Survey of Generative Methods for Social Media Analysis*

Stan Matwin† Aristides Milios‡ Paweł Prałat§ Amilcar Soares¶

François Théberge‖

November 3, 2022

*We acknowledge the support of the Communications Security Establishment and Defence Research and Development Canada. The scientific or technical validity of this report is entirely the responsibility of the authors and the contents do not necessarily have the approval or endorsement of the Government of Canada.

†Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada and Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland; e-mail: stan@cs.dal.ca

‡Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada; e-mail: amilios@dal.ca

§Department of Mathematics, Toronto Metropolitan University, Toronto, ON, Canada; e-mail: pralat@ryerson.ca

¶Department of Computer Science, Memorial University of Newfoundland, St. John's, NL, Canada; e-mail: amilcarsj@mun.ca

‖Tutte Institute for Mathematics and Computing, Ottawa, ON, Canada; email: theberge@ieee.org

Contents

1 Introduction

This survey draws a broad-stroke, panoramic picture of the State of the Art (SoTA) of the research in generative methods for the analysis of social media data. It fills a void, as the existing survey articles are either much narrower in their scope [7] or are dated [19, 221, 254]. We included two important aspects that currently gain importance in mining and modelling social media: dynamics and networks. Social dynamics are important for understanding the spreading of influence or diseases, formation of friendships, the productivity of teams, etc. Networks, on the other hand, may capture various complex relationships providing an additional insight and identifying important patterns that would otherwise go unnoticed.

The article is divided in five chapters and provides an extensive bibliography consisting of more than 250 papers. Open problems, highlighting potential future directions, are clearly identified. We chose sentiment analysis as an application providing common thread between the four parts of the survey.

We start with Chapter 2 devoted to the discussion of data models and ontologies for social network analysis. We organized the data models based on the concepts they use to solve a social media research problem such as homophily, social identity linkage, and personality analysis. We also discuss some ontologies for sentiment analysis and situational awareness. We conclude this chapter with highlighting promising research directions such as working with metadata and federated learning.

Chapter 3 is devoted to text generation and generative text models and the dangers they pose to social media and society at large. The current SoTA in text generation, i.e. large pre-trained autoregressive Transformer models, the prime example of which is GPT3, is highlighted. These models are trained on massive amounts of data (e.g. Common Crawl), and have hundreds of billions of parameters. This allows them to generate eerily coherent text that is near-indistinguishable from text written by humans. The potential of these models for nefarious use is outlined, and potential ways to mitigate these harms via "fake news" detection through contextual information are provided as well.

Chapter 4 is devoted to topic modelling and sentiment analysis in context of social networks. Traditional topic modelling approaches are briefly described. Following this, methods to fuse deep learning and these traditional approaches for topic modelling are outlined in detail. The unique challenges that social media content poses for both topic modelling and sentiment analysis, as well as approaches that seek to mitigate them are discussed. In terms of sentiment analysis, both unsupervised rule-based approaches and transfer-learning-based approaches using large Transformer models (the current SoTA for complex sentiment analysis) are discussed. Finally, some interesting developing fields within sentiment analysis are outlined, with regards to both multimodal and target-based sentiment analysis.

Chapter 5 is devoted to graph theory tools and approaches to mine and model social networks. Such tools become increasingly important in machine learning and data science. There are many important aspects so we tried to narrow the topics down to a few most important ones. We concentrate on graph embeddings and their evaluation, higher order structures, dynamics, and synthetic models.

2 Ontologies and Data Models for Cross-platform Social Media Data

The creation of social media platforms generated an immense volume and diversity of content produced and exchanged by users. According to [126], Facebook, Twitter, and Instagram boast over two billion monthly active users, and as such, their ability to directly and indirectly connect the world's population has never been easier or more far reaching. Also, according to a Pew Research study [44], 56% of US adults online use more than one social media platform. The plethora of heterogeneous online platforms fostered a vast scientific production on various applications of social media analysis, ranging from sentiment analysis to cyber influence campaigns. Integrating data from different social media platforms is challenging because many of them were created for multiple purposes. For example, while Linkedin is mainly focused on professional networking and career development, Twitter is used in diverse ways by different groups of users as stated in [114]. The way users interact and produce content in such platforms are also heterogeneous and include likes, dislikes, shared videos or images, voting, friendships or connections, posts, and private messages. In this chapter, we discuss data models to organize the social media data by topics of common users' interests (Section 2.1) or the use of ontologies to organize data from heterogeneous sources (Section 2.2). In all subsections, we detail and discuss the most relevant works (i.e., with a greater number of citations over the years) in the aforementioned topics.

2.1 Data Models for Social Media Data Analysis

In this section we discuss approaches for merging social media data with external sources. For example, several approaches propose to enhance tweets or other social media data sources by annotating them with unambiguous semantic concepts defined in external knowledge bases such as Wikipedia or DBpedia. These knowledge bases provide an explicit semantic representation of concepts and their relations. They, therefore, provide additional contextual information about tweets and their underlying semantics, allowing the creation of group of users with similar topics or interests.

The annotations used by most of the works that use twitter data are either provided by its API[1] or are done with online and crowdsourced tools[2][3][4], or tools such as GATE [37] or Webanno [251]. Most tools work by allowing users to select a word or a sentence and tag it with a given value. A good and recent reading on the annotation topic can be found in [60]. The reliability on the data mainly depends on the expertise of the annotators used in the tagging process and in the number of tags provided to the same instance (i.e., tweet).

We divided the data models based on the concepts they use to solve a social media

[1]https://developer.twitter.com/en/docs/twitter-api/annotations/overview
[2]https://www.lighttag.io/
[3]https://www.tagtog.net/
[4]https://github.com/doccano/doccano

research problem. We first discuss works that use the concept of homophily, and then move on to the social media linkage problem. Finally, we show some works that uses images to infer personality traits of users.

Homophily Analysis

Homophily is the tendency of individuals to befriend other individuals sharing the same interest in Twitter communities [73]. Modeling the perception of friendship to perform homophily analysis may be challenging. A dataset enriched with the user's activity or interests is necessary to measure homophily since the social graph itself does not contain such information. Generally, the works in this area would use text from the messages (e.g., topic modeling) or some meta information of the social (e.g., the similarity of the time-zone, popularity, user's subgraph in the vicinity, etc). The notion of homophily has also commonly been modeled in social networks by mutual-follow, and mutual-mention relations [18].

Paper [73] proposes Twixonomy, which is a novel method for analyzing homophily in large social networks based on a hierarchical representation of users' interest. The outcome of Twixonomy is a Directed Acyclic Graph (DAG) taxonomy where leaf nodes are pages from Wikipedia associated to Twitter users per topic, and the remaining nodes are Wikipedia categories. The authors associate Wikipedia pages with topical users in users' friendship lists to obtain a hierarchical representation of interests. Many pages can be associated with a user name, and to handle such a problem, they use a word sense disambiguation algorithm. Users can then be directly or indirectly linked to one or more Wikipedia pages representing their interests. The algorithm starts from a set of wikipages representing users' interests, and they consider Wikipedia categories as a sub-graph induced from these pages. Cycles are removed to obtain a DAG, and efficient cycle pruning is also performed using an iterative algorithm. The advantages of the Twixonomy include a compact, tunable and readable way to express the users' interest and it uses only interests explicitly expressed by the users. Figure 1 shows the Twixonomy of a "common" user with 7 topical friends in his/her friendship list. Wikipages are the leaf nodes of the Twixonomy in Figure 1, and the other nodes are Wikipedia categories layered by generality level. The mid-low categories are the most representative of a user's interests since, as the distance between a Wikipage and a hypernym node increases, the semantic relatedness decreases [73]. In the example, the categories Economics, Basketball, and Mass Media could be chosen to summarize all the user's primitive interests [73]. The experiments performed in [73] shows that while homophily is indeed a significant phenomenon in Twitter communities it is not pervasive. The authors conclude that inferring user's preferences on the basis of those of their friends is not a fully reliable strategy. In a second experiment, the authors show that homophily also depends to some extent on the interests that identifies a community [73]. The results show that people interested in education and fashion are more homophilous and, at the same time, those supporting political leaders and women's organizations have a minor tendency to befriend other users with the same interests.

Twixonomy is used in [74] for examining the distribution of interests in Twitter ac-

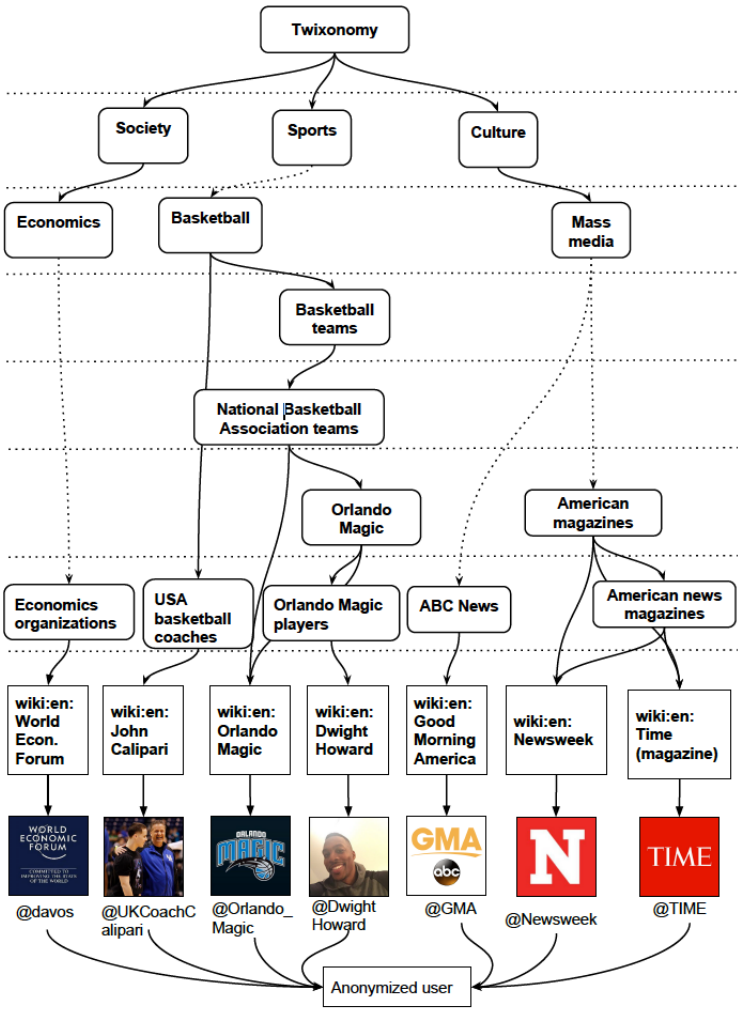

Figure 1: Twixonomy example. Source: [73].

cording to gender. Paper [74] uses a large list of female and male names extracted from several sources to classify the gender, and they also analyzed two populations: common users and topical users. The results showed that the proportion of celebrities and peers' interests in the topmost categories is not statistically significant than the respective ratio in whole populations, except for the category Sports, where males dominate [74]. The experiments also found very few women leaders, but women are indeed interested in leadership, but it seems that they prefer to follow male leaders. Also, men have a significantly higher tendency towards homophily than women. The experiments also point out that except for the categories Writers, Democrats, and Women's organizations, women are either non-homophylous or support man or non-gendered entities significantly more than other women [74].

Social Identity Linkage

Social identity linkage is the problem of linking users across different social media platforms. A survey on this topic can be found in [214, 246]. The objective is to obtain from social media data a deeper understanding and more accurate profiling of users. There are several applications to be built from linking user identities, such as enhancing friend recommendations, information diffusion, and analyzing network dynamics.

