logistic regression classifier for edge sign prediction** to investigate learning attribute embeddings separately from node embeddings.

We investigate learning topic embeddings separately from node embeddings in order to understand the value of jointly learning these as we proposed. In order to explore this, we alter how we train a link prediction classifier for topic-agnostic approaches to also learn a topic embedding table. For topic-aware SEM , we instead opt to freeze this topic embedding table to what was learned during graph embedding.

As depicted in Figure 3.8, for a given edge $e = (u_1, u_2, s, t)$, this classifier takes as input the pre-trained node embeddings u_1 and u_2, combined with a topic embedding t learned as part of this classifier training process for topic-agnostic approaches. The node embedding u_2 and topic embeddings t are combined via a mask or by addition. The resulting vector is combined with u_1 node embedding vector via functions $\Phi(\cdot, \cdot)$ defined in Table 3.2. The resulting edge embedding is the input to the LR classifier.

Note that we deliberately combine the topic embedding with the node embedding u_2 only. Indeed, edge operations ℓ_1 and ℓ_2 in Table 3.2 involve the difference between source and target node embeddings. Therefore, combining the topic embedding into source and target embeddings would cancel the topic out. Note also that when we use a mask, we effectively ignore this learned (or frozen) topic embedding, reducing to the same setting for LR in Table 3.3. For the addition choice, the topic embedding (learned or frozen from graph embedding) is used for edge prediction.

	TwitterSG		BirdwatchSG	
	mask	add.	mask	add.
SiNE	61.1	62.0	76.8	77.3
StEM	84.5	84.6	87.7	88.1
SIDE	82.1	82.2	82.7	82.6
SEM-m	84.5	84.5	87.0	86.6
SEM	**88.2**	**88.0**	**91.7**	**90.3**

Table 3.5: **Mean AUC from 5-fold CV on stance detection** where we learn topic embeddings learnt during link prediction, separately from node embeddings.

In Table 3.5, for each graph embedding approach, we report the best AUC found over functions Φ. The performance of SEM remains unmatched by the topic-agnostic methods even when topic-agnostic approaches learn topic embeddings during classifier training. Performance is still significantly degraded compared to our best results in Table 3.3, demonstrating that training topic and node embeddings in tandem remains the most beneficial way to incorporate context (topic) into signed graph embeddings. We do note, however, that SEM performance decreases if we use the learned topic embedding at test time.

Implicit topic relationships

The model simply treats each topic graph separately but is still able to capture relationships. In Figure 3.9, we depict the topic embeddings obtained with SEM trained on TwitterSG, and projected with tSNE [150]. We can discern clear clusters of topics associated to a specific sport (e.g. NFL, hockey, baseball) or group of sports (e.g. fighting sports: WWE, Wrestling). Among these clusters, we observe finer-resolution groups. For instance, English football clubs lie close to the Premier League topic.

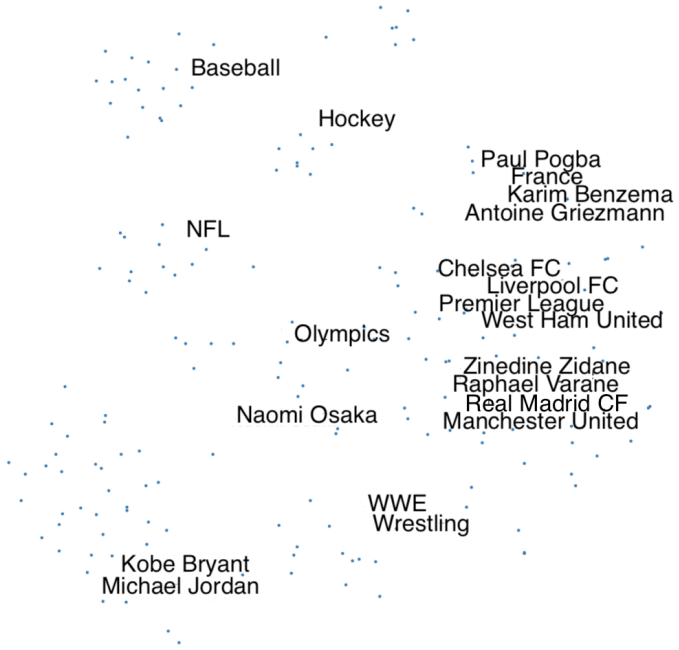


Figure 3.9: `TwitterSG` **topic embeddings learnt by** `SEM`. Related topics are embedded close to each other. Only certain representative topics are labelled for readability.

Karim Benzema, Antoine Griezmann and Paul Pogba are the closest neighbours to France, while Zinedine Zidane and Raphael Varane are close to Real Madrid CF. Michael Jordan and Kobe Bryant are closest neighbours. We observe similar patterns on `BirdwatchSG` topics. The presence of meaningful topical clusters demonstrate the ability of our method to capture topic similarities when a diverse range of topics are discussed. The US public debate is known to be politically polarised, and so are Birdwatch reports according to research in social science [161, 119, 59]. Consequently, we expect to observe two major clusters of user embeddings. Figure 3.10 displays the user embeddings obtained with `SEM` trained on `BirdwatchSG`, and projected with tSNE. The presence of two distinct opinion clusters prove the ability of our sign-informed context generation strategy to capture oppositions, and separate opposing views in the graph. Further, we visually inspect the ability of the model to distinguish positive and negative edges. Let $e = (u_1, u_2, w, t)$ be an edge of topic t going from user u_1 to u_2 with weight $w \in \{-1, 1\}$. For the sake of visualisation, we define the embedding of edge e as the hadamard product of the two user embeddings $u_1 \times u_2$. Figure 3.11 displays the projected `BirdwatchSG` edge embeddings obtained with `SEM` and tSNE. The positive (negative) edges are colored in dark (light). We observe distinct clusters of positive or negative edges, which confirms the capability of the model to discriminate positive and negative edges.

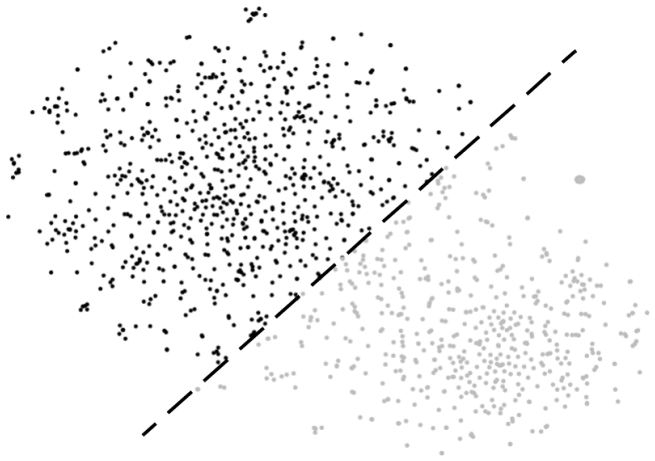


Figure 3.10: `BirdwatchSG` **node embeddings learnt by** `SEM`. Two opinion communities are observed in Birdwatch, in keeping with polarisation in the US political public debate.

3.7 Conclusion

In this chapter, we introduce `SEM`, a framework for learning latent associations via node and topic embeddings in signed, edge-attributed graphs. Utilising sign-informed random walks to generate training examples, we demonstrate how the scalable skip-gram objective can be successfully applied to learn signed-graph embeddings. Our approach is flexible and can incorporate arbitrary edge-attributes, such as topics, to provide context embeddings in signed interactions. We provide a theoretical analysis of our model, and links it to more traditional matrix factorisation approaches. Experimental results show that `SEM` embeddings outperform state-of-the-art signed-graph embedding techniques on two new datasets: `TwitterSG` and `BirdwatchSG`. We open-source these two datasets to the network science community to spur further research in signed graph and social network analysis.



Figure 3.11: `BirdwatchSG` **edge embeddings derived from** `SEM` **node embeddings**. Negative (positive) edges are depicted in black (grey). Edges with different signs form distinctive clusters.

Chapter 4

Modeling signed graphs with text

4.1 Introduction

In Chapter 3, we showed that the value of contextual information for the detection of latent associations in signed graphs. We realised that signed graph embedding methods are not designed to leverage context, like edge attributes, for fundamental tasks such as edge sign prediction and signed graph embeddings. An even more troublesome element is the fact that the open-source graph datasets they are tested on (see Chapter 3, section 3.5.3) do not provide multi-modal information, even though they are built upon multimodal sources: Slashdot [83], BitcoinOtc, and BitcoinAlpha [72, 71] include self-declared trust scores; Epinions [83] collects self-declared ratings; Wiki-Rfa [155] gathers members' votes.

Recognising that the context of the interactions has been overlooked in signed graph research, this Chapter - a dataset chapter - aims to fill this important gap in the literature. Because real-world interactions occur in contexts and manners more informative than any signal an adjacency matrix can capture, we aim to provide researchers with signed graphs that connect directly to the real-world mechanisms they represent. We do this by collecting multi-modal data to create a context-aware graph dataset.

We build, analyse and open-source `DEBAGREEMENT`, a signed graph with text data collected from Reddit discussions. In `DEBAGREEMENT`, the edges contain the textual exchanges of the discussion. A positive (negative) edge represents an agreement (disagreement) between the two nodes. This text can be used to consolidate with the graph structure for a multitude of downstream tasks such as predicting the sign of unobserved edges.

This chapter bridges the gap between two fields of research which can considerably benefit from each other: *edge sign prediction* and *(dis)agreement detection*.

Predicting (dis)agreements between two actors in a social network is similar to inferring the sign of the edge between their corresponding nodes in the signed graph. On the one hand, we have shown in chapter 3 that exploiting contextual information such as edge attributes is beneficial to edge sign prediction. On the other hand, recent work in stance detection has shown that the field would benefit from multimodal approaches [28, 64, 70, 157]. For instance, researchers have combined the textual content of tweets with hashtags, follow and retweet data to infer Twitter users' stances [31, 33, 16, 130, 38]. Despite these early-stage efforts, there exists no dataset providing reliable text and graph data to tackle edge sign detection.

We present `DEBAGREEMENT`, a signed graph built from real-world, text-based online interactions. `DEBAGREEMENT` follows active Reddit users across five forums, called subreddits: r/BlackLivesMatter, r/Brexit, r/Climate, r/Democrats, r/Republican. `DEBAGREEMENT` is therefore made of five temporal subgraphs totalling 42,894 comment-reply pairs (edges), with additional information on authorship (nodes), post content, timestamps of the interactions. Unlike existing datasets for (dis)agreement detection, `DEBAGREEMENT` is a graph, and contains real online discussions, with diverse writing styles, genres and topics of discussion. It does not rely on platform-specific features, such as retweets or hashtags. Through `DEBAGREEMENT`, structural principles, large language models, and extensive analysis and experiments, this chapter demonstrates how powerful it is to combine structural knowledge and textual understanding to unveil latent associations between nodes - the signs of the edges in this case.

We open source `DEBAGREEMENT`[1] to encourage future research. Beyond the opportunity to model signed graphs with text and context, `DEBAGREEMENT` offers researchers the ability to deploy network analysis tools to analyse actual online interactions. This includes investigating social theories and polarisation [149, 10], and studying opinion dynamics in social media [34, 1, 4]. Eventually, `DEBAGREEMENT` provides opportunities for combining recent advances in text-based machine learning (ML) [121] and graph representation learning (GRL) [47].

4.2 Existing datasets for (dis)agreement detection

Advances in stance detection have been surveyed recently [69]. There exists 24 publicly available text datasets for stance detection (see Table 6 of the survey). The available datasets have several drawbacks: i) they mostly provide textual data and ignore the underlying interaction graph, ii) all but two [32, 35] have fewer than 7,000

[1] https://scale.com/open-datasets/oxford

annotations, (iii) all but two [11, 35] are not annotated for (dis)agreement detection specifically, iv) they span a small range of topics and text genres, and v) the genre and writing style of the annotated samples is formal and structured. The latter point often simplifies the task of stance detection, but is not representative of real-world discussions. For instance, in the popular Perspectrum dataset [21], claims (e.g. *Animals should not be used for scientific or commercial testing*) and perspectives (e.g. *Any living entity should not be treated as objects or property as doing so allows them to be treated amorally*) are presented in a formal writing style. One dataset [27] tries to provide social network data (friends, replies, retweets, quotes' relations) along with 3,282 annotated Italian tweets related to the 2019 Sardines movement[2]. However, this dataset suffers from the two major limitations [69, 3]: (i) for privacy reasons, tweets are shared via tweet identifiers, making it impossible to retrieve deleted tweet content and network information of deleted users for future research; and ii) retweets may ease stance detection, but are feature-specific to the Twitter platform so cannot be generalised. `DEBAGREEMENT` alleviates each and every above-mentioned limitation.

4.3 DEBAGREEMENT

We create `DEBAGREEMENT` by retrieving data from the popular discussion platform, Reddit[3]. Reddit was the 11th most visited website globally as of November 2021[4], which makes it valuable to study popular topic of discussions and observe a diversity of behavioural patterns. Most Reddit discussions are downloadable for research purposes. Reddit is split into sub-forums of discussions, called *subreddits*, and denoted *r/** (e.g. r/Brexit). In each subreddit, Reddit users write titled posts, typically accompanied with a body of text and/or a link to an external website. These posts can be commented, upvoted or downvoted by other Reddit users (Figure 4.1). A ranking algorithm raises the visibility of a post based on the amount of upvotes it receives but decreases it with time so that the first posts visitors see are either recent or highly upvoted. Comments under a post are also visible and subject to a similar scoring and commenting system. This creates a tree structure where any comment can be commented on.

[2] https://en.wikipedia.org/wiki/Sardines_movement
[3] reddit.com
[4] https://www.statista.com/statistics/1201880/most-visited-websites-worldwide/



Figure 4.1: **A post and its comments on Reddit**.

4.3.1 Collection

We collected Reddit data from the Pushshift API[5] which provides historical open-source data for researchers [12]. We pull data from three subreddits of social movements r/BlackLivesMatter[6], r/Brexit[7], and r/Climate[8], and two subreddits of political affiliations r/Democrats[9] and r/Republican[10].

For r/Climate and r/Brexit, we collected all the posts and comments from the subreddits' creation dates through to May 2021 in order to track these social movements from their inception. For r/BLM, r/Democrats and r/Republican, we collected data starting in January 2020 in order to focus on the most recent and critical events: the protests following George Floyd's murder[11], and the 2020 USA presidential election[12].

Data cleaning

We first excluded empty comments, comments from deleted authors, comments that were hidden for user privacy reasons, and comments containing hyperlinks. Within each body of text (posts and comments), we removed paragraph breaks and replaced a small set of special characters (for example, we replaced *&amp* with *and*). Despite

[5] https://pushshift.io/
[6] https://www.reddit.com/r/BlackLivesMatter/
[7] https://www.reddit.com/r/brexit/
[8] https://www.reddit.com/r/climate/
[9] https://www.reddit.com/r/democrats/
[10] https://www.reddit.com/r/Republican/
[11] https://en.wikipedia.org/wiki/George_Floyd
[12] https://en.wikipedia.org/wiki/2020_United_States_presidential_election

Subreddit	Description	Creation date	# members
r/BLM	discusses news related to the Black Lives Matter movement .	2014	109K
r/Brexit	aims to foster debate about the United Kingdom's (UK) exit from the European Union (EU).	2014	53K
r/Climate	is a community for truthful science-based news about climate and related politics and activism.	2008	99K
r/Democrats	is a partisan subreddit. It aims to discuss political news, policies and how to ensure the election of Democratic party candidates.	2014	292K
r/Republican	is a partisan subreddit for Republicans to discuss issues with each other.	2008	172K

Table 4.1: **Subreddits**. The description of each subreddit is provided as written on Reddit, along with their member counts as of August 2021.

the presence of potentially offensive content in the discussions, we purposefully did not filter it out in order to capture online discussions in the most organic way.

Selecting graph interactions for annotation

Data annotation is commonly limited by time, human resource, and financial constraints. As a consequence, we must select the most informative comment pairs to annotate to serve our objectives. We look for comments most likely to express clear stances, be argumentative, and deal with key discussion topics. We apply a series of filters to retain the most argumentative content. Tables 4.2 and 4.3 display the share of data affected by each filtering step among the posts and the comments respectively.

We first assume that posts with higher engagement (in terms of number of comments) are more likely to be informative because they either deal with topics or propose argumentative elements other individuals are willing to engage with. A large number of posts receive no or a few comments. As a matter of fact, 92%, 45%, 83%, 75%, and 59% of posts received less than 5 comments in r/BLM, r/Brexit, r/Climate, r/Democrats, and r/Republican respectively. We also observe that a majority of short comments and posts express neutral stances, and are not as informative as longer content. For these reasons, we removed every post with fewer than 10 words and with

fewer than k comments, where k is the rounded average number of comments per post in each subreddit ($k = 2, 10, 5, 5, 5$ in r/BlackLivesMatter, r/Brexit, r/Climate, r/Democrats, and r/Republican respectively). We also filtered out comments with fewer than 10 or more than 100 words, and comments that contain hyperlinks. Annotation time constraints motivated our choice to filter out comments with more than 100 words. Through manual inspection of comment pairs across all subreddits, we deemed it sufficient for the human annotators to contextualize a comment with its 100 first words.

In a comment-reply pair, we call *child comment* the replying comment, and *parent comment* the comment it replies to. After further inspection of the tree structure of Reddit discussions (Figure 4.1), it became apparent that discussion threads between users with different opinions most often devolve into an onslaught of negative affronts with little substance. Conversely, conversations between people with similar opinions tend to contain only a few nested comments. These observations motivated our decision to retain only comment-reply pairs whose parent comment replied directly to the initial post (tree level 1), as proxies of lengthier discussion threads between two given users. We also remove comments whose authors have been deleted or whose contents have been removed. We retained all posts and comments which passed all the filters.

	r/BLM	r/Brexit	r/Climate	r/Dems	r/Reps
Posts with <k comments*	70%	58%	75%	71%	71%
Posts with <10 words	52%	32%	30%	43%	44%

Table 4.2: **Filtering rates at each step of processing the posts**. r/Dems and r/Reps are abbreviations for r/Democrats and r/Republican, respectively.

*where $k = 2, 10, 5, 5, 5$ for r/BlackLivesMatter, r/Brexit, r/Climate, r/democrats, and r/Republican respectively.

	r/BLM	r/Brexit	r/Climate	r/Dems	r/Reps
Comments with <10 words	34%	21%	21%	32%	31%
Comments with >100 words	6%	12%	14%	5%	5%
Comments with hyperlinks	6%	6%	14%	4%	6%
Comments with tree level >2	35%	55%	45%	41%	42%

Table 4.3: **Filtering rates at each step of processing the comments**. r/Dems and r/Reps are abbreviations for r/Democrats and r/Republican, respectively.

Graph creation

For each subreddit r/*, the resulting set of interactions forms a temporal graph $\mathcal{G}_{r/*}$, where nodes are users, and edges represent a comment-reply interaction between two users. Two users can have several interactions at different times. Because $\mathcal{G}_{r/Brexit}$ had significantly more comment-reply pairs than the other forums, we retained only the users (nodes) who at least commented on ten posts over the course of a given month. We keep all comment-reply pairs for all the other forums. The number of nodes and edges in each graph is provided in Table 4.5. In fine, the cleaned dataset contains 49,140 comment-reply pairs, forming a temporal user interaction graph for each subreddit.

Ethics and anonymity

Data from Reddit is anonymous in principle, as users choose usernames which are not associated with any personally identifiable information. We acknowledge that even anonymous use of social media poses potential risk due to accidentally revealing personal information. For this reason, we pledge to maintain a policy similar to Pushshift API which enables anyone to request the removal of their posts and usernames from our dataset. We also ensured that the data is used in compliance with Reddit's data policies.

4.3.2 Annotation

Crowdsourcing

Comment-reply pairs were annotated with *agree*, *neutral*, *disagree*, or *unsure* labels by a team of 529 English-proficient annotators from Scale AI[13]. We provided all annotators with introductory courses on the BlackLivesMatter and climate change movements, and on UK and US politics. We also provided them with examples for each of the four labels, with the reasoning behind the labels attributed to each example. Scale AI also ran internal training webinars with training courses and quizzes based on the instructions and a sample golden dataset (74 comment-reply pairs) we curated. A sample of our 10-page instruction document is presented in Appendix. The final group of annotators was selected based on their performance on the golden dataset. Based on that performance, each selected annotator was attributed a confidence score that impacts the confidence given to their annotations

[13] https://scale.com/

in the production phase. Every annotator has been compensated above minimum wage in their respective labor markets.

Labelling and inter-annotator agreement

Each task consists in annotating one comment-reply pair as *agree*, *neutral*, *disagree*, or *unsure*. The annotator interface is depicted in Figure 4.2. The annotator is provided with the name of the subreddit, the initial post (topic), the comment-reply pair, and a list of subreddit-specific abbreviations for reference.

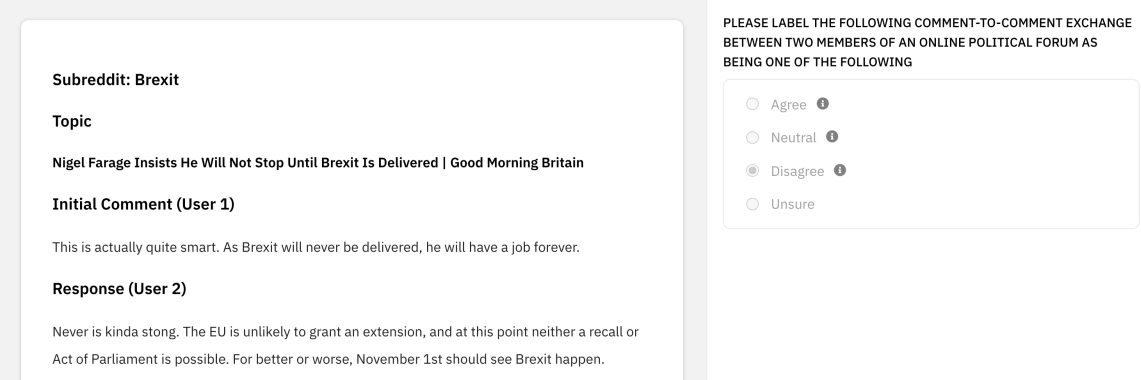


Figure 4.2: **Interface for annotators**. For each labeling task, the subreddit, the initial post (topic), the parent (initial) and child (response) comments are displayed to the annotator.

Each task is annotated by three to five annotators based on Scale AI's dynamic consensus process: the number of annotators per task is determined by averaging the confidence score of the annotators and comparing it against a minimal desired confidence threshold score. Starting with three annotators, more annotators are added until the minimal confidence threshold is reached. For each task, the final label is the majority class. Tasks where the annotators equally chose *agree*, *neutral* and *disagree* were reviewed by Scale AI's internal subject experts before finalizing the response. Overall, 33% of the annotations have full inter-annotator agreement. We drop the 6,246 comment-reply pairs whose final label is *unsure*, and the pairs whose inter-annotator agreement score is strictly lower than $\frac{2}{3}$. The final dataset contains 42,894 annotated interactions. The data is publicly available at `https://scale.com/open-datasets/oxford`, with adequate instructions to use the files. An edge contains the parent and child comments, their timestamps, the related post, and the related subreddit. A comprehensive description of the data with the column denominations is provided in Appendix.

Post	Parent comment	Child comment
Serious question: where does the idea that the EU will only make a Brexit deal at the last minute come from?	Misplaced negotiation tactics. It cannot work when 27 countries have to agree a new deal. The UK negotiators seem to think that they are only negotiating with a few suits who have the power to override 27 governments.	Yes. They think EU is only Germany and France. Not 27 independent countries in one Block.
Why are Brexiters so angry and furious?	Because the EU has politely declined to fulfill the promises they made?	The EU is fulfilling its promises and helping the UK achieving Brexit
Farmers sue German government over missed climate targets	The irony of dairy farmers suing over environmental damage is absolutely laughable	Pardon my ignorance, are dairy farmers causing lots of environmental damage?

Table 4.4: **Samples of** `DEBAGREEMENT`: an agreement in r/Brexit (top row), a disagreement in r/Brexit (middle row) and a neutral interaction in r/Climate (bottom row).

4.3.3 Analysis

Table 4.5 displays the characteristics of each subreddit-specific graph. r/Brexit is the densest graph, with the smallest number of nodes (722) and the largest number of edges (15,703). All subreddits have a substantial number of neutral edges, ranging from 22% to 29%. r/BLM and r/Democrats have the largest amount of agreements (45% and 42%, respectively). r/Brexit, r/Climate, and r/Republican are more polarised and display 42%, 40% and 40% of disagreements. The user life span, i.e. the number of days between the first and last interaction of a user, averages at more than a year for r/Brexit and almost 4 months for r/Climate. For r/BLM, the low average life span (11 days) can be explained by the death of George Floyd that sparked a high but short-term peak of engagement in the subreddit (see below, Figure 4.3). Similarly, the life spans in r/Democrats and r/Republican average at 41 and 27 days respectively, as we are only following the events surrounding the 2020 USA Presidential Elections. In fact, except in r/Brexit where each user interacted more than once and 43 times in average, most users only engage once in the other temporal graphs. 76%, 61%, 58% and 63% users only engage once in r/BLM, r/Climate, r/Democrats and r/Republican respectively. As a result, the number of individuals a user engages

	r/BLM	r/Brexit	r/Climate	r/Dems	r/Reps
Start date	Jan 2020	Jun 2016	Jan 2015	Jan 2020	Jan 2020
#nodes	2,512	722	4,575	6,915	8,814
#pairs	1,916	15,703	5,754	9,606	9,791
+	45%	29%	32%	42%	35%
-	22%	29%	28%	22%	25%
◦	33%	42%	40%	36%	40%
full agreement rate	42%	35%	41%	41%	41%
mean life span (days)	11	392	111	41	27
1-day life span	87%	0%	71%	70%	75%
mean # interactions	1.5	43	2.5	2.8	2.2
1-interaction users	76%	0%	61%	58%	63%
mean # respondents	1.5	33	2.5	2.7	2.2
1-respondent users	76%	0%	61%	58%	63%

Table 4.5: **Graph characteristics**, split per subreddit. + / - / o represent the number of positive, negative and neutral pairs, respectively. The start date is the first month of interactions in each subreddit dataset. The full agreement rate is the proportion of pairs whose labels are identical for all annotators.

with (number of respondents) is quite low in average: 1.5 for r/BLM, between 2 and 3 for r/Climate, r/Democrats and r/Republican.

The limited amount of interactions per user makes it particularly challenging to understand individuals' opinions (edge signs) with confidence, and makes it even more challenging to find communities of users with similar opinions. Unfortunately, such sparsity is typical of real-world graphs and adapted methods must be designed to handle it. In the next section 4.4, we explore the use of multimodality (graph structure and text) to predict (dis)agreements more efficiently in such settings when informative signal is scarce.

Figure 4.3 displays how the amount and the nature of interactions varies over time in `DEBAGREEMENT`. We observe activity peaks matching key events relating to the topics of the subreddits. In r/BLM, the peak of interactions occurs during the protests following the murder of George Floyd on 25 May 2020. The highest peak of activity in r/Brexit coincides with the official Brexit[14], i.e. the withdrawal of the United Kingdom from the European Union on 31 January 2020. It is also the time at which the percentage of disagreements is at its highest, and much higher than the

[14] https://en.wikipedia.org/wiki/Brexit

agreement rate. In r/Democrats and r/Republican, we observe activity spikes around US Election Day on 3 November 2020.

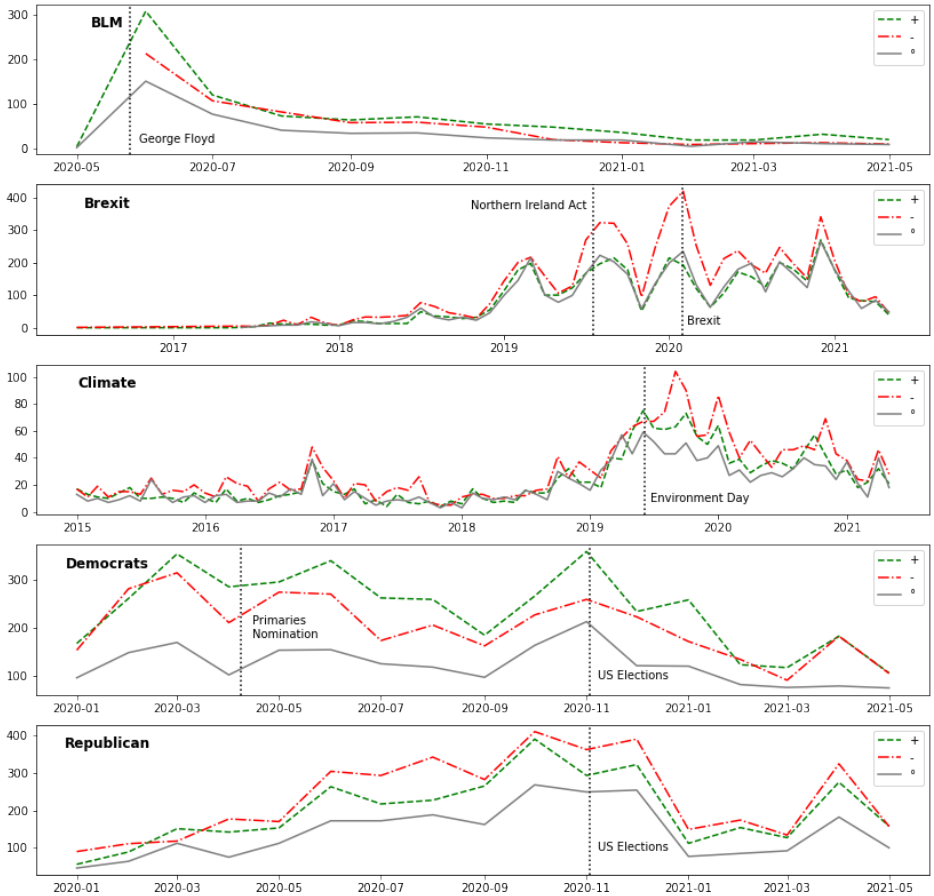


Figure 4.3: **Monthly evolution of interactions per subreddit**. + / - / o represent the number of positive, negative and neutral interactions, respectively.