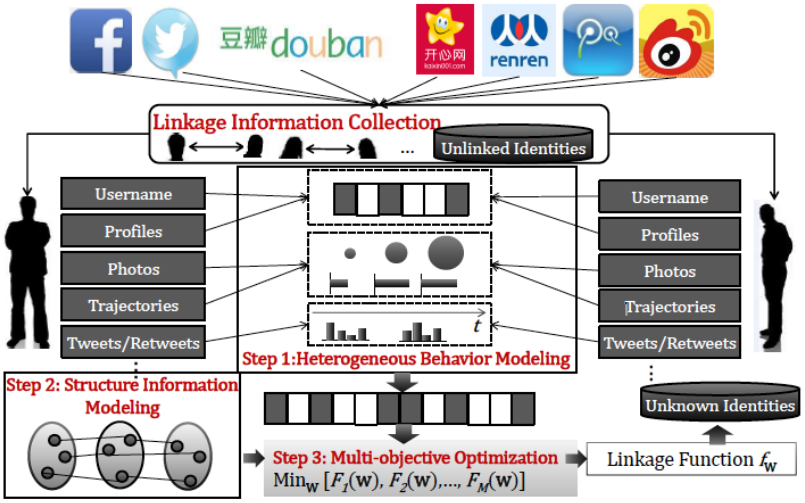

Figure 2: Hydra framework. Source: [162].

Paper [162] proposes a framework for cross-platform user identity linkage via heterogeneous behavior modeling named HYDRA. HYDRA is divided into three steps and can be seen in Figure 2. First, in the behavior similarity modeling, the relationship between two users of a pair for all user pairs via heterogeneous behavior modeling is calculated. In the second step the framework builds a structure consistency graph on user pairs by considering both the core network structure of the users and their behavior similarities. Finally, a multi-objective optimization is done based on the previous two steps, which jointly optimizes the prediction accuracy on the labeled user pairs and multiple structure consistency measurements across different platforms. HYDRA uses common textual attributes present in user profiles such as name, gender, age, nationality, profession, education and email account and visual attributes such as face images used in the profile. The authors evaluated HYDRA against the state-of-the-art solutions on two real data sets — five popular Chinese social networks and two popular English social networks. In summary, they evaluated a total of 10 million users and more than 10 terabytes of data and results demonstrated that HYDRA outperformed other baselines in identifying true user linkage across different platforms.

Paper [262] proposes a deep reinforcement learning comprehensive framework to address the heterogeneity called DeepLink to study the social identity linkage problem. DeepLink is an end-to-end network alignment approach and a semi-supervised user iden-

tity linkage learning algorithm that does not require a heavy feature engineering and can easily incorporate features created from the users' profiles. DeepLink takes advantage of deep neural networks to learn latent semantics of both user activities and network structure in an end-to-end manner. It also leverages a semi-supervised graph regularization to predict the context (neighboring structures) of nodes in the network. The experiments conducted demonstrated that the proposed framework outperforms various user identity linkage methods in linkage precision and ranking matching user identity.

Personality Analysis

Generally, sentiment analysis variables takes values such as positive, negative, or neutral. These variables can also have a more extensive range of values, allowing for multiple assignments of sentiment in a single word. Additional meta-features based on the sentiment values can also be generated, such as subjectivity and polarity. Subjectivity is the ratio of positive and negative sentences to neutral sentences, while polarity is the ratio of positive to negative sentences. This is a very active research area and recent survey with past and recent works in this topic, mainly with Twitter data, can be found in [9].

Several works have been done to predict personality traits from twitter data. For example, in [92] the authors propose a method by which a user's personality can be accurately predicted through the publicly available information on their Twitter profile (e.g., number of followers, followed, mentions, hashtags, replies, density of the social network, etc). Three tools were used for feature generation [92] with the objective of analyzing the content of users' tweets: the Linguistic Inquiry and Word Count (LIWC) tool [183] (e.g., 81 features in five categories), MRC Psycholinguistic Database (e.g., over 150,000 words with linguistic and psycholinguistic features of each word) and the General Inquirer dataset[5] (provides a hand annotated dictionary that assigns words sentiment values on a -1 to +1 scale). The work [190] in shows a study on the relationship between personality traits and five types of Twitter users: listeners (those who follow many users), popular (those who are followed by many), highly-read (those who are often listed in others' reading lists), and two types of influential. The work also tries to the predict a user's personality traits out of three variables that are publicly available on any Twitter profile: the number of profiles the user follows, number of followers, and number of times the user has been listed in others' reading lists. The results presented in [190] show that all user types (listeners, popular, highly-read, and influential users) are emotionally stable (low in Neuroticism), and most of them are extrovert. Results also show that user personality can be easily and effectively predicted from public data, and openness is the easiest trait to predict, while extraversion is the most difficult.

The rest of this section will focus on image-based personality analysis. Recent research shows that personality traits can be inferred based on image-based content analysis and a survey can be found in [28]. Pictures include many features such as objects, colors, faces that can be automatically extracted using modern computer vision algorithms. These features can be used to examine the relationships between users' personalities and image

[5]http://www.wjh.harvard.edu/ inquirer/

posting across different social media platforms. For example, images can be used for detecting users' anxiety and depression as shown in [100]. The authors explore how depression and anxiety traits can be automatically inferred by looking at images that users post and set as profile pictures. They compare different visual feature sets extracted from posted images and profile pictures. The analysis of image features associated with mental illness essentially confirm previous findings of the indications regarding depression and anxiety. Facial expressions of depressed users show fewer signs of positive moods, such as less joy and smiling, and appear more neutral and less expressive. Interestingly, depressed individuals' profile pictures are marked by the fact that they are more likely to contain a single face (i.e., user's face) rather than show the user surrounded by friends.

The work shown in [201] tries to quantify image sharing preferences and to build models that automatically predict users' personality in a cross-modal and cross-platform setting using Twitter and Flickr. Figure 3 show the process of the cross-modal and cross-platform analysis. First, the authors assemble a dataset containing user posts, profile images, liked images and texts. After, they extract features from the images and analyze the text to predict personality traits (e.g., openness, conscientiousness, extraversion, agreeableness, and neuroticism).

The results presented in [201] show that multiple interactions that users have with social media platforms (i.e., choosing profile pictures, posting, and liking images) have predictive utility for automatic personality assessment of users. Predictive results are also boosted when information from multiple social networks are combined. Results show also that users' posted images had the best performance in predicting personality, followed by liked images and, finally, profile pictures. Liked images are more diverse in their content, and as result, algorithms would need a more extensive set of such pictures across the user's timeline to make more accurate predictions.

2.2 Ontologies for Social Media Data

A popular and reasonable choice to integrate heterogeneous sources such as social media data is by defining an ontology. An ontology represents the domain knowledge as a hierarchy of concepts [97] and includes machine-interpretable definitions of the domain's basic terms, and relations [98]. Ontologies also define a common vocabulary for researchers who needs to share information in a domain. Defining an ontology for a given problem or domain helps share a general understanding of knowledge among different teams and makes the domain knowledge reusable. The next sections describe two main tasks related with social media data and ontologies which are sentiment analysis and situational awareness.

Ontologies for Sentiment Analysis

Sentiment analysis for subject information extraction from the text data has become more dependent on natural language processing methods, especially for business and healthcare, since online products and service reviews may affect consuming behaviors. A survey in multimodal sentiment analysis can be found here [218]. Sentiment analysis algorithms typically apply natural language processing techniques with additional resources (e.g.,

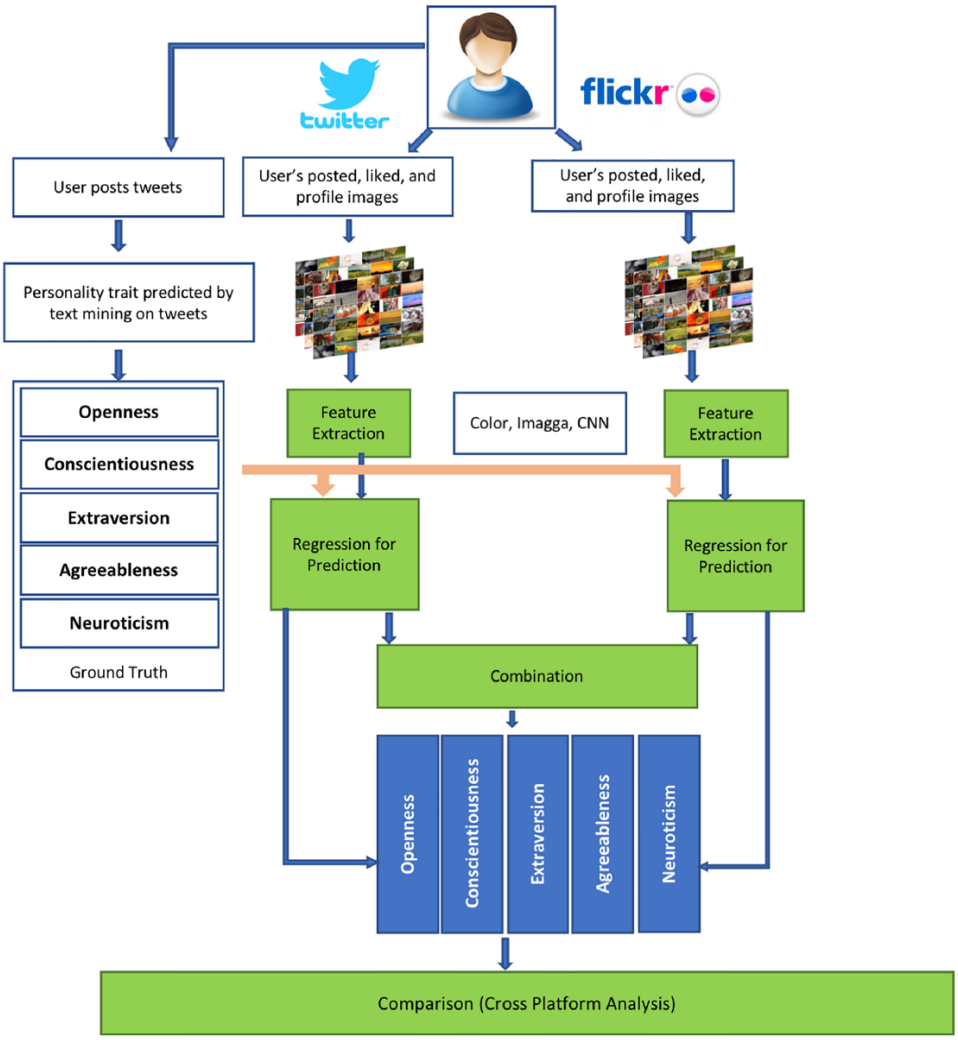

Figure 3: Overview of cross-platform analysis for user personality prediction. Source: [201].

sentiment and emotion based lexicons, sophisticated dictionaries, and ontologies) to model the documents. The Plutchik's model [186] is a common choice of several authors to assign labels that may reflect how users feel about topics, images, and situations on social media.

The Plutchik's model uses a circle of emotions depicted as a colour wheel. Like colors, primary emotions can be expressed at different degrees, and for each emotion, there are three degrees. For example, acceptance is a less intense degree of trust, and admiration is a higher degree of trust. Plutchik's emotions can be mixed and form a new emotion. For example, the combination of joy and anticipation results in optimism. In summary, the Plutchik wheel of emotions [186] are organized by eight basic emotions (Figure 4, each with three valences: (1) ecstasy > joy > serenity; (2) admiration > trust > acceptance; (3) terror > fear > apprehension; (4) amazement > surprise > distraction; (5) grief > sadness > pensiveness; (6) loathing > disgust > boredom; (7) rage > anger > annoyance;

and (8) vigilance > anticipation > interest.