To understand the linking behaviours, we can inspect the way triads form over time, and how they confirm or contradict social balance theories (see Chapter 2). Table 4.6 depicts the statistics in each graph. We order the edges temporally and build the temporal graph sequentially by adding edges one at a time. When a new edge appears between two nodes who already interacted in the past, we replace the old edge by the new one. We track each time an edge closes a triad and record the

nature of the triad it has formed. We observe three orders of cluster coefficients[15]: high (10^{-1} for r/Brexit), medium (10^{-2} for r/Climate and r/Democrats), low (10^{-3} for r/BLM and r/Republican). Only one (balanced) triad formed in r/BLM across the study. All graphs record at least 50% of strongly balanced triads except r/Climate (49%). By relaxing the strong balance assumption, all graphs have at least 70% of (weakly) balanced triads, except r/Democrats with 58% of such triads. Therefore, the evolution of triads is skewed towards forming (weakly) balanced triangles. This means that even if we do not observe all associations (edges), we can infer them using prior belief on the tendencies of triad formation (see next section 4.4).

	r/BLM	r/Brexit	r/Climate	r/Dems	r/Reps
# nodes	2,512	722	4,575	6,915	8,814
(S)	2,058	720	3,731	5,905	7,337
# edges	1,893	11,804	5,620	9,465	9,716
(S)	1,475	8,898	4,088	7,408	7,334
# triads	2	124,698	794	680	102
(S)	1	55,311	305	390	47
	0%	21%	26%	5%	2%
	100%	42%	40%	22%	34%
	0%	29%	25%	42%	28%
	0%	8%	9%	31%	36%
strong balance	100%	50%	49%	53%	70%
weak balance	100%	71%	75%	58%	72%
clustering coeff.	0.002	0.3	0.01	0.008	0.002
(S)	0.001	0.2	0.009	0.006	0.002

Table 4.6: **Triads in** `DEBAGREEMENT`. The (S) indicator stands for the graph resulting from only keeping the signed interactions (i.e. removing the neutral edges).

4.4 Triadic closure and language modeling

In this experimental Section, we exploit `DEBAGREEMENT` to investigate whether combining the observed evolution of the temporal graph and the textual information of

[15]The clustering coefficient of a node quantifies the node's ability to form triangles (triads) with its neighbours. It varies between zero and one. When the coefficient is one, it means that a node forms triangles with all its neighbours (i.e. a clique [102]).

the edges can predict the nature of unobserved edges in such signed temporal graphs. By doing so, we challenge linear algebra approaches to the sign edge prediction task. We show that solving this task with a multimodal approach is more efficient. We invariably use the terms (dis)agreement detection and sign edge prediction to describe the task of classifying the nature (or sign) of an interaction.

4.4.1 Evaluating pre-trained language models

The starting point is to evaluate the performance of four state-of-the-art pre-trained language models (LMs) on the (dis)agreement detection task: BERT [37], RoBERTa [90], DeBERTa [51] and XLNet [160]. The LMs are trained only on text data. As a common practice, we build training samples by concatenating the parent sequence, the $[SEP]$ token, and the child sequence of each comment-reply pair[16]. We split the data into 80%/10%/10% train/validation/test sets while maintaining the temporal order so that the validation and test are performed on the most recent data. This follows a realistic setting in which one uses a trained model to perform prediction on new, incoming temporal edges [80]. We train each model three times with different seeds. All experiments were performed on one NVIDIA T4 16Gb GPU. We use the HuggingFace implementation of the language models we evaluate. All models perform with an average accuracy around 62%. Due to their relatively similar performance, we experiment with BERT for the remainder of the section.

BERT	XLNet	RoBERTa	DeBERTa
62.1% (0.9%)	61.9% (1.2%)	62.2% (0.9%)	62.6% (0.7%)

Table 4.7: **Accuracy of large language models (LLMs) for (dis)agreement detection**: standard deviations are in brackets. We use the base, uncased versions of each model.

The confusion matrix (Table 4.8) shows that the model performs the worst at identifying neutral interactions. The result is intuitive as neutral interactions are often the sign of lack of engagement, or express both partial agreement and disagreement, unlike (dis)agreements which show marked support or animosity between the protagonists.

[16] $[SEP]$ is a special token used in natural language processing (NLP) and machine learning (ML) models to mark the separation between different segments of text. See, for instance, [37].

True Label	Predicted Label: -	0	+
-	75%	11%	12%
0	28%	45%	27%
+	17%	14%	69%

Table 4.8: **Confusion matrix of BERT**. The model is under-performing on neutral interactions.

Overall, there is clear room for improvement on the performance of the trained models. Predicting the nature of the interactions is particularly difficult for LMs as `DEBAGREEMENT` spans a large range of writing styles, language levels, and topics. State-of-the-art LMs still fail to capture all subtleties of human dialogue especially in the presence of complex or nuanced interactions [41, 121].

4.4.2 The power of triadic closure: `siBERT`

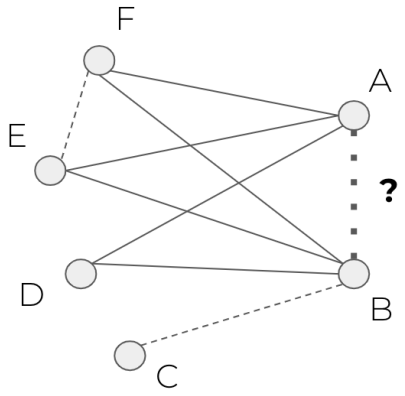


Figure 4.4: **The role of triadic closure in edge sign predictions**. Full-line edges belong to triangles involving A and B. Dotted-line edges are not used for predictions.

Facing the limited predictive power of LMs on this complex task, we design a simple experiment to explore the power of the triadic closure patterns (Table 4.6) observed during training, to infer edge signs on the test set. For this experiment, we drop all neutral edges, and retrain BERT for two-way classification (positive or negative interactions). In parallel, we estimate the share of each observed triad type on the training set in each subreddit. These shares are set as triadic probabilities on the test set.

At test time, we infer the edge signs by averaging the output probability of BERT and the probability estimated by triadic closure. We call this approach `siBERT`, for *structure-informed BERT*. As a matter of example, let us consider the example depicted in Figure 4.4. The full and dashed lines are past interactions, and the dotted line is a new interaction AB we want to infer the sign of. The full-line edges are all part of a triangle the edge AB is going to close. Suppose AD is positive and DB is negative. Triangle ADB can either be of type $T2$ (if AB is positive) or $T1$ (if AB is negative) - refer to Figure 2.1 in the Preliminaries for the definition of each triangle. Given their respective probabilities of appearance (observed during training) p_{T2} and p_{T1}, we estimate the sign of AB, given triangle ADB as:

$$\mathbb{P}(AB = +1 \mid ADB) = \frac{p_{T2}}{p_{T2} + p_{T1}}$$
$$\mathbb{P}(AB = -1 \mid ADB) = \frac{p_{T1}}{p_{T2} + p_{T1}}$$

Because edge AB is part of three triangles, we model the triad-based probability of AB as the average of the three related probabilities:

$$\begin{aligned}\mathbb{P}_{tr}(AB = +1) &= \mathbb{P}(AB = +1 \mid ADB, AEB, AFB) \\ &= \frac{1}{3}\left[\mathbb{P}(AB = +1 \mid ADB) + \mathbb{P}(AB = +1 \mid AEB) + \mathbb{P}(AB = +1 \mid AFB)\right]\end{aligned}$$

Eventually, the sign probability of AB is the average of its triad-based and BERT-based probabilities:

$$\mathbb{P}(AB = +1) = \frac{\mathbb{P}_{tr}(AB = +1) + \mathbb{P}_{BERT}(AB = +1)}{2}$$

Table 4.9 compares the accuracy of BERT and its combination with triadic closure. r/BLM is left out of the experiment as it contains only one triangle (see Table 4.6). The accuracy improves significantly across all subreddits, which highlights the strong benefits of complementing traditional network science approaches with multimodal data such as textual information.

4.5 Conclusion

In this chapter, we introduced `DEBAGREEMENT`, a dataset of 42,894 comment-reply interactions annotated through crowdsourcing for (dis)agreement detection, and forming 5 temporal graphs. The 5 graphs display different temporal evolution, density, node engagements, and structural balance patterns. We evaluated the performance

	r/Brexit	r/Climate	r/Dems	r/Reps
BERT	79%	77%	74%	83%
`siBERT`	82%	80%	79%	90%

Table 4.9: **Accuracy of BERT v `siBERT`, its combination with triadic closure**. The introduction of triadic closure significantly improves the performance and highlights the benefits of combining structural signals with language understanding approaches.

of several state-of-the-art large language models to predict (dis)agreements, and show that there is a large room for improvement. For this purpose, we provide a structural approach based on triadic closure: the past interactions of two individuals with their respective neighbourhood inform their propensity to (dis)agree with each other. `DEBAGREEMENT` not only opens new avenues for exploiting text as heterogeneous data in network science tasks, but provides opportunities to test social theories, quantify online polarisation, and study how people express opinions and change their stance on social media.

4.A Appendix

Data format

The dataset contains the following columns:

- The *subreddit* in which the interaction took place.
- The *exact_time* is the UTC timestamp at which the interaction occurred.
- The *msg_id_parent* (*msg_id_child*) is the unique Reddit identifier for the parent (child) comment.
- The *body_parent* (*body_child*) is the text body of the parent (child) comment.
- The *author_parent* (*author_child*) is the Reddit username of the parent (child) author.
- The *submission_id* is the unique Reddit identifier for the post the interaction occurred on.
- The *submission_text* is the text body of the posts.
- The *label* denotes as *disagree* (0), *neutral* (1), *agree* (2).
- The *agreement_fraction* is the fraction of annotators that agree with the final *label*.
- The *individual_kappa* is the Fleiss' kappa factor of the annotation ([21], Appendix A.2).

Annotation instructions

Figures 4.5, 4.6, 4.7 contain the first pages of the instructions provided to human annotators.

Oxford Spec Document*Categorization*

Summary of Task

- Compare two comments which relate to the same political subreddit.
- Label whether they **agree** with one another, whether they are **neutral** towards each other, or whether the comments **disagree** with one another.
- Select **unsure** If it is not possible to make a decision based on the information at hand.
    - **"Agree"** : the reply shows approval towards the initial statement or initial author
      Some examples of phrases that express agreement are: "I agree", "That's absolutely right", "That's so true", "Exactly", "That's how I feel", etc.
    - **"Neutral"** : there is a topical exchange between authors but agreement / disagreement is not expressed.
    - **"Disagree"** : the reply shows disapproval or dislike towards the initial statement or initial author.
      Some examples of phrases that express disagreement are: "No way", "I don't think so", "Not necessarily", "That's not always true", etc.
    - **"Unsure"** : It is not possible to make a decision based on the information at hand.

I.e. What side do we predict the initial author is on? → Does the reply support this opinion? OR → Do you think the reply author would support this opinion? → If neither of these, select **"Neutral"** or **"Unsure"**

Labels	**Attributes**
Agree	The reply shows approval towards the initial statement or initial author. Some examples of phrases that express agreement are: "I agree", "That's absolutely right", "That's so true", "Exactly", "That's how I feel", etc.
Neutral	There is a topical exchange between authors but agreement/disagreement is not expressed.

Figure 4.5: **Instructions for annotators** - summary of task.

Disagree	The reply shows disapproval or dislike towards the initial statement or initial author. Some examples of phrases that express disagreement are: “No way”, “I don’t think so”, “Not necessarily”, “That’s not always true”, etc.
Unsure	It is not possible to make a decision based on the information at hand.

Recommended Workflow:

- Read the Subreddit at the top of the task to understand the **context** of the discussion.
- Read the **“Initial Comment”** to understand the **content and opinion** of the person.
- Read the **“Response”** to determine whether the responder **agrees**, **disagrees** or is **neutral** towards the **“Initial Comment”.**
- If it is **impossible** to determine whether the responder agrees, disagrees, or is neutral towards the Initial Comment, select **“Unsure”.**

Rules and Tips:

- **Try to understand what each person is saying:**
  - If you don’t understand some of the internet slang being used, google it to find out (e.g based, gaslighted, blue-pilled, red-pilled)
  - Many of these exchanges will be very political (US/UK) so **the more you know about the political people/ideas involved, the better**
  - Check the **“Helpful Abbreviations”** Key at the bottom of each task to explain common Abbreviations used like: **GOP** (Grand Old Party / Republican Party), **POC** (Person of Color), **POTUS** (President of the United States), etc.
  - It’s important to clearly understand what is being said in order to determine if the two comments agree with one another.

Key information to look for:

- **Be on the lookout for keywords in the response that reveal the opinion. This can be key in determining how to label the task.**

Figure 4.6: **Instructions for annotators** - workflow and rules.

- When a clarifying **question is posed** in the **Response,** it can be an indication that the label is **Neutral**
- Look for **Neutral questions** that discuss the same topic but don't reveal the author's stance. This can be a great indicator for a **Neutral** label
  - Here's an example of a **Neutral** Response with a question in it:

Neutral Example	
Initial Comment: I cannot think of anything, but I'd be very interested to hear the post-mortem.	**Response:** Of the conversation or the country?

- **There can be exceptions to the above rule of thumb:**
  - In addition to being neutral, questions in the **Response** can also be sarcastic or angry, showing **Disagreement.**
  - Keep this in mind when reading a question in the **Response.** Not all questions are **neutral**.
- "Yes!", "I agree", "Exactly" are all great identifier words to show if the **Response agrees** with the **Initial Comment.**
- "No", "I don't think so", "But", "Nah" can show if the **Response disagrees** with the **Initial Comment.**

Don't do this:

- **Do Not** guess the answer when you're unsure.
  - If you're **uncertain** whether the two comments agree or not, or whether they are neutral towards each other, select **"unsure"**.
  - Some comments have nothing to do with one another, for these select **"unsure"** like this example:

Unsure Example	
Initial Comment: C c c c.COMBO BREAKER!!	**Response:** Ultraaaa Ultraaaaa Ultraaaa

- **Do Not** jump to conclusions
  - If you don't understand some of the words or phrases used in the comments, look it up on google.

Figure 4.7: **Instructions for annotators** - key information to look for.

Chapter 5

Louvain method for signed graphs

5.1 Introduction

In Chapter 4, we examined triads to understand associations in local structures. In Chapter 3, we learnt long-range associations by using random walk processes. We complete our study in this chapter by unfolding node associations at the macro scale through community detection in signed graphs.

In recent years, several methods have been presented to detect communities in (static) signed graphs. These methods are either based on random walks, spectral clustering, generative methods, or social balance theory. Methods based on random walks [159, 166] generalise the concept of random walks on unsigned graphs to signed graphs by defining a non-zero probability to walk on negative edges. Spectral clustering algorithms [29, 30, 24, 95, 96] exploit the eigenvalues of the signed laplacian. In these methods, the number of clusters is fixed a priori. Generative methods [62, 22, 158, 163, 89] propose Stochastic Block and Probabilitic Mixture Models to model non-overlapping and overlapping communities. Methods based on social balance theory [145, 5, 132] use the notions of frustration minimisation and/or modularity maximisation [44, 101] to find communities.

In this chapter, we consider the problem of community detection in signed graphs. Specifically, we study generalisations of the Louvain method [15], a popular and efficient method for modularity optimisation in unsigned graphs. We propose a new heuristics-based method, `Signed Louvain`, to maximise the signed modularity function [145]. We then expand the heuristics to temporal signed graphs. Through extensive synthetic experiments, we show the advantages and limitations of our proposed method. Eventually, we apply our model to unveil opinion dynamics in a real-world debate graph from `DEBAGREEMENT`.

5.2 Louvain on signed graphs

In Section 5.2.1, we describe the Louvain algorithm [15] and how it has been generalised to be applied on signed graphs. In Section 5.2.2, we evaluate the performance and caveats of the approach on signed graphs.

5.2.1 Approach

Given a signed, undirected, weighted graph $\mathcal{G}$, with n nodes, we note $A \in \mathbb{R}^{n \times n}$ its adjacency matrix. We decompose $A = A^+ - A^-$ where A^+ and A^- have non-negative entries and represent the positive and negative parts of the graph respectively.

Given this graph specification, the Louvain algorithm [15] detects communities on unsigned graphs ($A^- = 0$) using a modularity maximisation-based approach. The modularity of an unsigned graph partition is a scalar value between -1 and 1 that measures the density of links (edges) inside communities as compared to its expectation with respect to a null model. Based on the homophily assumption, the intuition is that a well partitioned graph should have most links inside communities and not between communities. Given an unsigned graph with adjacency matrix A, the (generalised) modularity of partition σ is defined as:

$$Q(\{\sigma\}) = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{5.1}$$

where A_{ij} is the edge weight between i and j, $k_i = \sum_j A_{ij}$ is the weighted degree of node i, c_i is the community to which node i is assigned, m is the sum of all edge weights, γ is a resolution parameter [124] - the original expression for modularity is recovered when $\gamma = 1$ [44] - , and δ-function is the indicator function.

Finding the partition that maximises Equation (5.1) is a NP-hard problem [17], so different heuristics have been proposed for its optimisation. Among those, the Louvain method uses efficient heuristics that approximate the best solution. Starting with a partition where each node is in its own community, it alternates between two phases until convergence: (1) a *move* phase where each node, called *target node*, is moved in the neighbouring community (i.e. community of its neighbours) that maximises modularity increase, (2) once no move increases modularity anymore, an *aggregate* phase consisting in building a new graph whose nodes are now the communities found during the first phase, and edges are the sum of link weights between these communities. It iterates between the two steps until no improvement on modularity is achieved. One bottleneck is that the method may converge to a local maxima

which is not a global maxima. It is therefore preferable to run the method several times to find the most optimal configuration.

Originally proposed for unweighted, undirected graphs, the modularity function has been extended to more general classes of graphs, together with the algoritms required for their optimisation, including Louvain [14]. In the case of signed graphs, a signed modularity function has been defined [145] as:

$$Q(\{\sigma\}) = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \left(\gamma^+ \frac{k_i^+ k_j^+}{2m^+} - \gamma^- \frac{k_i^- k_j^-}{2m^-} \right) \right] \delta(c_i, c_j) \tag{5.2}$$

where $k_i^\pm = \sum_j A_{ij}^\pm$ is the weighted degree of node i on the positive (resp. negative) part of the graph, $\gamma^\pm$ are resolution parameters, and $m^\pm$ is the sum of all positive (resp. negative) edge weights. This definition of modularity aims to reward configurations with positive internal links and absent internal negative links, and penalise configurations with negative internal links and absent internal positive links. In practice, this function can also be maximised with the original Louvain algorithm[1] or related extensions, as we discuss more in detail below.

5.2.2 Caveats

The preponderance of the negative part in the graph increases the existence of local maxima when optimising modularity Q in Equation (5.2). We note ΔQ the change in modularity occurring from moving an isolated node into a given community. There are graph structures and densities for which ΔQ is negative for any possible move. A simple example shows that this can even occur at the start of the optimisation process: we consider a signed graph with 5 nodes and 4 negative edges connecting node 0 to nodes 1, 2, 3 and 4. Following the Louvain algorithm, the initial partition is made of singletons. The variation in modularity occurring when moving any node in a neighbouring community is:

$$\Delta Q = \frac{1}{m} \left(A_{ij} + \gamma^- \frac{k_i^- k_j^-}{2m^-} \right) = \frac{1}{4} \left(-1 + \frac{1}{2} \gamma^- \right) < 0 \tag{5.3}$$

for usual values of γ, i.e. $\forall\ \gamma^- \in [0, 2]$. As a consequence, the final configuration of the optimisation process is the initial configuration: $\sigma_0 = \{\{0\}, \{1\}, \{2\}, \{3\}, \{4\}\}$.

What final configuration should we expect? In a balanced signed graph, in virtue of the strong structural balance assumption (*the enemy of my enemy is my friend*),

[1] https://louvain-igraph.readthedocs.io/en/latest/

we expect the optimal configuration to be $\sigma = \{\{0\}, \{1, 2, 3, 4\}\}$. This can only occur if target nodes are able to move into the communities of non-neighbouring nodes, which is not permitted in the original Louvain algorithm. An extension of the Louvain algorithm, that we denote Relaxed Louvain from now on[2], allows target nodes to move into any community, whether any neighbouring node belongs to this community or not. In essence, this means that the optimisation process explores significantly more configurations to find the modularity-maximising one. In that setting, the optimisation algorithm terminates at partition $\sigma_f = \{\{0\}, \{1, 2, 3, 4\}\}$. Numerically, for $\gamma^{\pm} = 1$, we confirm that maximum modularity is achieved with σ_f ($Q = 0.5$) and underachieved with σ_0 ($Q = 0.3125$).

In the following experiment, we show that the neighbourhood restriction imposed in the Louvain algorithm makes it converge towards partitions with plenty of small communities (often singletons) in sparse graphs, with and without a preponderant number of negative edges. This type of configurations is not realistic, nor useful, in most real-world cases, e.g., in opinion graphs or online debates where only a few major opinions exist. Relaxed Louvain is a way to lift this restriction and approximate the global maxima more closely. We consider a standard Signed Stochastic Signed Model (SSBM) [54, 62] with 3 communities of size 30, 20, and 10, where all intra- (inter-) community edges are positive (negative). We denote p_{in} and p_{out} the share of existing positive and negative edges respectively. We vary p_{in} and p_{out} between 80% and 0% and generate random graphs repeatedly. For each generated graph, we run Louvain and Relaxed Louvain, and compute the Normalised Mutual Information (NMI) [136, 43] of their final partition with the planted one. The NMI is averaged across all the 10 runs of a given couple (p_{in}, p_{out}).

Figure 5.1 depicts the results of the experiment. We observe the over-reliance of the Louvain algorithm on positive edges such that when the proportion of such edges is low ($p_{in} < 20\%$), the performance of the method dramatically decreases. A closer inspection shows that Louvain converges towards partitions composed of many small groups (2-5 nodes) when p_{in} is low. On the other hand, Relaxed Louvain is able to recover the true community structure even when p_{in} is low and p_{out} is higher, at the only condition that the graph is not too sparse (p_{in} and p_{out} close to 0).

Even though Relaxed Louvain seems to be a promising alternative to Louvain to unveil communities in signed graphs, it also suffers from limitations. The first caveat with Relaxed Louvain is the computational cost: scanning every community for every node at every optimisation step increases dramatically the number of operations

[2] https://louvain-igraph.readthedocs.io/en/latest/

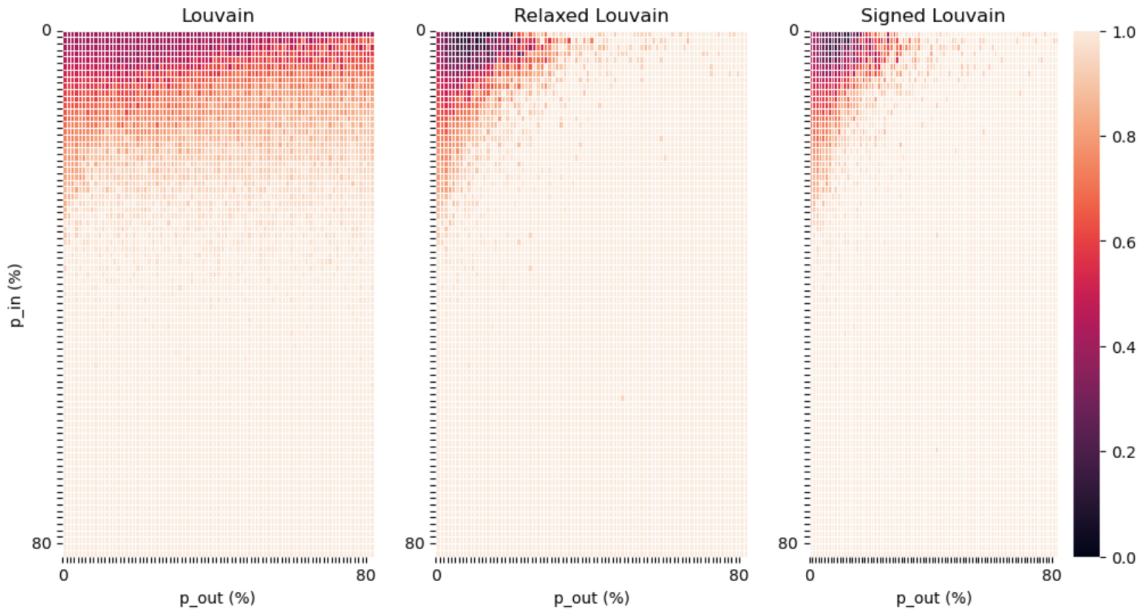


Figure 5.1: **Performance of Louvain v Relaxed Louvain v `Signed Louvain` on a Signed Stochastic Block Model**. The colour indicates the NMI between the final partition and the one used to build the SSBM. As can be seen, the quality of the original Louvain remains low in situations when the density of negative edges between communities is large.

involved. It is prohibitive in real-world settings where graphs are much larger than the synthetic ones considered in this chapter. Moreover, scanning all communities without taking into consideration their proximity to the current node may lead to inconsistent community structures. Indeed, in the example shown in Figure 5.2, from the perspective of structural balance, node 0 can be grouped with nodes 2 and 3, node 3 can be grouped with nodes 5 and 6, but to ensure consistency on the distance between nodes in the same community, node 0 should not be grouped with nodes 5 and 6, without being grouped with nodes 2 and 3 first. However, Relaxed Louvain converges towards configurations such as $\sigma = \{\{0, 5, 6\}, \{2, 3\}, \{1, 4, 7\}\}$ because it ignores the proximity in the graph and randomly selects communities to which a given node can be assigned. In larger graphs, this may bring severe consistency issues where nodes can potentially be in the same community as nodes which they cannot reach via any path.

This behaviour is not expected in unsigned graphs because moving a singleton into the community of a non-neighbouring node cannot increase modularity. However, in the presence of a negative null model, a singleton can join another singleton even if they are not connected. To illustrate this, let us take two non-connected nodes i and

j and look at their pairwise term in the modularity:

$$M_{ij} = \frac{1}{m}\left[A_{ij} - \left(\gamma^{+}\frac{k_i^{+}k_j^{+}}{2m^{+}} - \gamma^{-}\frac{k_i^{-}k_j^{-}}{2m^{-}}\right)\right]\delta(c_i, c_j) \tag{5.4}$$

$$= \frac{1}{m}\left(\gamma^{-}\frac{k_i^{-}k_j^{-}}{2m^{-}} - \gamma^{+}\frac{k_i^{+}k_j^{+}}{2m^{+}}\right)\delta(c_i, c_j) \tag{5.5}$$

as $A_{ij} = 0$. In that case, the modularity gain depends on the relative contribution of the two concurrent null models (positive and negative). In particular, the gain is positive when the negative null model supersedes. In the example shown in Figure 5.2, node 0 has no positive edge so $k_0^{+} = 0$ and only the first term in Equation (5.5) is non-zero, hence $M_{ij} > 0$.

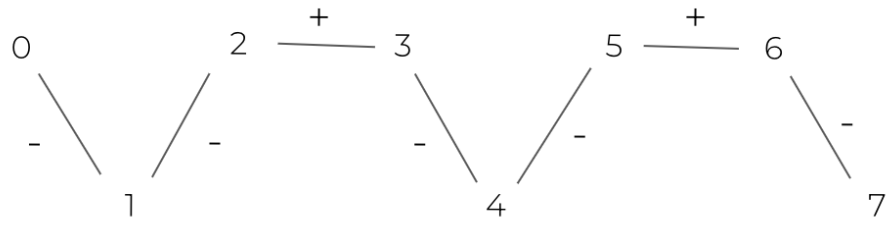


Figure 5.2: **Example of graph where Relaxed Louvain may produce inconsistent communities**, for instance $\{0, 5, 6\}$, due to to its lack of locality during the optimisation.

5.3 `Signed Louvain`: Louvain for signed graphs

In this section, we present `Signed Louvain`, our proposed extension of the Louvain method **for** signed graphs. Unlike Relaxed Louvain, it relaxes the neighbourhood restriction imposed by Louvain while taking into account the proximity in the graph. We will show, through synthetic and real-world experiments, that `Signed Louvain` offers a good tradeoff between speed, consistency and accuracy.

5.3.1 Method

`Signed Louvain` aims to optimise Equation (5.2) by adopting a multiplex graph approach [66] where different moves are allowed or forbidden depending on the layer the current node operates in. In this case, the layers are the positive and negative parts of the graph. We define d_{+} and d_{-} the number of hops a target node is allowed to move on the positive and negative graph to adopt a new community. At the *move*

step, the algorithm skims all the communities within d_+ hops and d_- hops from the target node and selects the one maximising modularity locally.

In order to find the best values for d_+ and d_-, it is important to carefully identify which nodes are likely to be in the same community as the target node that one tries to move. Ideally, the set of these nodes should be as small as possible, to accelerate the algorithm, but not too small, to avoid losing precision. In the case of signed graphs, the archetype of community structure is a k-clusterable graph, which are known to exhibit the weak version of structural balance [146] (refer to section 2.1 for a detailed definition of k-clusterable graphs). For this reason, when searching for nodes that might be in the same cluster as the target node, it is natural to search among the enemies of its enemies, in the negative layer, as those may be similar to the target node. For these reasons, `Signed Louvain` is defined with $d_+ = 1$ and $d_- = 2$, thus restricting target nodes to move either in the communities of their positive neighbours ($d_+ = 1$) or to join communities of the nodes that are 2 steps away in the negative layer ($d_- = 2$). As part of our experimentation, we also consider an extended version of `Signed Louvain` defined for $d_+ = d_- = 2$[3], thus allowing target nodes to move in any community within 2 hops on either layer. The value $d_+ = 2$ allows to consider a bigger set of transitions during the optimisation, and is expected to favour nodes that are in the same triangles as the target node, as those would be simultaneously one and two steps away from the target node. By definition, for $d_+ = d_- = 1$, the classical Louvain algorithm is recovered.

The `Signed Louvain` algorithm proceeds as follows. The first step is to build the d_+ and d_- graphs. In these graphs, a node i is neighbour of all the nodes within d_+ hops or d_- hops in the positive and negative layers of the original graph respectively. The second step is made of two sub-phases (*move* and *aggregate*) repeated sequentially, similar to the original Louvain algorithm. In the *move* sub-phase, we consider moving each target node into the community of any of its neighbours in the d_+ and d_- graphs, as long as it achieves a positive gain in modularity. The node is then placed in the community achieving the maximal positive modularity gain. If no positive gain is possible, the node stays in its original community. The process is applied repeatedly across all nodes until no further local improvement can be achieved. In the *aggregate*

[3]Note that despite the fact that d_+ and d_- take the same value, here 2, the exploration in the positive and in the negative layer are expected to be very different, due to the differences in their structure. The graph of positive connections is expected to present a high clustering coefficient, making the exploration in two steps very similar to that in one step. In contrast, the graph of negative connections is expected to have a small clustering coefficient [140], making the exploration in two steps move away from the direct neighbourhood of the target node.

phase, a new two-layer graph whose nodes are the communities found during the *move* phase is built. The weights of the edges between the new nodes are the sum of the weights of the edges between nodes in the corresponding two communities in each layer of the previous graph. The resulting graph is therefore also a graph with a positive layer and a negative layer. Edges between nodes of the same community lead to self-loops for this community in the new graph. The two sub-phases are repeated until no positive modularity gain is achieved.

The slowest part of the algorithm is the construction of the d_+- and d_--hop graphs in the first step as it requires to scope all d_+- and d_--hop paths starting at every single node of the graph. That being said, by construction, nodes cannot be attached to communities of nodes they cannot reach via a graph path. This property is an advantage of `Signed Louvain` over Relaxed Louvain.