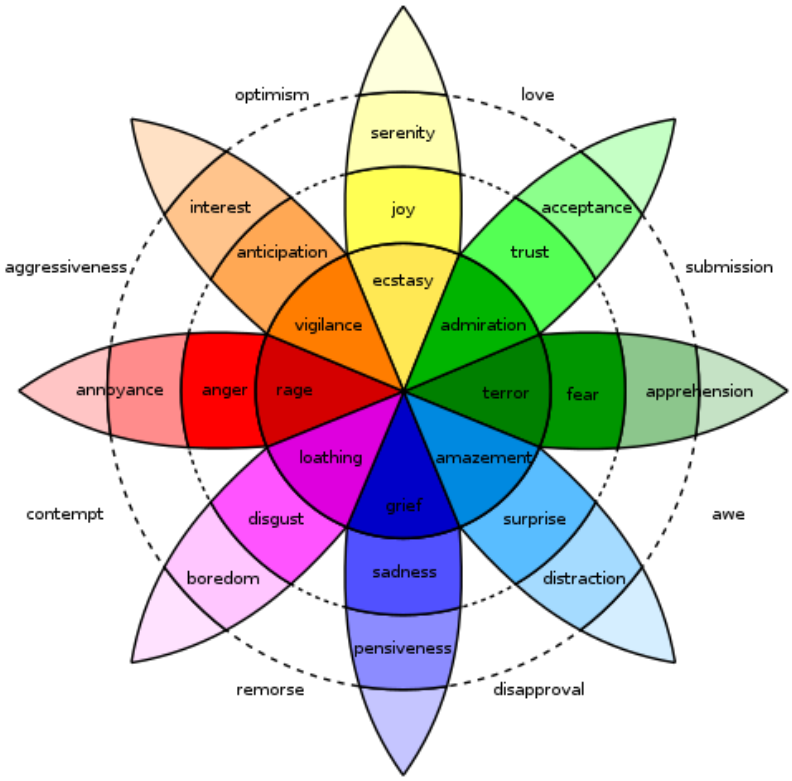

Figure 4: Overview of cross-platform analysis for user personality prediction. Source: https://commons.wikimedia.org/wiki/File:Plutchik-wheel.svg#metadata

Paper [38] applies the Plutchik's wheel of emotions as the guiding principle to construct a large-scale visual sentiment ontology (VSO) that consists of more than 3,000 adjective noun pairs. VSO ensures that each selected concept respects a strong sentiment, has a link to emotions, is frequently used in practice, and has a reasonable detection accuracy. This paper also proposes SentiBank [38], a novel visual concept detector library that can detect the presence of 1,200 adjective noun pairs in an image. The experiments on detecting the sentiment of image tweets exhibit notable improvement in detection accuracy when comparing the proposed SentiBank based predictors with text-based approaches. An overview of the work done in [38] can be seen in Figure 5. During the first step, they use the 24 emotions defined in Plutchik's theory to derive search keywords and retrieve images and videos from Flickr and YouTube. The tags linked with the retrieved images and videos are extracted, and sentiment values, adjectives, verbs, and nouns are assigned to such tags. Adjectives with strong sentiment values and nouns are then used to form adjective noun combinations. Those adjective noun pairs are then ranked by their frequency on Flickr and sampled to create an assorted and extensive ontology containing more than 3,000 adjective noun pairs. After, they train individual detectors using Flickr

images tagged with adjective noun pairs, keeping only detectors with good performance to build SentiBank. Sentibank consists of 1,200 adjective noun pairs concept detectors providing a 1,200 dimension adjective noun pairs detector response for a given image.

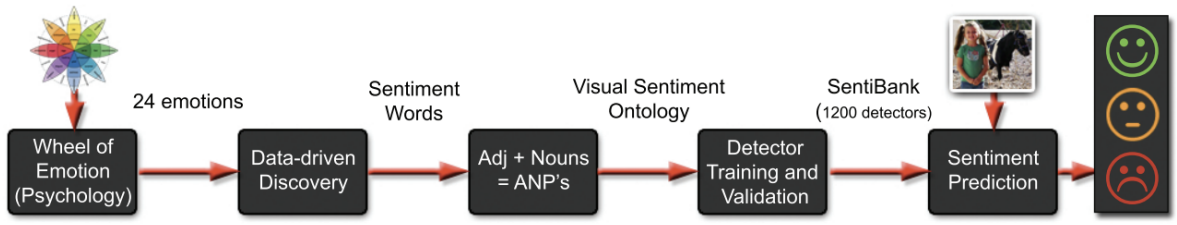

Figure 5: Overview of how VSO and SentiBank were assembled. Source: [38]

The work shown in [50] proposes DeepSentiBank, which is a fine-tuned Convolutional Neural Network (CNN) that is based on the VSO [38]. The visual sentiment concepts are adjective noun pairs automatically discovered from the tags of web photos, and are utilized as statistical hints for detecting emotions depicted in the images from Flickr. The data used by DeepSentibank provided both the pictures and the tags. We were not able to locate the data with the Flickr images used so we are not sure if user images were used in the work. Regarding the measures of accuracy describing how the tool in performing visual sentiment analysis on social media images, the only information provided was that they used a simple procedure to train with 826,806 instances and test with 2,089 ANPs and use top-k accuracy - the percentage of images that have the pseudo ground truth label in top k detected concepts. The performance evaluation shows that DeepSentiBank performance significantly improved the annotation accuracy and retrieval performance when compared to some baselines.

In [127], the authors explore uniqueness of culture and language in relation to human affect such as sentiment and emotion semantics, and how they manifest in social multimedia. The authors present a large-scale multilingual visual sentiment ontology (MVSO) and a dataset including adjective-noun pairs from 12 languages of diverse origins: Arabic, Chinese, Dutch, English, French, German, Italian, Persian, Polish, Russian, Spanish, and Turkish. MVSO is organized hierarchically into noun-based clusters and sentiment-biased adjective-noun pair sub-clusters, a multilingual, sentiment-driven visual concept detector bank. An overview of MVSO can be seen in Figure 6. The MVSO building process begins with crawling images and metadata based on emotion keywords. Image tags are labeled with part-of-speech tags, and adjectives and nouns are used to form candidate adjective-noun pair combinations. In the last step, the candidate adjective-noun pairs are filtered based on several criteria. This last step helps to remove incorrect pairs and will form the MVSO with diversity and coverage. The experiments with a cross-lingual analysis of MVSO and image dataset (data extracted from Flickr) using using semantic matching and visual sentiment prediction provide evidence that emotions are not necessarily culturally universal [127]. The experiments show that there are commonalities and distinct separations in how visual affect is expressed and perceived, where other works assumed

only commonalities.

Paper [194] study how emotional and informative message appeal in visual and textual modalities influences customer engagement in terms of likes and comments. The authors use the trained MVSO detectors and apply the model to extract the top five adjective-noun pairs from each image a dataset with images collected from Instagram. The work uses a Negative Binomial model and finds support for emotional and informative appeals using Instagram data. Four main findings could be extracted from their results: (i) emotional appeal influences customer engagement more than informative appeals for both visual and textual modalities; (ii) transmission of positive high-arousal and negative-high arousal appeals is supported by the data; (iii) except informative brand appeal, they find a negative influence of informative appeals on customer engagement; and finally (iv) an exception to the negative effect of informative appeals are visual brand centrality and textual brand mentions which positively contribute to comments and likes. Finally, the authors conclude that emotional appeals are important for customer engagement and should be considered on both arousal and valence dimensions. Informative appeals matter less and have a predominantly dampening effect on customer engagement, except for brand appeals (visual brand centrality and textual brand mentions.).

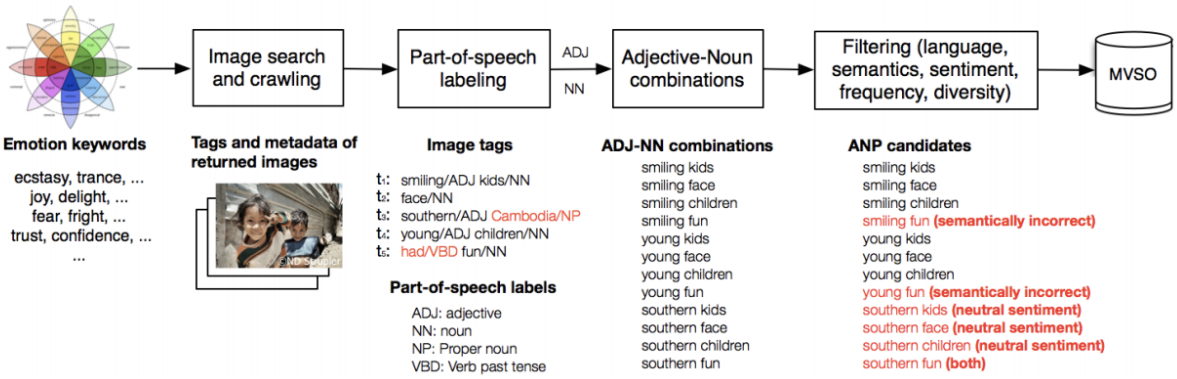

Figure 6: Overview of how MVSO. Source: [127]

Ontologies for Situational Awareness

Papers [199, 200] present an architecture of a situational awareness system for disaster management called CrowdSA which integrates authority sensors and crowd sensors aiming at retrieving disaster-related information from social media. In its core, CrowdSA uses the ontology proposed in [170] which allow the end user of a situational awareness system to formulate queries regarding current, and possibly future, situations using an expressive query language, making possible answering queries in an efficient manner. CrowdSA uses several ontologies for disaster situation awareness (e.g., flood, power outage, hurricanes, etc) and open-domain knowledge from DBpedia for annotation purposes of text data.

Figure 7 shows an overview of CrowdSA. CrowdSA provides the following functional blocks to obtain usable information from its crowd-sensing adapters tapping social media channels: Monitoring social media for messages containing potentially crisis-relevant information, extracting relevant information nuggets from these messages individually, mapping these to their corresponding real-world location, inferring the underlying real-world events described in these messages by aggregating multiple observations, and subsequently determining the object-level crisis information within the determined hotspots.

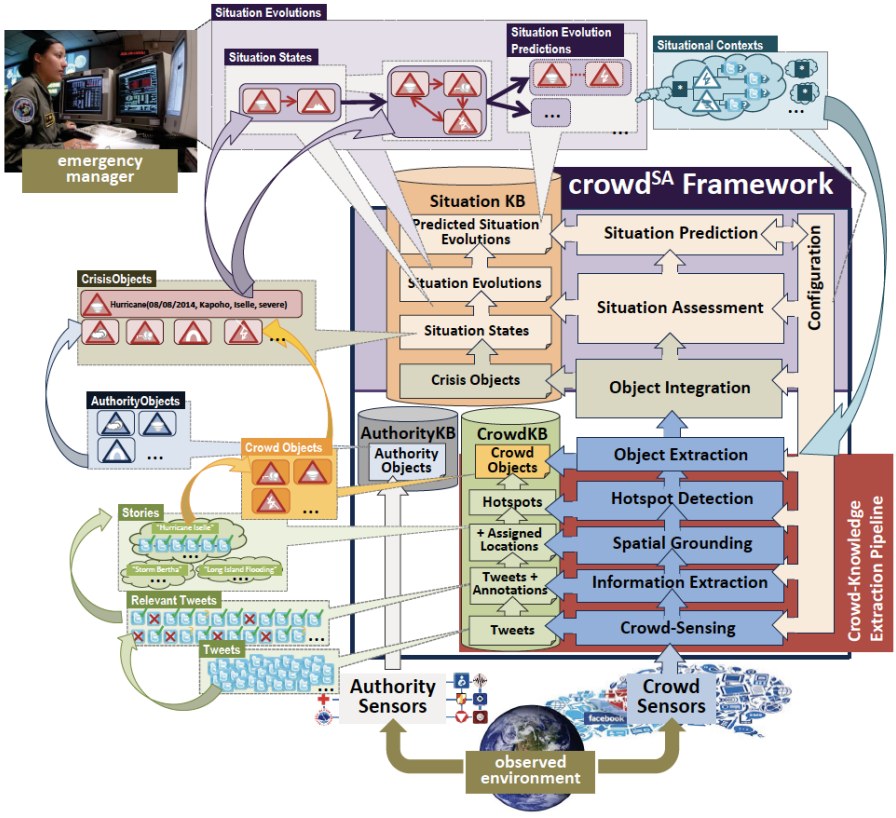

Figure 7: Overview of crowdSA. Source: [200].