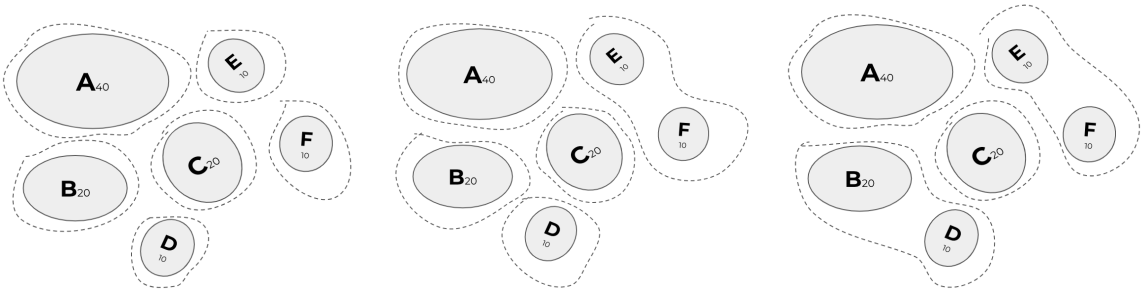
5.3.2 Experiments

In this section, we demonstrate that `Signed Louvain` exhibits a high quality of optimisation while being computationally much faster than Relaxed Louvain.

On the synthetic experiment presented in Figure 5.1, we observe that `Signed Louvain` has a similar performance as Relaxed Louvain. It is able to overcome the over-reliance of the Louvain algorithm on positive edges by exploiting negative edges as well as positive edges. Both `Signed Louvain` and Relaxed Louvain exhibit a drop of performance for very sparse graphs. The following experiment confirms the influence of $\gamma^{\pm}$ as resolution parameters. We consider a SSBM with 6 communities of sizes 40, 20, 20, 10, 10 and 10, with edge density of 15%, where all intra- (inter-) community edges are positive (negative). Figure 5.3 depicts the communities recovered for different values of $\gamma^{\pm}$. Increasing $\frac{\gamma^+}{\gamma^-}$ decreases the minimal size of the communities the algorithm can find [147, 78, 42].

As a next step, we compare the modularity and duration (in seconds) achieved by the methods on real-world graphs. We note L for Louvain, RL for Relaxed Louvain, SL_d the default version of `Signed Louvain` ($d_+ = 1$, $d_- = 2$) and SL_e its extended version ($d_+ = d_- = 2$). We use $\gamma^{\pm} = 1$ for all methods. We experiment with all 5 graphs of `DEBAGREEMENT` (BlackLivesMatter, Brexit, Climate, Democrats, Republican - see Chapter 4). Additionally, we arbitrarily selected topic-specific graphs from `BirdwatchSG` and `TwitterSG`(see Chapter 3). From `BirdwatchSG`, we retain the COVID-19 graph. From `TwitterSG`, we retain the graphs related to Joel Embiid[4]

[4] https://en.wikipedia.org/wiki/Joel_Embiid



(a) $\gamma^+ = 1, \gamma^- = 0.5$ (b) $\gamma^+ = \gamma^- = 1$ (c) $\gamma^+ = 0.5, \gamma^- = 1$

Figure 5.3: **Influence of resolution parameters $\gamma^\pm$ on community sizes**. The higher γ^+, the smaller the communities. The higher γ^-, the larger the communities.

	nodes	edges (+%)	density	average distance	diameter
BLM	459	461 (58%)	10^{-3}	12	34
Brexit	717	8,486 (42%)	10^{-2}	2.4	5
Climate	2,694	3,476 (43%)	10^{-3}	5	18
Democrats	4,782	6,749 (54%)	10^{-3}	5	15
Republican	5,051	5,985 (47%)	10^{-3}	7	20
Joel Embiid	10,640	15,180 (93%)	10^{-4}	5	17
Simon Biles	12,494	13,362 (61%)	10^{-4}	6	23
Zinedine Zidane	9,399	12,554 (96%)	10^{-4}	5	19
Novak Djokovic	4,080	8,044 (85%)	10^{-3}	5	17
Covid-19	1,682	6,648 (66%)	10^{-2}	3	7

Table 5.1: **List of empirical graphs used for the numerical experiments**, with some of their key structural properties. The graphs were collected from `DEBAGREEMENT`, `TwitterSG` and `BirdwatchSG`.

(2023 NBA Most Valuable Player), Simone Biles[5] (23-time Gymnastics World Champion and 4-time Olympics Winner between 2013 and 2023), Zinedine Zidane[6] (1998 Football *Ballon d'Or* and 3-time UEFA Champions League Winner as a coach), and Novak Djokovic[7] (24-time Tennis Grand Slam Winner).

Figure 5.4 depicts the result of the experiments. The algorithm, implemented in Python and open-sourced at https://github.com/lejohnyjohn/signed-louvain, is ran 10 times on each graph. The distance and modularity are averaged across the 10 iterations. We observe that `Signed Louvain` is consistently faster than Relaxed Louvain while also matching or improving its modularity. Speed-wise, `Signed`

[5] https://en.wikipedia.org/wiki/Simone_Biles
[6] https://en.wikipedia.org/wiki/Zinedine_Zidane
[7] https://en.wikipedia.org/wiki/Novak_Djokovic

`Louvain` is competitive with Louvain in most cases. Only on Brexit, SL_d and SL_e are closely on-par with Relaxed Louvain in terms of speed. Across the 10 graphs, `Signed Louvain` is 22 times faster than Relaxed Louvain in average, with a modularity drop of 2% only. We observe that the relative speed of Relaxed Louvain is sensitive to the size and diameter of the graph. Only for Brexit it is able to compete with the other methods speed-wise. Because the average pair distance is only 2.4 in r/Brexit, communities are reachable within the 2-hop neighbourhood of any node, which make it equivalent to `Signed Louvain` speed-wise. For all the others graphs, `Signed Louvain` is significantly faster, while providing a similar performance on most graphs in terms of accuracy. Taken together, these results show that the values $d_+ = 1$ and $d_- = 2$ provide a good compromise between speed and accuracy, which suggests to use them as a default choice when optimising the modularity of signed graphs.

5.4 Community evolution in signed graphs

In this section, we explore the evolution of community structures in time-evolving signed graphs. We adapt the definition of modularity and `Signed Louvain` to account for time evolution. Through synthetic experiments, we demonstrate that `Signed Louvain` is able to detect structural changes such as community births, splits and merges. We also assess the conditions of detection of these events. We conclude by applying `Signed Louvain` to a real-world evolving signed graph.

5.4.1 Related work

Representations of time-evolving graphs

There are historically two ways of representing time-evolving graphs [127]: snapshot and temporal graph representations. In the snapshot representation, the graph history is partitioned into a series of snapshots, each of them being either the state of the graph at a given time or to the aggregation of observed interactions over a time window. On the other hand, the temporal graph representation provides a temporally ordered and timestamped representation of the edges. The latter representation is particularly relevant because it allows a complete and fine-grained description of graph dynamics (for e.g. node appearance and edge removal).

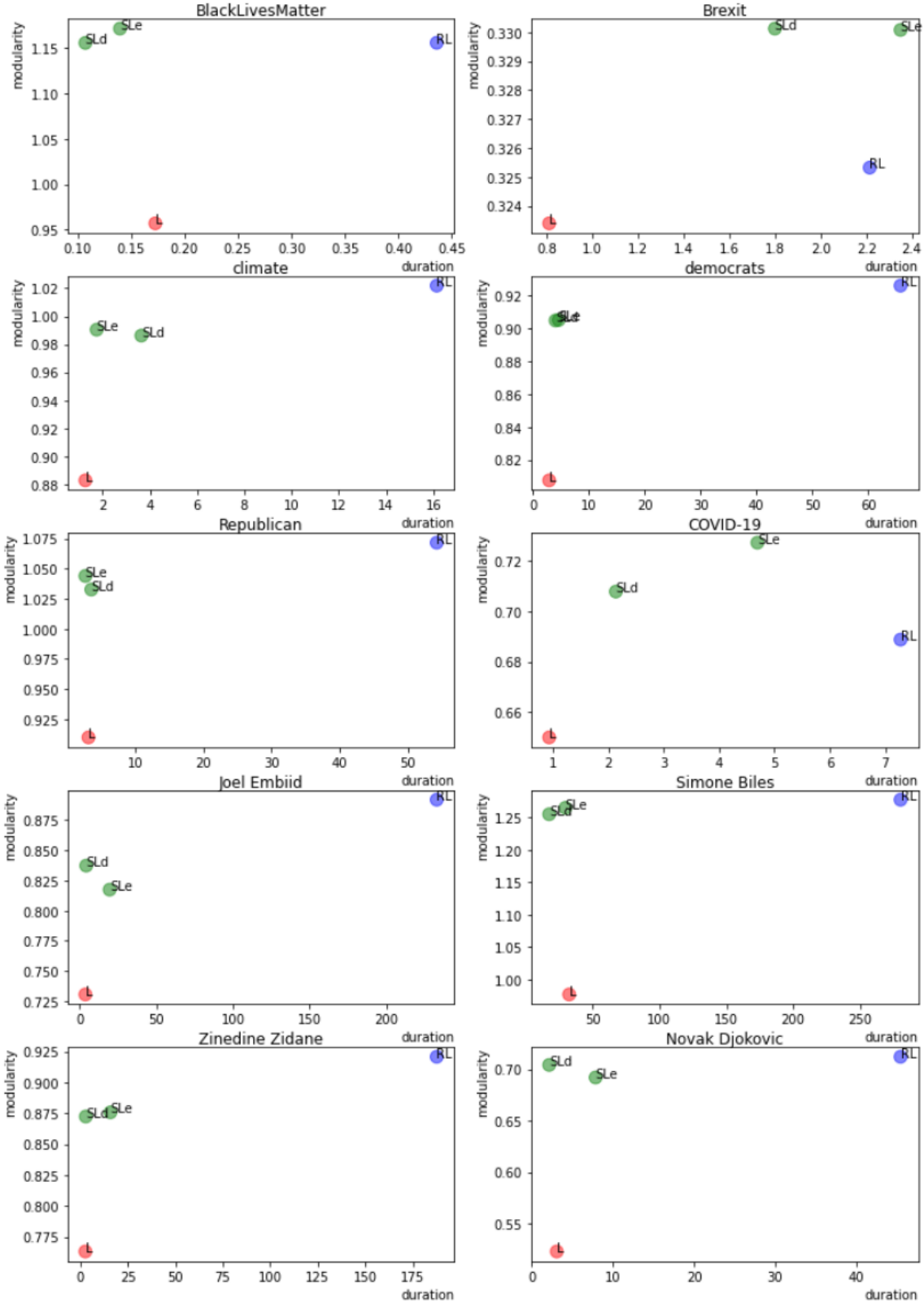


Figure 5.4: **Modularity vs Duration (in seconds)**. L stands for Louvain, SL_d for default `Signed Louvain`, SL_e for extended `Signed Louvain`, and RL for Relaxed Louvain.

Community detection on temporal graphs

Community detection algorithms for temporal graphs are called dynamic community detection (DCD) algorithms in the literature [8, 50, 91]. The most recent survey classifies DCD methods into three categories: instant optimal, temporal trade-off, and cross-time methods [127].

Instant Optimal methods build directly on static community detection algorithms. They consider that communities existing at time t only depend on the current state of the graph at t, so that static algorithms can be applied at each time t. In a second step, communities from different steps are matched by quantifying their similarities. With these methods, it can be difficult to distinguish between the changes due to the evolution of the community structure and the changes due to the instability of the algorithms (static community detection algorithms can find different communities over several iterations on the same graph) [128]. By contrast, temporal trade-off methods find communities for the initial state of the graph then iteratively find communities at step t using the graph at time t and the communities found in the previous step(s). These approaches are able to cope with instabilities while smoothing the community evolution. Eventually, cross-time methods detect communities in a single process by considering all states of the graph at the same time. A single graph, in which each meta-node corresponds to the presence of a node at a given time, is created. Such methods are not able to handle real-time community detection.

All above-mentioned methods are designed for unsigned temporal graphs. In this Section, we design a temporal trade-off method built on `Signed Louvain`. This model has the ability to detect community structures in time-evolving signed graphs, while preserving the continuity of their evolution over the course of time. To the best of our knowledge, this is the first method to detect communities in temporal signed graphs.

5.4.2 Temporal signed graph and modularity

We consider a stream of signed edges $\{e_t\}_t$ appearing in a graph one at a time. The graph is continuously evolving and the goal is to detect communities on-the-fly. An edge e_t involves two nodes i_t and j_t and has positive (resp. negative) weight w_t^+ (resp. w_t^-) so that it can be specified as $e_t = (t, i_t, j_t, w_t^-, w_t^+)$ with $w_t^+ + w_t^- = 1$. We denote G_t the state of the graph at time t.

As edges appear at different times, we assume that the importance of an edge in the evolving graph decreases with its age. By doing so, we mimic the belief that an older edge is less informative than a recent one. This assumption is modelled in

$A(t)$ - the adjacency matrix of the graph at time t - as an exponentially-decaying aggregation of the stream of edges up to time t [138]. Mathematically, for nodes i and j,

$$A_{ij}(t) = A^{+}_{ij}(t) - A^{-}_{ij}(t) \tag{5.6}$$

where

$$A^{+}_{ij}(t) = \sum_{\substack{t_k \leq t: \\ i_{t_k}=i \\ j_{t_k}=j}} w^{+}_{t_k} \exp(-\eta(t - t_k)) \tag{5.7}$$

$$A^{-}_{ij}(t) = \sum_{\substack{t_k \leq t: \\ i_{t_k}=i \\ j_{t_k}=j}} w^{-}_{t_k} \exp(-\eta(t - t_k)) \tag{5.8}$$

where η is a tie-decay parameter controlling the speed at which the importance (weight) of an edge decreases. For this graph specification, the instant modularity on graph G_t is as specified in Equation (5.2):

$$Q = \frac{1}{2m_t} \sum_{i,j} \left[A_{ij}(t) - \left(\gamma^{+} \frac{k^{+}_{i,t} k^{+}_{j,t}}{2m^{+}_t} - \gamma^{-} \frac{k^{-}_{i,t} k^{-}_{j,t}}{2m^{-}_t} \right) \right] \delta(c_{i,t}, c_{j,t}) \tag{5.9}$$

5.4.3 Inference

Signed Louvain for time-evolving graphs

We generalise `Signed Louvain` to temporal signed graphs: starting with a non-empty graph snapshot at $t = 0$, we detect initial communities with `Signed Louvain`. Once an initial community structure has been found, the timed edges are processed one at a time. When a new edge appears, the method first updates the graph according to Equations (5.6), (5.7) and (5.8). Then, starting its initial node partition with the community structure inferred on G_{t-1} at the previous time, it uses `Signed Louvain` on G_t to update node allocations. `Signed Louvain` is unable to create new communities because it can only aggregate communities that exist in its initial partition. To account for the fact that new nodes and communities may appear in a temporal graph, we modify the first *move* phase to allow the nodes to either join any reachable community ($d_+ = 1$, $d_- = 2$) or to create their own (singleton) community if it increases the modularity gain. By doing so, the final configuration at time t can contain more communities than the configuration the algorithm started at (i.e. the final configuration at time $t - 1$). We call this generalisation *temporal* `Signed Louvain` from now on.

Hyperparameters

There are 3 hyperparameters that need to be set in Temporal `Signed Louvain`. We have seen in section 5.3 that γ^+ and γ^- — see Equation (5.9) — control for the size of the communities. Following common practice[42], we set them with parameter search by choosing the values maximising the similarity of communities across several runs of the method. η in Equations (5.7) and (5.8) is the tie-decay parameter and controls for how fast the absolute weight of past edges converges towards 0. The decay is characterised by a characteristic time $\tau = 1/\eta$ which controls the speed at which communities can be found. A higher tie decay (low τ) gives more importance to recent interactions and makes older ones obsolete so that recent structural changes are detected faster. However, decreasing the decay makes the graph denser as edges persist longer.

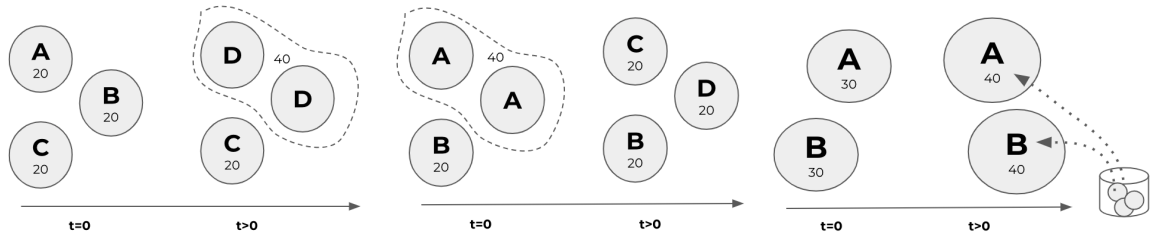
5.4.4 Community events

We now use temporal `Signed Louvain` to detect structural changes, a problem also called *change point detection* [110]. We will see that the tie decay plays an important role in change point detection. Six major structural changes, also called community events [127, 91], can occur in the evolution of a graph: the birth of a community, the split of a community into two or more communities, the merger of two or more communities together, the growth of a community (more nodes join the community), the shrinking of a community (nodes leave the community) and a community death (all nodes vanish).

Experimental design

We design three scenarios that cover the community events mentioned above: a community merger, a community split, and the appearance of new nodes in the graph (Figure 5.5).

Figure 5.5a shows a merger between two communities: at $t = 0$, the graph contains 60 nodes split into three equally-sized communities (**A**, **B** and **C**), and 500 randomly selected edges are generated between the communities, with positive edges inside communities and negative edges between communities. At $t = 1$ onwards, **A** and **B** merge to create community **D**, and each edge created over the course of the next 3,000 time steps are in keeping with this new configuration: at each time step, two nodes are chosen at random and a negative edge is created between them if they are



(a) Merger (b) Split (c) Graph growth

Figure 5.5: **Three scenarios of community dynamics**: a community merger, a community split, and the growth of the graph.

in different communities (i.e. one node is in community **D** and the other is in **C**), and positive if they are both in the same community (**C** or **D**).

The experimental design for community split mirrors the community merger and is displayed in Figure 5.5b: the graph contains 60 nodes initially, with two communities of 40 and 20 nodes respectively and 500 randomly chosen edges in and between the communities. At $t > 0$, the larger community split into two smaller ones of 20 nodes each and 3,000 new edges are randomly created according to this new configuration over the course of the next 3,000 time steps.

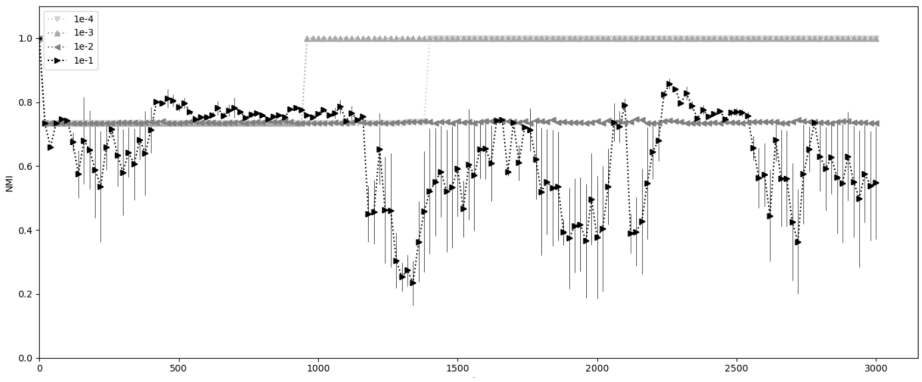
The last scenario simulates the growth of the graph, when new nodes appear in the graph. The initial graph ($t = 0$) has 60 nodes split into two equally-sized communities, with 500 randomly sampled edges between them; again, edges are positive within communities and negative across communities. At $t = 1$, we introduce 20 new nodes with ten of them assigned to each of the two communities, and one interaction per new node. At t=2 onwards, edges are randomly generated with equiprobability, akin to the two previous scenarios. However, this time we introduce noise to the experiment: any edge originated at $t > 0$ is noisy with probability 20%, i.e. has 20% of chances to be negative if it links two nodes in the same community, or 20% of chances to be positive if it links two nodes in different communities. This is displayed in Figure 5.5c with the existence of a few positive (resp. negative) edges between (resp. within) communities **A** and **B**.

We analyse the performance of temporal `Signed Louvain` for these scenarios. For each experiment, we choose the values of $\gamma_{\pm}$ that maximise community similarity across 5 experimental runs, among the combinations $(\gamma_+, \gamma_-) \in \{(1, 1), (1, 0.5), (0.5, 1)\}$. We measure community similarities by computing and averaging the NMI between communities obtained in successive runs of the algorithm. The highest average NMI determines the best pair of parameters. Once the pair of resolution parameters has

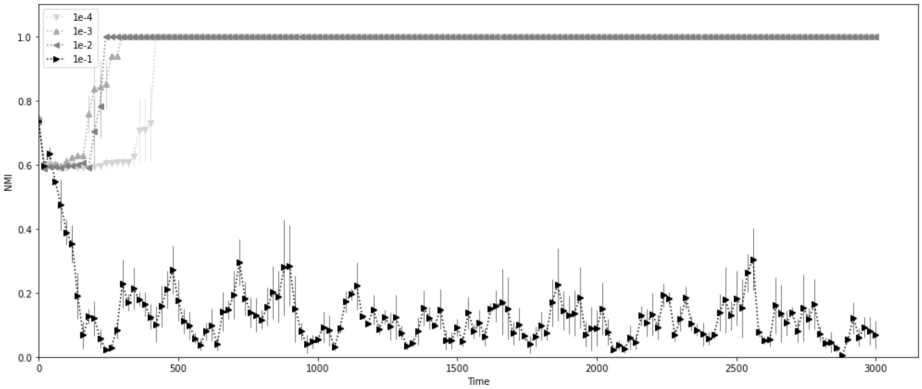
been chosen, we average performance results presented below over 10 runs.

Tie decay

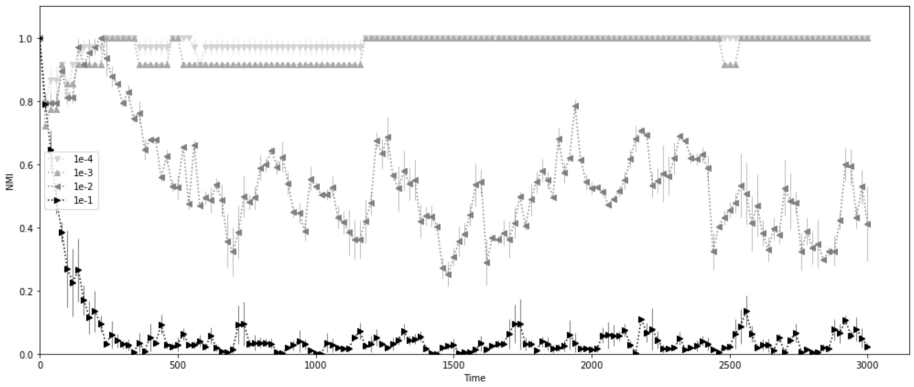
We test the influence of the tie-decay η on the speed and accuracy of recovery of the ground-truth communities. We use four tie-decay values $\eta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. For community merger in Figure 5.6a, we initially observe NMI = 1 as the algorithm correctly identifies three communities at $t = 0$. At $t > 0$, the NMI initially declines because the edges corresponding to the merged communities appear but there is not enough of them yet to overwrite the initial community configuration. We observe that the algorithm is stable (low standard deviation across runs) and recovers the split communities for the lowest tie decay values. In contrast, the recovery process is unstable and inaccurate for $\eta = 10^{-1}$. Indeed, high tie-decay processes have low characteristic times: for $\eta = 10^{-1}$, the characteristic time $\tau = \frac{1}{\eta} = 10$ so an interaction's absolute weight vanishes to $e^{-1} \approx .37$ after ten steps, and is less than .01 after 50 timesteps. The algorithm must therefore infer communities using information provided by fewer than 50 edges (recorded over the last 50 time steps). This makes the 60-node graph so sparse (less than 1 edge per node) that the algorithm cannot confidently unveil communities. The communities are recovered for both $\eta = 10^{-3}$ and $\eta = 10^{-4}$, though faster for $\eta = 10^{-3}$ because η also controls the speed of community recovery as it makes obsolete edges disappear faster. Eventually, the NMI remains constant (NMI $\approx .75$) at $t > 0$ for $\eta = 10^{-2}$ because the edges disappear fast enough for the algorithm to never realise three communities are at play. After further inspection, we notice that the algorithm believes that two communities of size 30 are present. For community split in Figure 5.6b, we observe fast community recovery for $\eta \in \{10^{-3}, 10^{-2}, 10^{-1}\}$, and the recovery is slower for lower decay $\eta = 10^{-4}$. Again, for high decay $\eta = 10^{-1}$, communities are not detectable. We also observe that temporal `Signed Louvain` detects splits faster than mergers. We leave the study of this pattern for future work. Eventually, the method is also able to detect the growth of the graph in the presence of noise, for $\eta = 10^{-3}$ and $\eta = 10^{-4}$ (Figure 5.6c).



(a) Merger



(b) Split



(c) Graph growth

Figure 5.6: **Influence of tie-decay**. The performance in terms of NMI varies with respect to the choice of the tie-decay parameter η for each of three scenarios.

Detectability thresholds

In Figure 5.3, we have shown that the parameters $\gamma_{\pm}$ control the sizes of the communities detected by `Signed Louvain`. In addition to this size resolution, `Signed Louvain` has a time resolution limit controlled by the tie-decay η. We consider the following scenario. Starting from a configuration with 3 communities **A**, **B**, **C** with

20 nodes each, a merger between **A** and **B** occurs at $t = 1$ until $t = 1250$, forming a community **D** with 40 nodes. At $t = 1250$ onwards, 10 nodes from community **D** join community **C** such that both communities **C** and **D** are equally sized (30 nodes). Figure 5.7 depicts the results. `Signed Louvain` can unveil both community events for $\eta = 10^{-3}$: the merger is detected around $t = 1000$ and the node swaps are detected around $t = 2300$. For $\eta = 10^{-4}$, only the final configuration is detected, as if the merger never happened because the duration of the merger (first 1,250 steps) is significantly lower than the characteristic time $\tau = \frac{1}{\eta} = 10^4$ at which the algorithm operates. In practice, this means that the choice of η determines the minimal time window on which a structural change must occur in order to be detected.

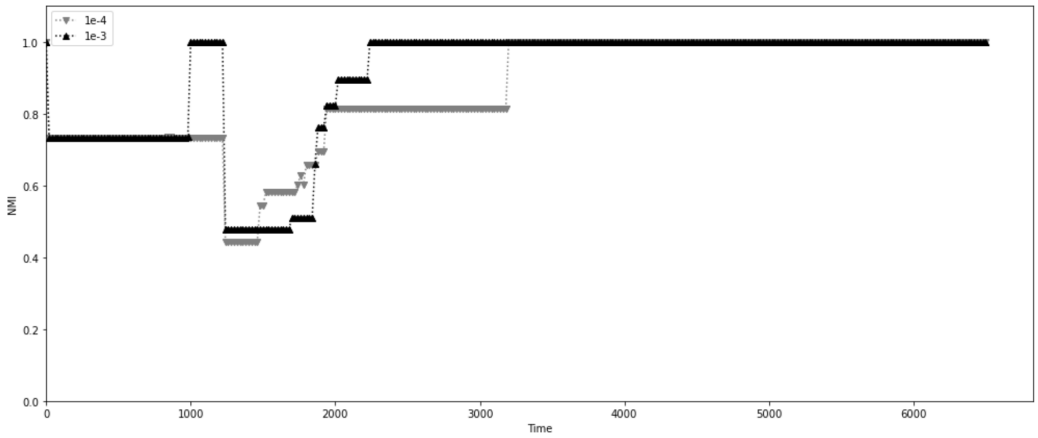


Figure 5.7: **Time resolution limit**. Temporal `Signed Louvain` is not able to detect community events which take place faster than its characteristic time $\tau = \frac{1}{\eta}$. The community merger which occurs around $t = 1000$ is undetected for $\eta = 10^{-4}$, and detected for $\eta = 10^{-3}$.

Temporal trade-off

We motivated the design of temporal `Signed Louvain` as a temporal trade-off method by the fact that trade-off methods are less prone to instability and are able to smooth community evolution (see section 5.4.1). In this paragraph, we validate this choice by comparing the approach to its instant optimal counterpart. In instant optimal approaches, we run a static community detection algorithm without initialising it with communities found at previous steps. Therefore, at each time step t, instead of initialing `Signed Louvain` with the node partition found at $t - 1$, we initialise with the default initial configuration in which communities are singletons. It means that we restart the optimisation process from scratch each time, as if the graph did not have any history. We compare how the merger event (Figure 5.5a) is detected in both

cases. To make the task more complex, 20% of edges appearing are noisy. Figure 5.8 shows that the instant optimal design is prone to instability even though it uncovers the right configuration eventually. In practice, it makes it impossible to distinguish between changes due to the evolution of the community structure and changes due to this instability. In contrast, the temporal trade-off design remains consistent (the standard deviation of the NMI is close to zero), which makes it more reliable and auditable.

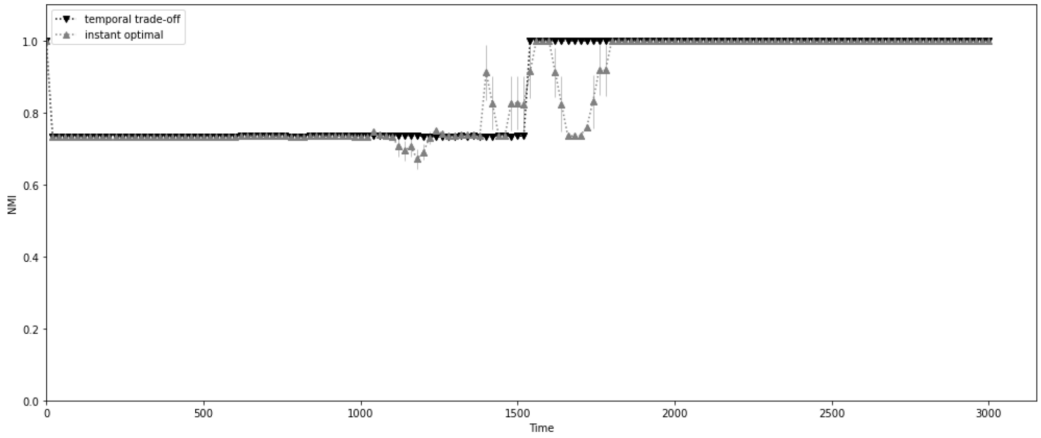


Figure 5.8: **Temporal trade-off v Instant optimal approach**. The instant optimal version of temporal `Signed Louvain` is unstable compared to its temporal trade-off version, which shows the benefits of smoothing the community evolution through carefully chosen optimisation initialisation.