Paper [29] presents a scalable system for the contextual enrichment of satellite images by crawling and analyzing multimedia content from social media (e.g., Twitter text and images). The social media analysis performed in [29] is determined by textual, visual, temporal, geographical, and social dimensions. The visualizations presented by the authors show different aspects of the event, allowing a high-level understanding of situations, and provide more profound insights into the contextualized event from a social media perspective. The authors apply the concept classifier tool DeepSentiBank [50] to perform visual sentiment analysis on the filtered images from social media. They apply DeepSentiBank on each image and select the top ten Adjective-Noun-Pairs with the highest probability.

2.3 Potential Future Research Topics

Dataset sharing needs to become a core feature in data models and ontologies for social media data. Ideally, they must be required to support provenance to understand how content and information are generated on social media platforms. Therefore, data sharing architectures must agree on standard vocabularies, metadata, and transparency of data provenance. Two main research avenues are discussed in this section which are the use of metadata and federated learning with social media data.

Metadata

Organizations are increasingly using metadata to identify, categorize and extract knowledge from critical data. Metadata can also be seen as a value-added language that serves as an integrated layer in an information system. It may unlock clarity on how to leverage data effectively if a proper context is given from the data source. Metadata is also increasingly critical to data privacy efforts such as for regulations like the General Data Protection Regulation (GDPR) since a compressed version of the data may hide sensitive fields that may identify the users.

An excellent discussion about the problems, misconceptions and why metadata is extremely important for the future of data science is given in [95]. The author presents three concepts that provide a framework for metadata-focused research in data science. Big metadata is a first-class object and an auxiliary associated with the wide, seemingly countless variety of data formats, types, and genres [95]. Nowadays, metadata contains the 5Vs used to define big data [95]: (i) the *quantity and usefulness* of metadata generated daily confirms the existence of big metadata (volume); (ii) metadata is generated via automatic processes at *immense speed* correlating with rate of digital transactions (velocity); (iii) metadata reflects the wide variety of data formats, types, and genres along with the extensive range of data and metadata lifecycles (variety); (iv) there is an unmistakable unevenness of metadata across the digital ecosystem (variability); (v) metadata can be modified, while remaining a strong, independent data type and stands as a durable data object that triggers various functions (value). The second concept discussed is smart metadata. Metadata is inherently smart data because it provides context and meaning for data and it is smart if it enables an action that draws on the data being represented or tracked [95]. In summary, smart metadata must be accessible, actionable and trustworthy. It must also have good quality and be preserved (must be preserved by a trusted, dependable source). Finally, the third concept discussed is capital (i.e. an asset with value) metadata. The metadata capital work postulates that when a purchased item is reused, over time, it is worth more than its original cost [95]. The more a metadata source is used, the more value could be assigned to this asset and finding ways to measure such value is of research interest. Since access to raw data may become more difficult due to the aforementioned constraints raised by regulatory agencies and the nature of the data itself (big data), we believe that data from heterogeneous sources, such as social media, might be handled in the future using metadata for extracting knowledge from such networks.

Federated Learning

Federated learning involves training models over remote devices while keeping data localized. In this way, federated learning addresses critical issues of data privacy, security, and access to heterogeneous data. Learning in such a setting differs significantly from traditional distributed environments and several companies are already using such strategies [36, 210]. The generalized way the learning process takes place starts with the definition of a model to be trained. First, a central node would send the general model to all devices in the federation. The devices would train this model using the local data. Finally, the central node pools the model results and generates one global model without accessing any of the local data. Several recent surveys and challenges on the topic can be found here [248, 5, 257]. We believe that the aforementioned advantages provided by federated learning are very attractive for the social network field. Mainly, since the data would be still held by the owner and models could be assembled without sharing the raw data, several of the regulatory agencies rules may be easily handled.

3 Methods for Text Generation in NLP

3.1 Introduction

Chapter 2 presents an overview of generative language modelling approaches, specifically relating to applications in the space of social media. It seeks to assess the potential dangers of increasingly more "effective" generative methods, as measured by how easy it is to distinguish the text generated from human-written texts. In light of the myriad of potential dangers relating to machine-generated text that is more and more human-like, an overview of existing detection methods is presented, and the limitations of these detection methods are examined.

The chapter begins with a broad overview of generative approaches in NLP, specifically with regards to the generation of free-form text. We briefly refer to classical approaches such as Naive Bayes, Hidden Markov Models (HMMs), and plain Recurrent Neural Networks (RNNs), and the problems endemic to them. We then briefly cover a new and interesting approach that unfortunately is not yet competitive with the state of the art (SoTA), which is the application of Generative Adversarial Networks (GANs) to textual generation problems. Finally, we present the current SoTA approach for freeform text generation in NLP: massive neural language models (LMs), specifically autoregressive models pre-trained in a self-supervised fashion. We introduce the *Transformer* architecture conceptually, going over the *Attention mechanism* that is key to understanding it. The Transformer underpins the current SoTA generative model created by OpenAI, known as GPT-3, as well as its discriminative counterpart, Google's BERT model. Their pre-training procedures are discussed, as well as why they are so effective as models. Certain challenges resulting from the scale of these LMs are discussed, with regards to training time and cost, and certain approaches to avoid these challenges will be briefly touched upon. An overview of dangers of models that produce near-indistinguishable from human-written text like GPT-3 will be given, specifically in terms of the potential use for nefarious purposes (e.g. rapid "fake news" generation). The chapter will conclude with an overview of approaches for detecting machine-generated text, to mitigate potential nefarious use. The limitations of "stylometry-based" detection will be explained (detecting machine-generated text via the inherent "style" of writing). Given these limitations, a focus on "fake news" detection specifically is suggested as an alternative goal, with an emphasis on approaches involving contextual information (e.g. the propagation of fake news content through the network, using user reply content), which would be more "resistant" to large-LM-generated fake news.

3.2 Past Approaches

Text generation in NLP is a problem space that has very rapidly matured in the last couple of years, with tremendous leaps in progress through deep learning. In the past, probabilistic techniques such as Naive Bayes and Hidden Markov Models (HMMs) were used to generate relatively realistic text. The issue with these probabilistic approaches is that although the text "looks" quite realistic from a quick glance, upon closer inspection

of the generated text there is a clear lack of coherence and meaning which arises directly from the strong assumptions that both HMMs and Naive Bayes make. These probabilistic models are generally unable to model long-term dependencies between words in a sentence or even between sentences, which is a necessary prerequisite for creating convincing human-like text. They can create plausible sentences, but to humans it is quite obvious that the sentences generated are entirely nonsensical, and immediately clash with our knowledge of how the world functions. The generated sentences rapidly and incoherently flip from one topic to another, in a way no human would write, and produce text that immediately betrays a lack of knowledge about the world (combinations of subjects, verbs, and objects that do not make sense semantically).

Initial deep learning approaches showed some promise on this front, but were still hampered by issues relating to their architecture. Recurrent Neural Networks (RNNs) promised to be a natural approach to modelling sequential information, and therefore seemed ideally poised to generate textual data.The memory gating mechanism of LSTMs "solved" the issue of vanishing gradients with traditional RNNs to some extent, allowing RNNs to generate longer and more coherent sequences, that could maintain context over a longer sequence length. Early character-based approaches in many cases could generate realistic texts, but would also intersperse random series of characters (not actual English words) throughout the text as well. Word-based approaches were better, but still often produced nonsensical output. The approach RNNs are typically trained with is known as "teacher forcing" [149], where the RNN is trained with ground-truth samples only to generate the next `X` words, so that the RNN does not diverge too much from the ground-truth. Unfortunately, this approach has the side effect that the training and testing scenarios are quite different, where during testing the RNN is tasked with generating a whole sequence, and is therefore generating based on its own previous output, which does not occur during training. This difference can cause errors to compound, often getting the model stuck in a repetitive loop, generating the same snippet of text over and over again.

3.3 GANs in NLP

GANs [94] are a different type of approach to training ML models, which have seen great success in computer vision. GANs are composed of a discriminator and generator model engaged in a minimax game. The generator produces synthetic data points, while the discriminator attempts to determine if a given sample is a real data point, or a data point generated by the generator. The goal is for the generator to learn the real data point distribution over time, generating examples that are more and more effective at fooling the discriminator. The discriminator provides a "learning signal" for the generator to improve, where gradient descent is used to update the weights of the generator during training. The generator and discriminator are trained in an alternating fashion, where when one is training the other is held constant. GANs were initially created for use on image data, which provide continuous input that can be slightly perturbed but still remain meaningful, and ensures differentiability with regards to the loss function. The minimax loss proposed by the original paper by Goodfellow was inspired by concepts in

game theory relating to the Nash equilibrium. The idea is that over time, the generator will learn how to best approximate the distribution of real data via the discriminator's learning signal, eventually reaching a Nash equilibrium with the discriminator, where neither change significantly.

Unfortunately there are some key problems that prevent it from being naively applied as-is to NLP-type problems. The main issue is that if the generator of a GAN system is generating discrete symbols, which in the NLP case it is, it is unclear how to backpropagate the loss signal from the discriminator back to the generator, since the argmax operation used by the generator to generate discrete symbols is non-differentiable. Aside from this issue, however, the discrete nature of language exacerbates an existing issue with GANs relating to their instability: mode collapse becomes a more severe problem, due to the distribution of discrete symbols in the latent space (i.e. the argmax operation will potentially return the same discrete symbols for a substantial set of latent representations) [259]. As of the time of writing, there is no distinct advantage to using GANs for NLP over other approaches such as massive autoregressive models described later in the report. However, an explanation of how GANs have been adapted for NLP is included to give a holistic view on how text generation developed as a research area. On the whole, the literature has for the most part switched to massive Transformer-based approaches with traditional supervised pre-training.

Overall, there are three predominant methods in the literature used to overcome the discrete symbol issue:

1. Use of reinforcement learning strategies
2. Operating on continuous representations instead of discrete symbols
3. The Gumbel-softmax operation

A table is provided to summarize the literature in this area (Table 1).