5.4.5 Opinion dynamics during the 2020 BLM protests

We use temporal `Signed Louvain` on r/BLM in `DEBAGREEMENT` to demonstrate possibilities to study the opinion dynamics around the protests organised across the world over the next months following the murder of George Floyd on 25 May 2020. We focus on the period from his murder up to 15 August 2020, where the activity in r/BLM peaked (see Chapter 3, Figure 4.3). We choose an arbitrary tie decay η of 15 days given that people are more likely to discover, learn more, take and/or change their stance in periods of social turmoil following a tragedy [1, 81]. In this context, a community is a community of individuals mostly agreeing with each other, and disagreeing with the rest of the graph. We call them *opinion communities*. Figure 5.9 displays the alluvial diagram of the analysis. An alluvial diagram describes the evolution of the community structure in a temporal graph. At each time step, a dark block represents an opinion community whose size is proportional to the number of users it contains. Between two successive times t and $t+1$, the light-gray flow from

community A_t to community B_{t+1} represents the proportion of users in A_t who are in B_{t+1} in the next time. We depict the state of the communities every 15 days for readability. The initial snapshot graph contains all interactions up to 15 June 2020.

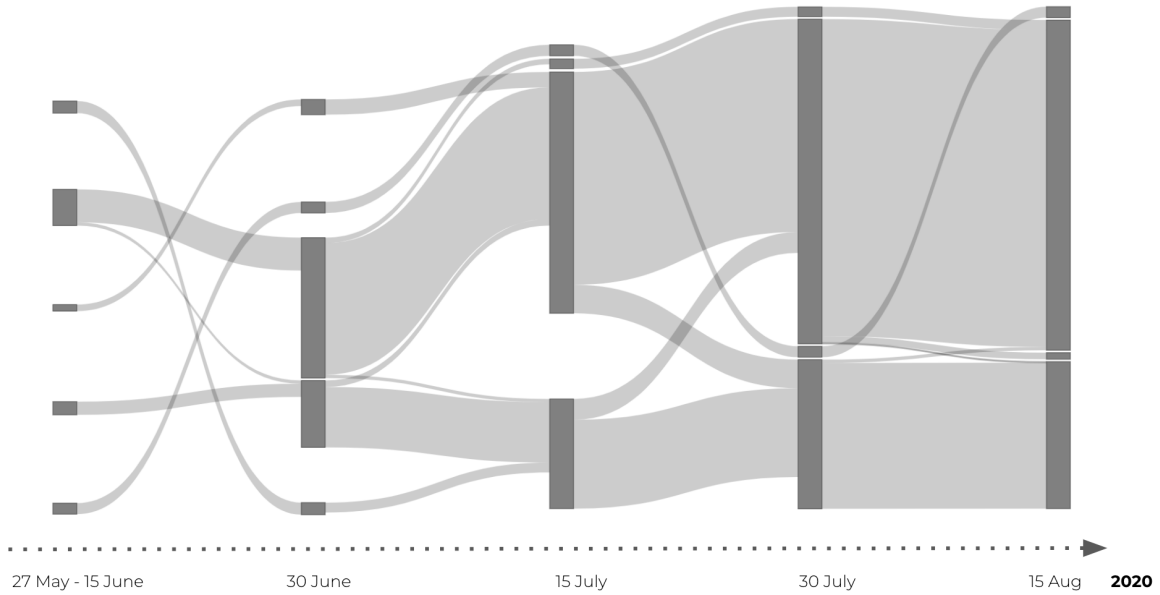


Figure 5.9: **Opinion dynamics in r/BLM**. The alluvial diagram depicts the state of opinion communities (gray blocks) at different times. The flows between communities represent the amount of nodes going from one community to the other at the next time step. A non-significant portion of nodes change of opinion (community) between 15 and 30 July 2020.

We observe two large communities and one small community until 15 August. With further inspection, we notice that the two large communities both recognise the need to eradicate systematic racism against Black communities, especially in the USA, but disagree on the way to protest to make the change happen. One believes in a drastic approach to protesting, while the other advocates for a non-violent line of action. Two topics are discussed: the perception of the police as an oppressive body for Black communities, and the removal of statues of former slave traders. We observe a small community of a few dozen of people with a balanced view[8] on these topics and seem to understand the arguments brought by the two opinion communities. In particular, we observe them oscillating between both communities from 15 to 30 July 2020. The small community represents people who do not believe racism exists—especially in Europe, and think that protests are illegitimate.

[8]We use *balanced view* not as a positive judgment on the individuals being described, but as an empirical reflection of the evenly weighted and/or oscillating views compared to the larger and opposite opinion communities.

5.5 Conclusion

In this chapter, we designed `Signed Louvain`, a generalisation of the popular Louvain method for signed graphs. Through theoretical analysis and extensive experiments, we showed that `Signed Louvain` overcomes the limitations of the Louvain method on signed graphs while maintaining a competitive computational efficiency. We also expanded the method to temporal signed graphs, validated our modeling choices, and experimentally studied the importance of the tie decay on the optimisation scheme. Eventually, we used the method to demonstrate how the model could be applied to understand opinion dynamics during the 2020 BLM protests.

Chapter 6

Learnings and future directions

In this thesis, we have studied the concept of latent associations in signed graphs. We have advanced research on signed graphs through three methods for representation learning (`SEM`), text and structure-informed predictions (`siBERT`), and community detection (`Signed Louvain`) in signed graphs, and three signed graph dataset contributions (`DEBAGREEMENT`, `TwitterSG`, and `BirdwatchSG`).

We have learnt that the contextual information surrounding an interaction is a powerful signal to exploit, whether it is an edge attribute such as the topic (Chapter 3), or the content of a discussion in a social network, or temporal signals as the local evolution of the graph (Chapter 4). `SEM` and `siBERT` are specifically designed to leverage these cues. On the one hand, we have shown that `SEM` can infer the position of nodes relative to topics they have never engaged with, proving its ability to uncover unobserved patterns in the graph. On the other hand, we have shown `siBERT`'s capability to take advantage of triadic closure to predict the sign of future edges.

We have learnt that deriving node relations iteratively, through strong structural balance assumptions, is efficient locally (paths in 2-hop graphs with `Signed Louvain`, Chapter 5) and at lengthier distances (k-hop paths in `SEM`'s random walks). The principle of two-hop moves introduced to design `Signed Louvain` can be used to generalise other heuristics, e.g. the Leiden algorithm [148], for signed graphs.

Overall, we have proposed notable contributions using simplifying but intuitive principles such as triadic closure and strong structural balance.

We believe that even further improvements can be achieved by relaxing some of our assumptions. By analysing `DEBAGREEMENT`, we have noticed that a significant amount of triads are weakly balanced or unbalanced across five real-world signed graphs (Table 4.6). We have shown in section 3.4.1 that unbalanced graphs may bring confusion to the learning process of `SEM`. This suggests that other random walk strategies should be explored to model weak structural balance instead. In

that case, a random walk terminates as soon it encounters two negative edges. This creates a different, asymmetric walk process on graph $\bar{\mathcal{G}}$ (section 3.4.3), that leads to a different matrix factorisation scheme. Our initial explorations on synthetic graphs suggest that this modeling choice would be beneficial in graphs containing at least 3 communities with unbalanced sizes. Other sampling strategies such as random walk with restart [144] could also be investigated. Still, the spectral properties of the matrices implicitly factorised by `SEM` for each of these processes - see Equation (3.21) for the strong structural balance case - can be inspected in future research. For instance, previous work has derived interesting properties for the unsigned and signed random-walk laplacians, such as their semi-definite positiveness [58, 77] and bounds for their eigenvalues [57].

Graph sparsity remains a challenge that we have not been able to overcome in this thesis: further investigation remains to improve the performance of community detection algorithms in sparse graphs (see Figure 5.1) [43]. One possibility is to generalise methods based on non-backtracking walks to the signed setting [68]. Sparsity makes it even more challenging to detect communities in temporal signed graphs, and fine-tuning $\gamma^{\pm}$ the resolution and η the tie-decay parameters is critical (section 5.4.4) [42].

As proven with `DEBAGREEMENT`, crafting representative signed graph datasets is complex (section 4.3). Despite the processing steps performed to ensure the annotations of informative interactions, only 33% of the `DEBAGREEMENT` pairs were annotated with full inter-annotator agreement. This speaks to the difficulty in annotating short exchanges containing nuanced statements, sarcasm, and diverse writing genres [109]. As observed in our experiments, the boundary between a neutral and a negative comment, or a positive one, may be blurred. As a consequence, it is particularly difficult for language models to identify neutral interactions. One potential solution is to set up annotation loops in which the language models are trained further on interactions where annotators did not all agree in order to emphasise examples that need additional attention [137].

Publications and pre-prints over the course of the DPhil

2020 Diop, B. Z., Ngom, M., Pougué-Biyong, C., & Pougué-Biyong, J. The relatively young and rural population may limit the spread and severity of COVID-19 in Africa: a modelling study. BMJ global health, 5(5), e002699. [39]

2020 John Pougué-Biyong, Bo Wang, Terry Lyons, and Alejo Nevado-Holgado. Information Extraction from Swedish Medical Prescriptions with Sig-Transformer Encoder. In Proceedings of the 3rd Clinical Natural Language Processing Workshop, pages 41–54. Association for Computational Linguistics. [118]

2021 Pougué-Biyong, John, Valentina Semenova, Renaud Lambiotte, J. Doyne Farmer, et al. "DEBAGREEMENT: A comment-reply dataset for (dis) agreement detection in online debates." Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). [117].

2023 Pougué-Biyong, J., Gupta, A., Haghighi, A., & El-Kishky, A. Learning stance embeddings from signed social graphs. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (pp. 177-185). [116]

2023 Steer, B., Arnold, N., Ba, C. T., Lambiotte, R., Jeub, L., Alnaimi, A., Pougué-Biyong, J., et al. Raphtory: The temporal graph engine for Rust and Python. arXiv preprint arXiv:2306.16309. [135]

2023 Karim Lasri, Pedro Vitor Quinta Da Castro, Mona Schirmer, Luis Eduardo San Martin, Linxi Wang, Haaya Naushan, John Pougué-Biyong, Samuel Fraiberger et al. EconBERTa: Towards Robust Extraction of Named Entities in Economics. Findings of the 2023 Conference on Empirical Methods in Natural Language Processing. [79]

Bibliography

[1] Daron Acemoglu and Asuman Ozdaglar. Opinion dynamics and learning in social networks. *Dynamic Games and Applications*, 1(1):3–49, 2011.

[2] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *nature*, 466(7307):761–764, 2010.

[3] Abeer AlDayel and Walid Magdy. Stance detection on social media: State of the art and trends. *Information Processing & Management*, 58(4):102597, 2021.

[4] Victor Amelkin, Francesco Bullo, and Ambuj K Singh. Polar opinion dynamics in social networks. *IEEE Transactions on Automatic Control*, 62(11):5650–5665, 2017.

[5] Pranay Anchuri and Malik Magdon-Ismail. Communities and balance in signed networks: A spectral approach. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ASONAM '12, page 235–242, USA, 2012. IEEE Computer Society.

[6] Samin Aref and Mark C Wilson. Balance and frustration in signed networks. *Journal of Complex Networks*, 7(2):163–189, 2019.

[7] Robert Axelrod and D Scott Bennett. A landscape theory of aggregation. *British journal of political science*, 23(2):211–233, 1993.

[8] Thomas Aynaud, Eric Fleury, Jean-Loup Guillaume, and Qinna Wang. Communities in evolving networks: definitions, detection and analysis techniques. In Niloy Ganguly, Animesh Mukherjee, Bivas Mitra, Fernando Peruani, and Monojit Choudhury, editors, *Dynamics On and Of Complex Networks*, volume 2, pages 159–200. Springer, 2013.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[10] Christopher A Bail, Lisa P Argyle, Taylor W Brown, John P Bumpus, Haohan Chen, MB Fallin Hunzaker, Jaemin Lee, Marcus Mann, Friedolin Merhout, and Alexander Volfovsky. Exposure to opposing views on social media can increase political polarization. *Proceedings of the National Academy of Sciences*, 115(37):9216–9221, 2018.

[11] Ramy Baly, Mitra Mohtarami, James Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. Integrating stance detection and fact checking in a unified corpus. *arXiv preprint arXiv:1804.08012*, 2018.

[12] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839, 2020.

[13] VI Bityutskov. Bunyakovskii inequality, 2001.

[14] Vincent Blondel, Jean-Loup Guillaume, and Renaud Lambiotte. Ffast unfolding of communities in large networks: 15 years later. *arXiv*, 2023.

[15] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[16] Javier Borge-Holthoefer, Walid Magdy, Kareem Darwish, and Ingmar Weber. Content and network dynamics behind egyptian political polarization on twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '15, page 700–711, New York, NY, USA, 2015. Association for Computing Machinery.

[17] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE transactions on knowledge and data engineering*, 20(2):172–188, 2007.

[18] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

[19] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637, 2018.

[20] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9:e15, 2020.

[21] Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. Seeing things from a different angle: Discovering diverse perspectives about claims. *arXiv preprint arXiv:1906.03538*, 2019.

[22] Y. Chen, X. Wang, B. Yuan, and Buzhou Tang. Overlapping community detection in networks with positive and negative links. *Journal of Statistical Mechanics: Theory and Experiment*, 2014, 02 2014.

[23] Yiqi Chen, Tieyun Qian, Huan Liu, and Ke Sun. " bridge" enhanced signed directed network embedding. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 773–782, 2018.

[24] Kai-Yang Chiang, Joyce Jiyoung Whang, and Inderjit S. Dhillon. Scalable clustering of signed networks using balance normalized cut. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 615–624, New York, NY, USA, 2012. Association for Computing Machinery.

[25] KR Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649, 2020.

[26] Kenneth Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[27] Alessandra Teresa Cignarella, Mirko Lai, Cristina Bosco, Viviana Patti, Rosso Paolo, et al. Sardistance@ evalita2020: Overview of the task on stance detection in italian tweets. In *EVALITA 2020 Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, pages 1–10. Ceur, 2020.

[28] Pedro Colon-Hernandez, Catherine Havasi, Jason Alonso, Matthew Huggins, and Cynthia Breazeal. Combining pre-trained language models and structured knowledge. *arXiv preprint arXiv:2101.12294*, 2021.

[29] Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. Sponge: A generalized eigenproblem for clustering signed networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1088–1098. PMLR, 2019.

[30] Mihai Cucuringu, Apoorv Vikram Singh, Déborah Sulem, and Hemant Tyagi. Regularized spectral methods for clustering signed networks. *Journal of Machine Learning Research*, 22(264):1–79, 2021.

[31] Kareem Darwish, Walid Magdy, Afshin Rahimi, Timothy Baldwin, and Norah Abokhodair. Predicting online islamophopic behavior after# parisattacks. *The Journal of Web Science*, 4, 2018.

[32] Kareem Darwish, Walid Magdy, and Tahar Zanouda. Improved stance prediction in a user similarity feature space. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 145–148, 2017.

[33] Kareem Darwish, Peter Stefanov, Michaël Aupetit, and Preslav Nakov. Unsupervised user stance detection on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 141–152, 2020.

[34] Abhimanyu Das, Sreenivas Gollapudi, and Kamesh Munagala. Modeling opinion dynamics in social networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 403–412, 2014.

[35] Delip Rao Dean Pomerleau. Fake news challenge, 2017.

[36] Tyler Derr, Charu Aggarwal, and Jiliang Tang. Signed network modeling based on structural balance theory. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 557–566, 2018.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[38] Kuntal Dey, Ritvik Shrivastava, Saroj Kaushik, and Vaibhav Mathur. Assessing the effects of social familiarity and stance similarity in interaction dynamics. In *International Conference on Complex Networks and their Applications*, pages 843–855. Springer, 2017.