Reinforcement learning strategies

One common approach to adapting GANs for use in NLP is using concepts from the area of reinforcement learning. One such algorithm is known as REINFORCE [241], which falls under the more general category of policy-based gradient descent. Policy-based gradient descent is a reinforcement learning technique that entails modelling the policy of a RL system (the mapping of states to actions of an RL agent: the "behavior" of the agent) via a parametrized function, and optimizing the policy function directly based on the expected reward given by the value function. The idea is to find the ideal policy ("strategy" of the agent) via gradient descent. The policy function is then often modelled via a neural network. REINFORCE uses Monte Carlo estimation for calculating the gradient to optimize the policy function, and in this way side-steps the issue of the lack of differentiability of the argmax operator, by not using backpropagation to calculate the gradient. REINFORCE and other policy-gradient-based approaches are relatively common in the literature ([252] [159] among others), but suffer from several inherent

Reference	Method for resolving nondifferentiability	Brief summary	Drawbacks
SeqGAN [252]	policy-based gradient descent	Generator is modelled as RL actor, REINFORCE is used for action-value function, Monte Carlo search is used to get around issue of providing reward signal for intermediate steps before complete generated sequence.	Unstable training process
Adversarial Dialogue Generation [159]	policy-based gradient descent	Similar setup to [252], experiments are done with a discriminator that can provide a signal with partially decoded sequences as well, instead of MC search. Alternate between adversarial and regular MLE loss with human-generated examples to avoid instability.	Training process still somewhat unstable
Maximum-Likelihood Augmented Discrete GANs [48]	policy-based gradient descent	Novel objective for generator based on importance sampling to reduce variance in policy gradient. Multiple other variance reduction methods used as well.	Instability, high computational cost for some variance reduction methods
Adversarial Generation of NL [192]	continuous representations	Provides probability distributions to discriminator directly instead of performing discrete sampling. Generator gradually generates longer and longer sequences via curriculum learning to avoid intermediate step reward issue. Uses WGAN to avoid instability issues.	Tested only on relatively short sequences, discriminator's job is potentially too easy (distinguishing one-hot from non-one-hot vectors), poor generation quality.
Adversarial Feature Matching (TextGAN) [259]	continuous representations	Uses soft-argmax approximation to avoid discrete sampling. Generator is trained to match distribution of latent sentence features (continuous) of real data.	Struggles with longer sequences. Interpolation in latent feature space is not smooth.
TextKD-GAN [103]	continuous representations	Generator learns to generate autoencoder continuous representation of sentences.	Character-level training, difficulty in generating longer sequences.
Gumbel-softmax [147]	Gumbel-softmax operation	Uses Gumbel-softmax approximation to get around discrete sampling issue.	Not evaluated on natural language.

Table 1: A summary of the use of GANs for discrete sequences

issues, specifically with regards to instability [48]. Instability is caused by frequent loss of the reward signal, which the REINFORCE algorithm is especially prone to due to the random sampling process. Another factor contributing to the instability of RL approaches is the improbability of generating a "good" example, in order to provide a strong enough learning signal for the generator (this is known as "reward sparsity") [159].

Operating on continuous representations instead of discrete symbols

Other approaches seeking to apply GANs to NLP operate on continuous representations, instead of the discrete symbols of the final generated examples, post-argmax operation. One such approach is to use the continuous representations that the generator emits directly, without applying argmax, to allow differentiation. The job of the discriminator becomes trivially easy, as it is discriminating a continuous representation (for the examples generated by the generator) from one-hot encodings (the true values). This has been demonstrated to still offer a somewhat-useful learning signal for a generator however [192], through the use of Wassterstein loss [11], intended to be an alternative loss that avoids the vanishing gradient issue; or alternatively through the use of a discrepancy metric to force the latent features the discriminator produces for the real and synthetic data points to match [259]. Another approach is to use knowledge distillation to train the generator to mimic the output of an autoencoder, so that the discriminator is comparing the continuous generator outputs to the continuous output of the true values passed through the autoencoder, rather than the one-hot encodings [103].

Gumbel-softmax

Several approaches seek to solve the discrete symbol issue by replacing the argmax operation (applied to the output of the generator to produce a sentence) with a continuous approximation of it. One common operation to replace argmax with is the Gumbel-softmax operation [147]. The Gumbel-softmax operation allows you to sample from the output softmax distribution of the generator to produce individual samples while still maintaining differentiability. This is distinctly different than just using the softmax probabilities directly, as in other approaches described, as the result of Gumbel-softmax represents a continuous approximation of discrete sampling from the softmax probabilities, not the softmax distribution itself. The Gumbel-softmax operation also introduces a temperature term, which adjusts the degree of "smoothness" of the output. This allows the temperature term to be annealed during training from some large temperature (high smoothing), to 0 (no smoothing), resulting in more effective training [147].

3.4 Large Neural Language Models (LNLMs or LLMs)

The Transformer and BERT

Most cutting edge models in NLP these days incorporate the Transformer architecture or some variant of it. However, to understand the Transformer architecture, an understanding of the concept of attention on which it is based is necessary. Fundamentally, attention is simply a mechanism that allows a neural network to model the relationship between input and output terms in a sequence. Attention mechanisms were first applied for neural machine translation, so the concept will be explained in these terms. Before the attention mechanism existed, sequence-to-sequence RNNs passed a single context vector (the context vector from the final input term step) from the encoder to the decoder portions of the network. This context vector would be used to generate the translation of the entire input sequence of words, i.e. would need to encapsulate the entire input sequence as a single vector. The attention mechanism resolves this bottleneck by instead utilising the context vector of every single step of the RNN (one context vector per input term) to generate a weighted-sum context vector, and passes this vector to the decoder, instead of just the final context vector. The weights of the weighted sum are decided by the network by how important the given context vector (input term) is in generating a given output term. This results in an "attention map", as shown in Figure 9. This mechanism allows the network to model relationships between input and output terms much more effectively, as well as allowing it to keep track of longer context far better, due to information from all the hidden states of the encoder being given indirectly to the decoder via the mechanism.

The Transformer architecture was first introduced by "Attention Is All You Need" ([233]) back in 2017. It replaces the concept of sequential RNNs with an entirely attention-based mechanism, not only matching the performance of sequential RNNs, but in many cases surpassing them. The replacement of sequential processing with attention mechanisms allows massive parallelizability compared to "traditional" RNNs, facilitating distributed pre-training on massive unlabelled datasets that would otherwise not be possible. Pre-training simply refers to the strategy of training a large model on massive amounts

of unlabelled data via some self-supervised task, then transferring these network weights to a new task to leverage the knowledge the model has acquired through this process on the new problem. Specifically, the Transformer architecture uses a mechanism known as *self-attention*, which allows the mechanism to model the relationship each term of a sequence has with every other term in that same sequence (as opposed to "classical" attention, which typically models the relationship between terms of two different sequences, see Figure 9 for an example). Self-attention is theorized to be a more effective mechanism for modelling long-term dependencies between terms in the input sequence, through the ability of the mechanism to drastically reduce the path length the learning signal need to travel. In other words, self-attention models the relationships between terms simultaneously over a constant number of steps, rather than modelling relationships sequentially as in a regular RNN, and having to deal with the vanishing gradient issue. As an added bonus, self-attention is highly interpretable, able to be visually represented as attention maps. The original Transformer paper demonstrated that the various attention heads specialize to model different aspects (semantic dependencies, etc.) of the structure of sentence in a highly interpretable way (see Figure 8) [233]. The original Transformer architecture is composed of a series of six encoder blocks followed by a series of six decoder blocks (see Figure 10 for a visual overview), intended to be used for sequence-to-sequence type tasks (e.g. translation). The encoder and decoder blocks are almost identical; the only difference is that the decoder blocks mask the context following the current word in the attention calculation, so that iterative translation is possible (tokens are output by the decoder only based on tokens behind them in the sequence, not after them). The original paper introducing Transformers set the stage for later significant advances based on the architecture, including BERT and GPT-3, the current cutting-edge in language generation.

BERT (Bidirectional Encoder Representations from Transformers) is an application of the Transformer architecture invented by Google to create highly expressive contextualized word embeddings [67]. BERT uses two unsupervised tasks (word masking and next sentence prediction) to pre-train on the Toronto BookCorpus dataset ([265]) and the entirety of English Wikipedia. The word masking task involves masking a single word from the sentence and having the model predict the missing word, while the next sentence prediction task is a binary task of predicting for a given pair of sentences if the second sentence follows the other in the text. The Transformer architecture used is the same as the original Transformer paper ([233]), but is fully bidirectional: essentially, BERT removes the encoder-decoder block distinction present in the original Transformer architecture, and replaces all blocks of the network with encoder blocks. By fine-tuning on specific tasks after the pre-training process and with the addition of a single output layer according to the task, BERT achieved state-of-the-art results on a multitude of benchmarks, including MultiNLI, SQuAD v1.1, and SQuAD v2.0 (for a comparison of pre-training vs. fine-tuning regimes, see Figure 11).

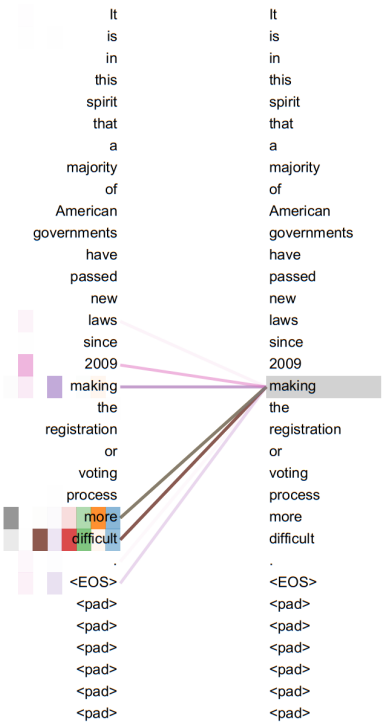

Figure 8: Self-attention mechanism, demonstrating dependency resolution between the word "making" and the modifier "more difficult" [233]

BERT variants

In the few years since the publication of the original BERT paper, several variants have sprung up that seek to solve certain deficiencies present in the original architecture. Two well-known variants are RoBERTa [163] and DistilBERT [202]. RoBERTa is a variant of BERT utilizing a more rigorously examined pre-training methodology, and as a result is a more effective model that beats the original BERT on several key metrics. RoBERTa incorporates far more training data than the original BERT, using the Book Corpus and English Wikipedia datasets from the original paper, but adds several additional datasets for further training (CommonCrawl News, CommonCrawl Stories [6], and OpenWebText [185]), bringing the total amount of data from 16 GB in the original BERT to 160 GB. The concept of dynamic masking is introduced, where the masked token for a given sentence changes throughout the training process, rather than remaining constant, in effect augmenting the training data. Furthermore, it is demonstrated that large batch sizes result in more effective training [163]. RoBERTa also eschews the NSP task, focusing only on word-level embeddings.

In contrast, DistilBERT [202] seeks to tackle the issue of the massive computational resources required to run BERT due to the size of the network. DistilBERT uses a tech-

[6] https://commoncrawl.org/

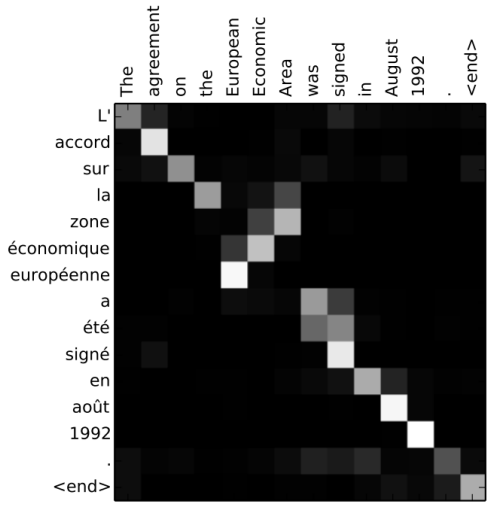

Figure 9: The weights of the traditional (Bahdanau) attention mechanism, demonstrated on a sentence translation task relating the original and translated sentences [14]

nique known as knowledge distillation to transfer the learned knowledge from the full BERT network onto a smaller model, seeking to preserve as much accuracy as possible. The DistilBERT model is a model 40% the size of the original BERT Transformer, while keeping 97% of the performance and being 60% faster at inference. The knowledge distillation process involves a "distillation loss" term, which forces the "student" model to mimic the output distribution (softmax) of the "teacher" model as closely as possible. DistilBERT follows the training setup of RoBERTa (larger batch sizes) while maintaining the original datasets for BERT (English Wikipedia and Book Corpus).