[39] Binta Zahra Diop, Marieme Ngom, Clémence Pougué Biyong, and John N Pougué Biyong. The relatively young and rural population may limit the spread and severity of covid-19 in africa: a modelling study. *BMJ global health*, 5(5):e002699, 2020.

[40] Eli Eisenberg and Erez Y Levanon. Preferential attachment in the protein network evolution. *Physical review letters*, 91(13):138701, 2003.

[41] Allyson Ettinger. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48, 2020.

[42] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41, 2007.

[43] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics reports*, 659:1–44, 2016.

[44] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[45] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[46] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[47] William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2017.

[48] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[49] Frank Harary. On the notion of balance of a signed graph. *Michigan Mathematical Journal*, 2(2):143–146, 1953.

[50] Tanja Hartmann, Andrea Kappes, and Dorothea Wagner. Clustering evolving networks, 2014.

[51] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[52] Fritz Heider. Attitudes and cognitive organization. *The Journal of Psychology*, 21(1):107–112, 1946. PMID: 21010780.

[53] Kenneth Hoffman. *Linear algebra*. 1971.

[54] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[55] Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.

[56] Emoke-Agnes Horvát and Katharina A Zweig. One-mode projection of multiplex bipartite graphs. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 599–606. IEEE, 2012.

[57] Yao Ping Hou. Bounds for the least laplacian eigenvalue of a signed graph. *Acta Mathematica Sinica*, 21(4):955–960, 2005.

[58] Yaoping Hou, Jiongsheng Li, and Yongliang Pan. On the laplacian eigenvalues of signed graphs. *Linear and Multilinear Algebra*, 51(1):21–30, 2003.

[59] Zexi Huang, Arlei Silva, and Ambuj Singh. Pole: Polarized embedding for signed networks, 2021.

[60] Mohammad Raihanul Islam, B Aditya Prakash, and Naren Ramakrishnan. Signet: Scalable embeddings for signed networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 157–169. Springer, 2018.

[61] Amin Javari, Tyler Derr, Pouya Esmailian, Jiliang Tang, and Kevin Chen-Chuan Chang. Rose: Role-based signed network embedding. In *Proceedings of The Web Conference 2020*, pages 2782–2788, 2020.

[62] Jonathan Q. Jiang. Stochastic block model and exploratory analysis in signed networks. *Physical Review E*, 91(6), Jun 2015.

[63] Tali Kaufman and Izhar Oppenheim. High order random walks: Beyond spectral gap. *Combinatorica*, 40:245–281, 2020.

[64] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

[65] Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang. Side: representation learning in signed directed networks. In *Proceedings of the 2018 World Wide Web Conference*, pages 509–518, 2018.

[66] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.

[67] Jens Krause, David Lusseau, and Richard James. Animal social networks: an introduction. *Behavioral Ecology and Sociobiology*, 63:967–973, 2009.

[68] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

[69] Dilek Küçük and Fazli Can. Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1):1–37, 2020.

[70] Vivek Kulkarni, Shubhanshu Mishra, and Aria Haghighi. Lmsoc: An approach for socially sensitive pretraining. *arXiv preprint arXiv:2110.10319*, 2021.

[71] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and VS Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 333–341. ACM, 2018.

[72] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 221–230. IEEE, 2016.

[73] Jérôme Kunegis. Applications of structural balance in signed social networks. *arXiv preprint arXiv:1402.6865*, 2014.

[74] Jérôme Kunegis, Julia Preusse, and Felix Schwagereit. What is the added value of negative links in online social networks? In *Proceedings of the 22nd international conference on World Wide Web*, pages 727–736, 2013.

[75] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. Spectral analysis of signed graphs

for clustering, prediction and visualization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 559–570. SIAM, 2010.

[76] Renaud Lambiotte, Jean-Charles Delvenne, and Mauricio Barahona. Random walks, markov processes and the multiscale modular organization of complex networks. *IEEE Transactions on Network Science and Engineering*, 1(2):76–90, 2014.

[77] Renaud Lambiotte and Michael T Schaub. *Modularity and Dynamics on Complex Networks*. Cambridge University Press, 2021.

[78] Andrea Lancichinetti and Santo Fortunato. Limits of modularity maximization in community detection. *Physical review E*, 84(6):066122, 2011.

[79] Karim Lasri, Pedro Vitor Quinta de Castro, Mona Schirmer, Luis Eduardo San Martin, Linxi Wang, Tomáš Dulka, Haaya Naushan, John Pougué-Biyong, Arianna Legovini, and Samuel Fraiberger. Econberta: Towards robust extraction of named entities in economics. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11557–11577, 2023.

[80] Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Sebastian Ruder, Dani Yogatama, et al. Pitfalls of static language modelling. *arXiv preprint arXiv:2102.01951*, 2021.

[81] Francis LF Lee, Hai Liang, and Gary KY Tang. Online incivility, cyberbalkanization, and the dynamics of opinion polarization during and after a mass protest event. *International Journal of Communication*, 13:20, 2019.

[82] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650, 2010.

[83] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370, 2010.

[84] Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640, 2010.

[85] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27, 2014.

[86] Ted G Lewis. *Network science: Theory and applications*. John Wiley & Sons, 2011.

[87] Yang Li, Bo Yang, Xuehua Zhao, Zhejian Yang, and Hechang Chen. Ssbm: A signed stochastic block model for multiple structure discovery in large-scale exploratory signed networks. *Knowledge-Based Systems*, 259:110068, 2023.

[88] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert systems with applications*, 41(4):2065–2073, 2014.

[89] Xueyan Liu, Wenzhuo Song, Katarzyna Musial, Xuehua Zhao, Wanli Zuo, and Bo Yang. Semi-supervised stochastic blockmodel for structure analysis of signed networks. *Knowledge-Based Systems*, 195:105714, 2020.

[90] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[91] Naoki Masuda and Renaud Lambiotte. *A Guide to Temporal Networks*. Complexity Science. World Scientific publishing, September 2016.

[92] Naoki Masuda, Mason A Porter, and Renaud Lambiotte. Random walks and diffusion on networks. *Physics reports*, 716:1–58, 2017.

[93] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[94] Jane Memmott, Nickolas M Waser, and Mary V Price. Tolerance of pollination networks to species extinctions. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 271(1557):2605–2611, 2004.

[95] Pedro Mercado, Francesco Tudisco, and Matthias Hein. Clustering signed networks with the geometric mean of laplacians, 2017.

[96] Pedro Mercado, Francesco Tudisco, and Matthias Hein. Spectral clustering of signed graphs via matrix power means, 2019.

[97] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[98] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[99] Corrado Monti, Giuseppe Manco, Cigdem Aslay, and Francesco Bonchi. Learning ideological embeddings from information cascades. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1325–1334, 2021.

[100] Sarang Narkhede. Understanding auc-roc curve. *Towards Data Science*, 26(1):220–227, 2018.

[101] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[102] Mark Newman. *Networks*. Oxford university press, 2018.

[103] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[104] Mark EJ Newman. The structure and function of networks. *Computer Physics Communications*, 147(1-2):40–45, 2002.

[105] Jae Dong Noh and Heiko Rieger. Random walks on complex networks. *Physical review letters*, 92(11):118701, 2004.

[106] Rogier Noldus and Piet Van Mieghem. Assortativity in complex networks. *Journal of Complex Networks*, 3(4):507–542, 2015.

[107] OpenAI. Gpt-4 technical report, 2023.

[108] Larry Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. Pagerank: Bringing order to the web. Technical report, Stanford Digital Libraries Working Paper, 1997.

[109] Desmond U Patton, William R Frey, Kyle A McGregor, Fei-Tzin Lee, Kathleen McKeown, and Emanuel Moss. Contextual analysis of social media: The

promise and challenge of eliciting context in social media posts with natural language processing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 337–342, 2020.

[110] Leto Peel and Aaron Clauset. Detecting change points in the large-scale structure of evolving networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[111] Leto Peel, Jean-Charles Delvenne, and Renaud Lambiotte. Multiscale mixing patterns in networks. *Proceedings of the National Academy of Sciences*, 115(16):4057–4062, 2018.

[112] Leto Peel, Daniel B Larremore, and Aaron Clauset. The ground truth about metadata and community detection in networks. *Science advances*, 3(5):e1602548, 2017.

[113] Matteo Pellegrini, David Haynor, and Jason M Johnson. Protein interaction networks. *Expert review of proteomics*, 1(2):239–249, 2004.

[114] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[115] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[116] John Pougué-Biyong, Akshay Gupta, Aria Haghighi, and Ahmed El-Kishky. Learning stance embeddings from signed social graphs. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 177–185, 2023.

[117] John Pougué-Biyong, Valentina Semenova, Alexandre Matton, Rachel Han, Aerin Kim, Renaud Lambiotte, and Doyne Farmer. Debagreement: A comment-reply dataset for (dis) agreement detection in online debates. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[118] John Pougué-Biyong, Bo Wang, Terry Lyons, and Alejo Nevado-Holgado. Information extraction from Swedish medical prescriptions with sig-transformer encoder. In Anna Rumshisky, Kirk Roberts, Steven Bethard, and Tristan Naumann, editors, *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 41–54, Online, November 2020. Association for Computational Linguistics.

[119] Nicolas Pröllochs. Community-based fact-checking on twitter's birdwatch platform. *arXiv preprint arXiv:2104.07175*, 2021.

[120] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 459–467, 2018.

[121] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26, 2020.

[122] Inzamam Rahaman and Patrick Hosein. A method for learning representations of signed networks. In *Proceedings of the 14th International Workshop on Mining and Learning with Graphs (MLG)*, 2018.

[123] Karthik Raman. Construction and analysis of protein–protein interaction networks. *Automated experimentation*, 2:1–11, 2010.

[124] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical review E*, 74(1):016110, 2006.

[125] Manoel Horta Ribeiro, Pedro H Calais, Virgílio AF Almeida, and Wagner Meira Jr. " everything i disagree with is# fakenews": Correlating political polarization and spread of misinformation. *arXiv preprint arXiv:1706.05924*, 2017.

[126] Sara Rosenthal and Kathleen McKeown. I couldn't agree more: The role of conversational structure in agreement and disagreement detection in online discussions. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 168–177, 2015.

[127] Giulio Rossetti and Rémy Cazabet. Community discovery in dynamic networks: A survey. *ACM Comput. Surv.*, 51(2), February 2018.

[128] Martin Rosvall and Carl T Bergstrom. Mapping change in large networks. *PloS one*, 5(1):e8694, 2010.

[129] Majid Saberi, Reza Khosrowabadi, Ali Khatibi, Bratislav Misic, and Gholamreza Jafari. Topological impact of negative links on the stability of resting-state brain network. *Scientific reports*, 11(1):2176, 2021.

[130] Younes Samih and Kareem Darwish. A few topical tweets are enough for effective user stance detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2637–2646, 2021.

[131] Michael T Schaub, Jean-Charles Delvenne, Martin Rosvall, and Renaud Lambiotte. The many facets of community detection in complex networks. *Applied network science*, 2(1):1–13, 2017.

[132] Xiao Shen and Fu-Lai Chung. Deep network embedding for graph representation learning in signed networks. *IEEE Transactions on Cybernetics*, 50:1556–1568, 04 2020.

[133] Georg Simmel. *Georg Simmel on individuality and social forms*. University of Chicago Press, 2011.

[134] Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100, 1997.

[135] Ben Steer, Naomi Arnold, Cheick Tidiane Ba, Renaud Lambiotte, Haaroon Yousaf, Lucas Jeub, Fabian Murariu, Shivam Kapoor, Pedro Rico, Rachel Chan, et al. Raphtory: The temporal graph engine for rust and python. *arXiv preprint arXiv:2306.16309*, 2023.

[136] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[137] Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*, 2022.

[138] Kashin Sugishita, Mason A Porter, Mariano Beguerisse-Díaz, and Naoki Masuda. Opinion dynamics on tie-decay networks. *Physical Review Research*, 3(2):023249, 2021.

[139] Panagiotis Symeonidis, Nantia Iakovidou, Nikolaos Mantas, and Yannis Manolopoulos. From biological to social networks: Link prediction based on multi-way spectral clustering. *Data & Knowledge Engineering*, 87:226–242, 2013.

[140] Michael Szell, Renaud Lambiotte, and Stefan Thurner. Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences*, 107(31):13636–13641, 2010.

[141] Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th international conference on world wide web*, pages 613–624, 2016.

[142] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[143] Yu Tian and Renaud Lambiotte. Spreading and structural balance on signed networks. *arXiv preprint arXiv:2212.10158*, 2022.

[144] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, pages 613–622. IEEE, 2006.

[145] V. A. Traag and Jeroen Bruggeman. Community detection in networks with positive and negative links. *Phys. Rev. E*, 80:036115, Sep 2009.

[146] Vincent Traag, Patrick Doreian, and Andrej Mrvar. Partitioning signed networks. *Advances in network clustering and blockmodeling*, pages 225–249, 2019.

[147] Vincent A Traag and Jeroen Bruggeman. Community detection in networks with positive and negative links. *Physical Review E*, 80(3):036115, 2009.

[148] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

[149] Joshua A Tucker, Andrew Guess, Pablo Barberá, Cristian Vaccari, Alexandra Siegel, Sergey Sanovich, Denis Stukal, and Brendan Nyhan. Social media, political polarization, and political disinformation: A review of the scientific literature. *Political polarization, and political disinformation: a review of the scientific literature (March 19, 2018)*, 2018.

[150] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[151] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[152] Albertha JM Walhout. Unraveling transcription regulatory networks by protein–dna and protein–protein interaction mapping. *Genome research*, 16(12):1445–1454, 2006.

[153] Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596, Montréal, Canada, June 2012. Association for Computational Linguistics.

[154] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. Signed network embedding in social media. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 327–335. SIAM, 2017.

[155] Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2:297–310, 2014.

[156] Magdalena Wojcieszak. Deliberation and attitude polarization. *Journal of Communication*, 61(4):596–617, 2011.

[157] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. Graph neural networks for natural language processing: A survey. *arXiv preprint arXiv:2106.06090*, 2021.

[158] B. Yang, Xueyan Liu, Y. Li, and Xuehua Zhao. Stochastic blockmodeling and variational bayes learning for signed network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29:2026–2039, 2017.

[159] Bo Yang, Kwok-Wai Cheung, and Jiming Liu. Community mining from signed social networks. ieee trans knowl data eng. *Knowledge and Data Engineering, IEEE Transactions on*, 19:1333–1348, 11 2007.

[160] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[161] Taha Yasseri and Filippo Menczer. Can the wikipedia moderation model rescue the social marketplace of ideas? *arXiv preprint arXiv:2104.13754*, 2021.

[162] Shuhan Yuan, Xintao Wu, and Yang Xiang. Sne: signed network embedding. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 183–195. Springer, 2017.

[163] Xuehua Zhao, Xueyan Liu, and Huiling Chen. Network modelling and variational bayesian inference for structure analysis of signed networks. *Applied Mathematical Modelling*, 61, 04 2018.

[164] Quan Zheng and David B Skillicorn. Spectral embedding of signed networks. In *Proceedings of the 2015 SIAM international conference on data mining*, pages 55–63. SIAM, 2015.

[165] Quan Zheng and David B Skillicorn. Modeling signed social networks using spectral embedding. *Social Network Analysis and Mining*, 11(1):1–18, 2021.

[166] Jianlin Zhou, Lingbo Li, An Zeng, Ying Fan, and Zengru Di. Random walk on signed networks. *Physica A: Statistical Mechanics and its Applications*, 508, 05 2018.

[167] Katharina Anna Zweig and Michael Kaufmann. A systematic approach to the one-mode projection of bipartite graphs. *Social Network Analysis and Mining*, 1:187–218, 2011.