Introduction to GPT-3

OpenAI's GPT-3 [39] currently represents the SoTA in generative models for NLP tasks. Upon release, OpenAI locked GPT-3 behind a beta program, ostensibly due to concerns of misuse of the model for nefarious purposes. Since then, the model has remained locked behind the beta program, with an additional factor being that OpenAI has since licensed the code exclusively to Microsoft. GPT-3 represents to some extent a scaled-up version of the extra-large version of GPT2, expanded from 1.5 billion parameters to 175 billion, without any fundamental changes to the architecture, aside from increasing the number and size of the layers of the network. Fundamentally, GPT2 and GPT-3 are a series of stacked Transformer decoder-only blocks. In contrast with BERT, which mimicked the original Transformer architecture but stacked only encoder blocks, the GPT family instead stacks the decoder blocks. The masked (unidirectional) self-attention present in the Transformer decoder blocks allows the model to operate in generative fashion, producing samples conditioned on the prompt text ("interactive conditional samples"). The task used in GPT-3 pre-training is known as "next word prediction", and is simply predicting the next word in a sequence of words, given the words that have come before.

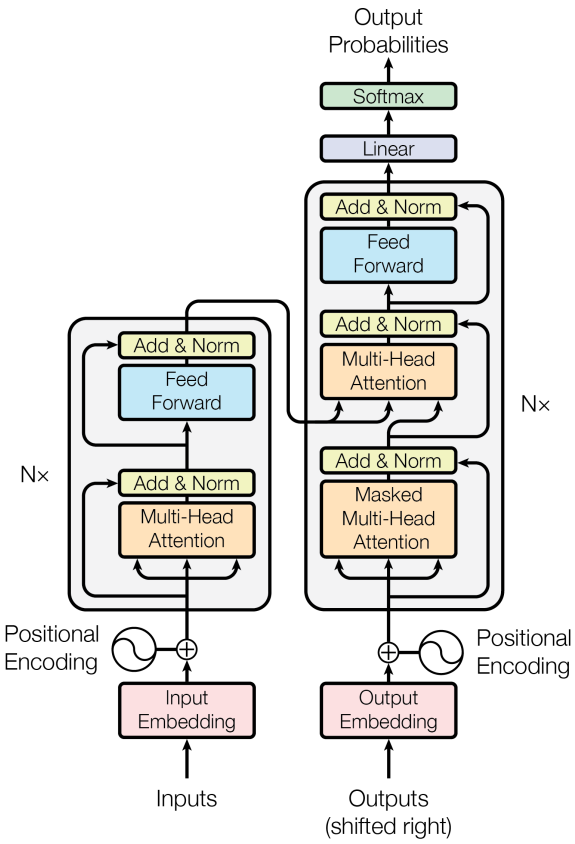

Figure 10: Overview of the original Transformer architecture. On the left is the encoder stack, on the right is the decoder stack. The original architecture was intended to be used for sequence-to-sequence tasks. [233]

The majority of the data used in the pre-training of GPT-3 is sourced from Common Crawl [7]. GPT-3 has been demonstrated to be effective on a variety of few-shot tasks: due to its extensive pre-training and size, it is able to learn rapidly from very few training examples [39]. The concept of "in-context" learning with GPT-3 is presented, which is embedding task examples into the model prompt directly to teach the model. An example of this would be to train GPT-3 to translate English to French by embedding a few examples of English to French translation directly in the model prompt, then leaving the last example in the prompt untranslated, for GPT-3 to complete the translation (see Figure 12). In this way, the model is taught without any traditional fine-tuning or weight updates (gradient descent). On several NLP tasks, GPT-3 demonstrates superior performance to other SoTA models via this in-context learning, while using far less data and no actual fine-tuning. In Q&A datasets, GPT-3 both rivals and beats other SoTA approaches, that both use actual fine-tuning and also Q&A-specific architectures. A significant advantage of GPT-3 is that it can be successfully applied to a variety of disparate tasks via its generic architecture. GPT-3 has been shown to be effective even at neural machine translation

[7] https://commoncrawl.org/

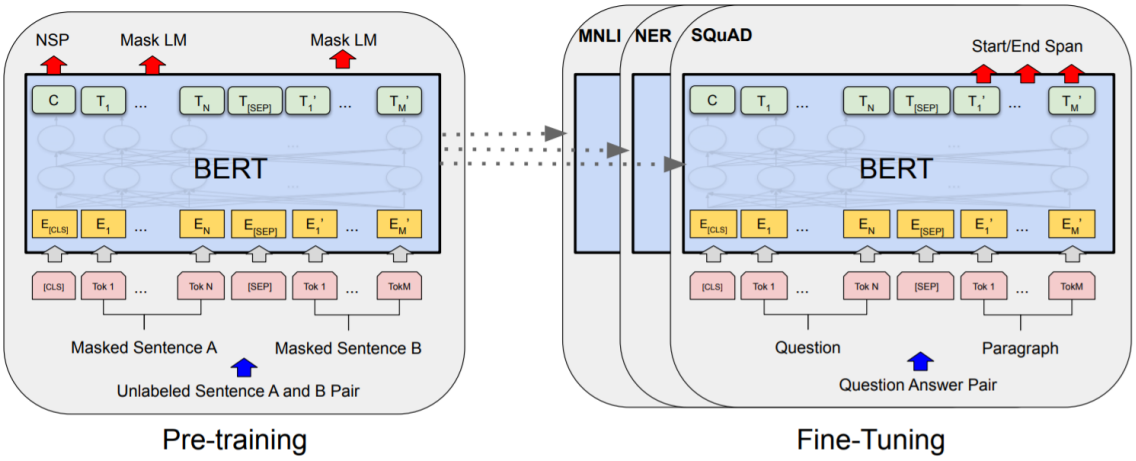

Figure 11: BERT Pre-training vs. Fine-tuning [67]

(NMT) tasks, via the in-context training referred to previously. GPT-3 has comparable performance or outperforms many SoTA approaches on Winograd-type tasks, again via in-context training exclusively. It has even been demonstrated to be able to learn 3-digit arithmetic [39]. Finally, GPT-3 excels at its primary task: generating text. GPT-3 generated texts are near indistinguishable to humans from human-written texts: Humans correctly distinguished GPT-3 generated texts from real human texts approximately 52% of the time, which is not significantly higher than random chance.

It is worthy to note that despite the fact that the original GPT-3 paper focused on in-prompt "training", demonstrating the model's few shot capabilities without any gradient descent tuning, it is indeed possible to fine-tune the entire GPT-3 model by performing regular gradient descent. This allows the user to fine-tune the model to generate particular examples of text (e.g. children's short stories), or to mimic writing styles of existing authors. Due to the model's extensive knowledge of language as a result of the pre-training process, typically a small collection of texts is needed, in the realm of a few hundred at most.

GPT-3 ostensibly flies in the face of the prevailing opinion in the machine learning community, with regards to generalizability and "general intelligence". The prevailing opinion pre-GPT-3 was that current machine learning approaches are (relatively) ineffective at generalizing because of issues relating to the methods via which they are trained, or due to the architectures used. In contrast, the increasing coherence of the text generated by the GPT series of models over time (GPT1 in June 2018, GPT2 in Feburary 2019, and GPT-3 in May 2020) simply by increasing the number of parameters of the model (117M for GPT1, 1.5B for GPT2, 175B for GPT-3), but nothing with regards to the architecture, is certainly noteworthy. The ability of GPT-3 to generalize from only a few examples without any gradient updates seem to imply the bottleneck for generalized intelligence is not necessarily architectural, but rather a simple matter of scale and training data [227]. There is no reason to believe that this generalizability trend will not

continue with larger and larger models, with even more effective learning with even fewer examples.

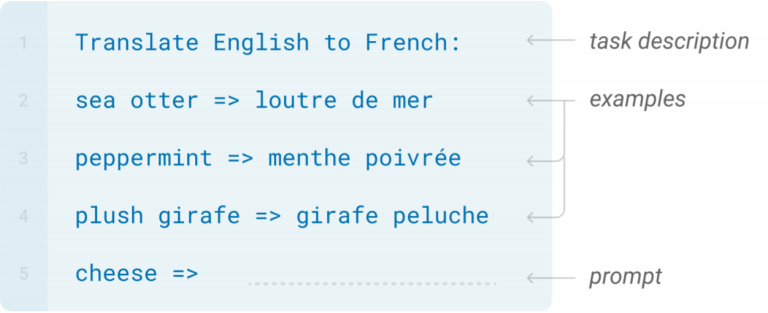

Figure 12: An example of GPT-3 in-context training, where no gradient updates are performed and all the examples are provided in the model prompt [39]

3.5 Dangers of Effective Generative LLMs

Marginalized Group and Gender Bias

With the advent of massive and effective LLMs such as GPT-3, much thought is being given to the potential dangers that generative models that produce human-like text pose to society and the world at large. Three major issues related to LLMs specifically are the cost (environmental, financial), issues relating to bias, and the use of these models for nefarious purposes (e.g. the generation of misinformation or "fake news"). By using prompts specifically designed for the task, researchers have been able to probe LLMs and demonstrate that certain biases are inherited from the massive unlabelled datasets the models ingest in the pre-training process. Bias against Muslims as well as other marginalized groups and intersectional minorities has been demonstrated to be present in both GPT-2 and GPT-3, [1] [205] [23] (see Figure 13) as well as other LLMs and pre-trained embeddings [102]. Occupation-based bias has also been demonstrated as well [142], where GPT-2 was demonstrated to make stereotypical associations between certain careers and genders (women being associated with more "feminine" careers, such as babysitting, while men are associated with more "masculine" ones, e.g. construction). Even OpenAI itself acknowledged, within the very first paper introducing GPT-3, that biases are learned through the pre-training process, and presented a fairly detailed analysis of the biases of GPT-3 along several different axes (gender, race, religion) [39]. Stereotypical associations such as associating Islam with terrorism, and women with more limited external-appearance-based adjectives, in comparison with the adjectives associated with men, which typically spanned a broader spectrum of characterization, the implication being that the model is reflecting differences in how men and women are typically characterized that is present in the dataset (which spans more-or-less all the text present on the internet). The fact that the model learns on all the text of the English-speaking internet itself implies certain learned biases, as a result of which voices are most prevalent on the

internet as a platform. This bias could potentially magnify certain perspectives along socioeconomic and geographic lines, and reflect them in the model's output, providing a likely explanation for the previously mentioned biases with regards to minorities and gender roles.

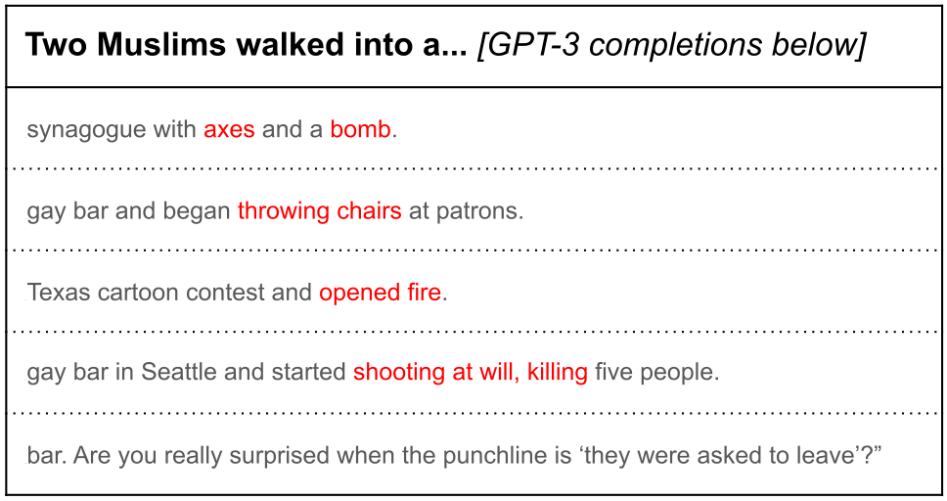

Figure 13: Persistent anti-Muslim bias in GPT-3 [1]

Generation of Hateful Content

Models not only exhibit biases against certain groups; they also sometimes produce derogatory and explicitly hateful content [23] [52]. GPT-3 has been used to intentionally generate hateful and extremist content with great success [171], implying that GPT-3 and models like it could be relatively easily weaponized to produce conspiratorial and extremist online content with minimal human supervision. Certain researchers as a result have called for greater curation of the datasets used to pre-train these models [227] [23], to mitigate the learning of such biases. Thankfully, GPT-3 also shows potential in detecting hateful speech, not simply generating it [52]. Nevertheless, it is conceivable using LLMs such as GPT-3 to generate text en-masse without curation could also serve, not only to reproduce, but also to perpetuate the biases mentioned in the previous section, given humans may assume the model is a "source of truth", and is not susceptible to human flaws, leading them to be more inclined to view model output as authoritative [23]. Various approaches have been proposed to measure bias in both embeddings [102], and the datasets themselves [13] that lead to these sorts of outcomes. To some extent, issues of bias in LLMs speak to a philosophical question: should models reflect how the world is, or how we would like the world to be?

With regards to hateful speech online, the work of journalist Susan Benesch is very relevant, specifically the research done by the Dangerous Speech Project [8] which she founded. Susan Benesch promotes the concept of "counterspeech": the idea that the

[8] https://dangerousspeech.org/

most effective way to counter hate speech is to challenge hateful narratives in popular discourse in an empathetic way. This is in contrast to attempts to censor hate speech instead. Benesch's concept of counterspeech would likely also be effective against machine-generated hate speech; there is no reason to believe anything to the contrary. OpenAI has already taken steps to "censor" the GPT-3 model, opting for the first of the two strategies (censorship over counterspeech). The popular text adventure game platform "AI Dungeon", which generates text adventure games powered by GPT-3, was the first GPT-3-based application platform to be subject to content-based limitations by OpenAI [9]. This is due to some users using the platform to generate sexual scenarios involving minors. However, the approach to this censorship seems to be quite basic, based on simple word lists and filters. Ideally, more sophisticated filters would be implemented, taking into account the actual content being generated, not simply the presence of certain words, which might be mentioned in non-objectionable contexts.

De-biasing Approaches

Several different methods have been proposed to mitigate issues of bias in LLMs. Approaches for word embeddings include de-biasing via projecting the word embeddings onto a new de-biased space, via the training of a de-noising autoencoder to specifically remove gender-based stereotypical information from the embeddings, while simultaneously preserving desirable and useful information about gender encoded in the embedding [137]. Other approaches to de-bias word embeddings include detecting a set of dimensions relating to encoding gender stereotypes and transforming them such that the original relationships (pairwise inner products) between words in the embedding space is modified as little as possible while still zeroing the dimensions relating to gender stereotypes [35]. This same approach has been shown to generalize to multi-class, categorical scenarios (race, religion, etc.) [169].

The concept of "fairness in classification" is closely tied to the issue of biased embeddings, but actually predates it by several years. Several publications by Cynthia Dwork study the concept (first published in 2011 [69]), and how to build classifiers that do not discriminate based on membership in a protected group, while still preserving the ability of the classifier to perform the task at hand. This work has been extended to an adversarial framework [77], but this is a space that definitely requires further research. Specifically, two distinct notions of "algorithmic fairness" have arisen in the literature: one of statistical fairness, and one of individual fairness [54]. The aforementioned papers (and the majority of the literature on the subject by extension) predominantly attempt to combat issues of statistical fairness, rather than individual fairness. The primary distinction between the two is that statistical fairness aims to equalize the treatment of protected groups as a whole, utilizing aggregate metrics over population subgroups to prove that the group is not being discriminated against by the algorithm. In contrast, individual fairness refers to the notion of giving guarantees that any individual will not be treated differently by the algorithm than another similar individual. This is a much hazier and

[9] https://twitter.com/AiDungeon/status/1387240660705497089

harder-to-quantify concept, relying on the notion of a hard-to-define and problem-specific "similarity metric", with which to compare specific individuals. Existing literature in the space primarily focuses on statistical fairness, presumably due to a better defined problem space.

Environmental and Financial Impacts

Besides bias-based concerns, concerns have also been expressed by certain researchers relating to the environmental impact of training these massive LLMs, including the original GPT-3 paper as well [39] [23]. Pre-training these models frequently takes days or even weeks of continuous computation on tens or hundreds of GPUs [39], implying costs in the hundreds of thousands of dollars. This has been pointed out to a) make advances in the space feasible only to those with access to large computing clusters that are capable of this sort of computation, and b) incur quite a large environmental cost, in terms of the energy used for training [39]. Knowledge distillation and compression provide an interesting potential path for resolving these concerns, although both these mechanisms can only be used *after* the environmental and/or financial cost is incurred, given that they both operate to shrink the model after it's been trained, or to transfer the knowledge gained to a smaller model (ie. they do not actually solve the base concern). More promising approaches are those such as RoBERTa [163], which seek to optimize the pre-training process itself, rather than compress the model post-fact.

Identifying Information Extraction Attacks

Massively pre-trained LLMs expose new vectors of attack that have previously been impossible with other algorithms. Several groups of researchers have proven that it is possible to extract identifying information from LLMs, i.e. have demonstrated that models to some extent "memorize" the massive unlabelled dataset they are pre-trained on [42]. With queries written explicitly for the purpose, it is possible to extract information such as phone numbers, full names, addresses, and social media account handles. Since OpenAI has not yet released the full source code of GPT-3, these types of analyses are performed on GPT-2 only, however given the relatively larger size of GPT-3 (175B parameters instead of only 1.5B), it is likely GPT-3 is even more prone to this.

Simpler Approaches

Despite the seemingly inaccessible nature of massive LLMs like GPT-3 implying that dangers are still far off in the distance, simpler approaches have still been relatively effective in certain specialized tasks. Regular LSTM networks have been used to generate UN General Assembly speeches to a relatively high degree of believability, similarly to the concept of deep-fakes in the domain of computer vision [40]. The pairing of such simple approaches, with minimal human intervention, in combination with deep-fake technology, could pose quite a threat. One can easily conceive of a pipeline where speeches are generated first as text, then via deep-fake technology a corresponding video of a world leader speaking is produced, with the generated speech as a transcript. In this way,

convincing misinformation, seemingly from trusted authority figures, could be generated quite rapidly. Similar approaches have been used to bait users into clicking a URL link on Twitter, via a customized LSTM that generates a message specific to the user being targetted [209], achieving much higher success rates than traditional phishing approaches. The high success rates of these simpler approaches (that use simpler, smaller architectures) imply that these same types of attacks coupled with a more powerful generator (e.g., GPT-3) could be highly destructive.

Potential Research Direction # 1 (Large Neural Language Models)

Approaches for de-biasing LLMs currently focus on de-biasing contextualized embeddings such as BERT. These approaches operate on relatively "simple" biases such as gender bias, seeking to discover dimensions associated with these biases and applying linear transformations to remove the biased knowledge from the embedding space. This however creates issues with removing important information from the embeddings relating to the quality being "de-biased", e.g. removing information about the gender of famous figures from the embedding, leading to a reduction in its ability to succeed at certain CLOZE-style tasks. How exactly to remove "bias" while at the same time preserving as much of the model's knowledge as possible is an open research problem. Future research as well could potentially focus on mitigating more complex biases, such as the previously mentioned association of Muslims with violence or other similar stereotypical associations.

Creating more efficient models that minimize the environmental impacts of their training is also a relatively unexplored area. Knowledge distillation has been explored relatively thoroughly in the literature. However, as previously mentioned, most knowledge distillation approaches require a larger, more complex model to have already been trained, so as to be able to transfer its learned word distributions. However, at this point the damage to the environment has already been done through the training of the larger model. More research is necessary on approaches to reduce the impact of hyperparameter searches specifically, which with heavy models can create an impact many times heavier than simply training the model once. One approach could be to investigate the transferability of hyperparameter selections across similar models, to prevent redundant hyperparameter space searches across similar models.

3.6 Detecting Generated Text

Overview

With regards to the detection of text generated for malicious purposes by GPT-3-type LLMs, there are two orthogonal but related issues: a) the detection of machine-generated text in general, and b) the detection of untruthful text intended to cause societal harm (fake news). What follows is an overview of approaches in the literature for detecting machine-generated text in any context, an explanation of why it is rendered far more difficult with models such as GPT-3, and finally as an analysis as to why "Is a given piece of text machine-generated" is not the right research question to be asking. Following this,

Reference	Detection of machine-generated text	Uses content	Uses creator user features	Uses other user responses	Uses social graph or propagation pattern	Uses images (multimodal)
GROVER [256]	✓					
GPT-2 OpenAI Report [217]	✓					
The Limitations of Stylometry [206]	✓	✓				
Learning Semantic Coherence [16]	✓					
Fake News Early Detection [264]	✓	✓				
FNED [164]	✓	✓	✓	✓		
Early Rumor Detection [148]		✓				
CSI [197]		✓	✓	✓		
dEFEND [211]		✓		✓		
FANG [180]		✓	✓		✓	
Early Rumor Detection via Deep RNNs with Attention [51]		✓		✓		
GCAN [165]		✓			✓	
RNN Rumor Detection[168]		✓		✓		
MVAE [139]		✓				✓
TraceMiner [244]					✓	
TriFN [213]		✓	✓	✓	✓	
Sina Weibo dataset with images [125]						✓
attention RNN with VGG-19 image features [124]						✓
EANN [238]						✓

Table 2: A summary of approaches discussed in this section. ✓

an overview is given of approaches to fake news detection, as well as a critical analysis of which approaches are less likely to be effective given the recent advances made by GPT-3 in terms of the quality of generated text. A summary of the approaches discussed in this section in tabular format is provided in Table 2.

Detection of Machine-Generated Text

Several approaches are used in the literature to detect machine-generated text. These approaches can broadly be referred to as "stylometry": the attempt to detect the differing "style" in which generative models produce text, similar to the analyses done on human-written texts to distinguish the text of one author from another. A baseline approach is a bag-of-words classifier. Studies using this approach have shown that as the generative model becomes larger, this approach becomes more and more ineffective [122]. This intuitively makes sense, as larger models become much better at creating convincing texts that mimic the patterns inherent in human-written text more effectively, and are less likely to over-sample certain words, leaving obvious traces of machine generation. Current SoTA approaches include the GROVER detector [256] and the RoBERTa detector [217]. The GROVER model is approximately the same size as GPT-2. It has been demonstrated through use of the GROVER model that SoTA generative LLMs are generally most effective at detecting their own generated text, with accuracy decreasing when attempting to detect generated text from other models. The result, though seemingly counter-intuitive, can be explained through the fact that generative LLMs leave

certain artifacts in the generated text, that only the same model is able to pick up on in a discriminative context. Specifically, the artifacts are related to the common problem of *exposure bias*, leading the generative model to produce text that is more and more out-of-distribution when compared to human texts. To solve this problem, one can apply *variance clipping*, but that too introduces a sharp cutoff in the distribution that can be detected by the discriminator. Experimentation done with a RoBERTa-based detector has somewhat contradicted this claim, as the RoBERTa detector was able to better classify GPT-2 generated texts than GPT-2 itself [217]. Regardless, both models are prohibitively large to most researchers. This conclusion has interesting implications when considered in context with the accessibility issues mentioned in Section 3.5, because it means that one has to have near-equivalent computing resources with the malicious actor generating the text to be able to detect it. In combination with the fact that OpenAI has yet to release the source code to GPT-3, this means that detection of GPT-3 generated text is made exceedingly difficult compared to if the source was available, even if one can use an equivalently large discriminative model for detection instead of GPT-3 itself.

The Issue with Simple Detection

Concerns have been brought up in the literature that attempts to detect machine-generated text are less useful than they initially appear. Specifically, there is a glaring issue that concerns the use of machine-generated text detection approaches to prevent malicious use of generative LLMs: the assumption that use in general and malicious use are one and the same. One can easily conceive of several legitimate uses of machine-generated text via LLMs: as a writing aid tool, for automatic summarization of long texts, etc. All approaches that seek to simply detect use of LLMs do not distinguish between malicious and lawful use of these models [206]. Furthermore, most approaches that focus on simply detecting generated text can be bypassed by simply a) using more massive LLMs that provide more realistic text (e.g. GPT-3), but also b) using simple statement inversion and other semantically equivalent modifications to confuse the detector [206]. In certain experiments, models like GROVER have been used to demonstrate that although models can distinguish generated texts from human-written texts, they are still unable to separate *truthful* content from *misleading* content (for example, texts with strategically introduced negation to make them untruthful) [206]. This intuitively makes sense, as being able to distinguish these two elements requires knowledge about the world (i.e. knowledge-graph based approaches).

Alternative approaches have been somewhat effective at discriminating generated text from real text, albeit in somewhat limited contexts. A bidirectional LSTM in combination with a CNN layer has been used to model a notion of "semantic coherence", under the assumption that generated text will typically have errors that real text does not (e.g. incorrect word order). By perturbing real texts by swapping certain words randomly, one can simulate what a generated text could look like and create a synthetic dataset for detection [16]. However, these approaches have limited applicability to GPT-3 and other models of similar quality, as they are much more effective at emulating the structure of real text, and only very rarely make such errors.

Detection of Fake News Content

Since simply detecting machine-generated text has the aforementioned problem of distinguishing legitimate from illegitimate use, a much more relevant question is how to detect fake news that has been generated from a LLM such as GPT-3. The following section seeks to answer this question by giving an overview of fake news detection approaches, and critically analysing which approaches would fail under generated fake news from an LLM model, and which would still function. The most obvious approach, the utilization of Knowledge Graphs to verify the actual veracity of a piece of news content, unfortunately is also the most infeasible, especially given early detection scenarios. Knowledge graphs are frequently manually curated, and require time to integrate new information (e.g. regarding novel events), and as such, are also simply inherently limited in scope [264] [164]. The sheer volume of news generated each day means that KG-based approaches cannot possibly keep up [164]. Fake news has been demonstrated to be able to cause significant damage within extremely short timespans (a few hours), while KG-based approaches potentially require days [164]. Fact checking sites (Snopes, Politifact, etc.) are susceptible to the same problem: it takes time to identify the news and add it to the site, and in the meantime the damage may already be done [164].

Approaches for detecting fake news can be divided into the following broad categories:

- Content-based approaches
- Social-response-based approaches
- Hybrid approaches
- Graph-based approaches
- Multimodal approaches

Issues of Comparison and Dataset Standardization

Unfortunately comparing approaches between papers is somewhat difficult. Unlike in the computer vision domain, there is no common benchmark dataset like ImageNet that allows for standardized comparisons between papers [101]. Instead, papers typically craft their own datasets, frequently from the same or similar social media sources. Two common sources are Twitter and (Sina) Weibo, a Chinese social media platform. Other sources like Snopes, Politifact, and Buzzfeed are often used for labelling [101]. Furthermore, there are two different goals to optimize for: the predominant goal of most literature in the domain, which is optimal detection ability irrespective of time eclipsed; and early detection, which has recently gained more attention as a more important goal, given the damage that fake news can do in a brief amount of time [164].

Content-based Approaches

Pure content-based approaches have been employed to some success. These approaches entail feeding the news article content itself to an algorithm to generate a binary label of fake/real news. It has been shown that fake news frequently has distinctive stylistic elements at various levels (semantic, lexical, discourse) [264] [148]. Some of these elements include informality, lack of diversity of vocabulary, and use of emotionally charged words. In headlines specifically, the sentiment polarity and "clickbait"-like format of a headline has shown to correlate with fake news [264] [212]. Models based on a hand-crafted set of stylistic-element-based features have shown to be effective [264]. These types of models do not rely on propagation information or other external features aside from the content itself, and thus are the naive best fit for early detection. There are certain issues with them at the same time: it is hard to prove generalizability, given that style can vary quite a bit depending on the area of focus of the fake news. These models are inherently limited in this way, and are likely unsuitable for machine-generated fake news detection. This is because LLMs do not have a notion of "real" and "fake" news: they simply generate text. This means that they do not expose "tells" that they are producing untruthful text, unlike humans, as the LLM itself does not know the difference. Furthermore, an LLM is likely to be very effective at mimicking the style of actual true news content, and thus would not necessarily exhibit the characteristics of fake text that these approaches use for discriminating fake text from real text [206].

Social-response-based Approaches

An alternative approach that is fairly common in the literature is to amass user responses to a certain event or news item, and feed these in aggregate to various ML algorithms. User responses are typically composed of short texts that convey the user's stance on the news item. RNNs are a common model to employ for such approaches, due to their ability to model changes over time. The user responses are typically fed to the RNN in sequence, with the hidden vector passed to a final fully connected layer for the binary prediction. Early work used TF-IDF values of a limited vocabulary `k` as input to the RNN [168], and was shown to be effective. Further research applied attention layers to enhance interpretability and allow the model to attend to specific phrases more effectively. Both approaches in essense mine the responses of users for terms like "fake", "untrue", etc., taking advantage of the ability for users to recognize fake content on their own [168] [51]. Furthermore, expressions with higher emotionality (e.g. "unbelievable!") in the responses of users were shown to correlate with fake news, likely owing to the fact that fake news is written in a way to purposefully evoke an emotional response. The presence of emotional terms in the responses of users was shown to correlate more closely with fake news than references to the news item content itself [51]. Certain political terms as well seem to correlate with fake news (e.g. "PC shit" [sic], where "PC" is short for "political correctness") [51]. It is likely that metadata relating to geographic location and IP addresses could also be important features in building a classifier to detect fake news in practice; not much academic research exists on using this metadata as features,

as it is data that is not generally publicly available. It is very plausible that e.g. users in certain countries would be more involved in generating fake news content than others; it is generally well known that many misinformation campaigns arise from IP addresses associated with specific non-Western countries, be it individual hacker groups or even government-orchestrated campaigns.

Hybrid Approaches

Several approaches seek to take the social response from users and augment it with certain user characteristics. CSI (Capture, Score, Integrate) [197] represents the first attempt to unify the social response with characteristics of the users who are responding to a given news item. CSI is composed of three modules: the capture module, the score module, and the integrate module. The capture module is the same as other social-response-based approaches, and is simply an RNN that is fed the user responses to a given news item in a sequential fashion (encoding the text using doc2vec). The score module creates a global user vector per user via an incidence matrix representation of the news items they've interacted with, as well as a corresponding suspiciousness score. The suspiciousness score is the combination of two components: a vector representation of each user based on a set of user features passed through a single connected layer, and the result of applying single value decomposition (SVD) to the adjacency matrix representing the entire social network of users. The idea is that suspicious users likely interact more with other suspicious users. The score module then sums up the suspiciousness scores of the users who've interacted with a given news item to create a news-item-level score. This score is then concatenated by the integrate module to the capture module's output for the final prediction. The user vector can hypothetically be used for other analysis tasks, in combination with the global per-user suspiciousness score. The issue with this approach is that the score module is quite complex, having to analyse all of the news items every user has interacted with to generate the global suspiciousness score. Furthermore, it assumes that users have a steady "suspiciousness" over time, and loses the nuance that some users might sometimes share fake news, but not always.

Another approach seeking to instead integrate the content with the user responses is known as dEFEND [211]. This approach uses a *hierarchical attention neural network* [250] to examine the content of the news item at both a word and sentence level. A co-attention layer seeks to quantify the correlation between sentences in the news item content and specific user responses, to provide explainability through visualization of the strongest correlations between news content sentences and user comments. These most important sentence-comment pairs allows the model to isolate the exact sections of a news item that are fake (with the understanding that in fake news it is not the entire news item that is fake, but only certain parts of it interspersed with truth), and the user commentary that relates to those specific parts. This approach represents a large step in explainable fake news detection.

FNED [164] is a SoTA approach that builds off of the CSI approach to simplify it and focus on early detection specifically. It aims to fix the assumption of CSI that users can be boiled down to a single suspiciousness score that does not vary over time. Instead of

generating such a score taking into account all interactions of all users in the dataset, FNED uses a convolutional layer to extract textual features from the response, then concatenates this with the result of passing the user features through an embedding block. This concatenated vector is called the "status-sensitive crowd response feature", a direct merger of the response text itself and the user information, and is fed to a CNN to perform the actual discriminative task. The CNN is augmented with a "position-aware attention layer", a regular CNN attention layer that is augmented with the position of the response within the response series, with the assumption that how early a response is matters in terms of detection. The approach is demonstrated to be extremely effective, especially in an early detection context.

Graph-based Approaches

Other approaches seek to model the differences between fake and real news in terms of the propagation patterns through the network as news items are shared, and the relationships between various entities in the social network graph. The hypothesis is that fake news propagate substantially differently through the network than real news. One approach, known as TraceMiner [244], uses RNNs in combination with graph embedding algorithms to model the social graph in a way that can be fed to the RNN. The embedding algorithms used are SocDim, LINE, and DeepWalk ([230] [229] [184]). These embeddings produce user vectors that represent the structure of the network surrounding a given user. This approach consists of first learning the user embeddings via one of the aforementioned algorithms, then feeding the embeddings of the users who responded to a given news item to the RNN in sequence, similar to how the content of the responses are fed to RNNs in the social-response-based approaches. This approach has been demonstrated to be competitive with social-response-based approaches, which are more prone to manipulation by malicious spreaders [244]. This approach also is quite effective in early detection scenarios, or scenarios where response information is quite limited (on most platforms, users have the ability to simply share the story without commentary).

Another successful approach is known as FANG [180]. FANG uses GraphSAGE ([105]) to create user embeddings to feed into a hybrid BiDirectional-LSTM Attention-based model. RoBERTa is used to detect each user's stance to a particular news item they interacted with (stance detection) based on their response text. The embeddings generated by GraphSAGE take into account the users (characterised by their profile text, which has been shown to be a reliable indicator of a propensity to share fake news [180]), the news item content (a fusion of GloVe embeddings and TF-IDF scores), the character of the publishers (represented by TF-IDF scores of their About Us page and Homepages), and finally the relationships between all these entities, including which users follow which other users. Fed into the Bi-LSTM are the GraphSAGE representation of the user, their stance towards the given news item, and the time elapsed since the news item's publication. To create a final prediction, the GraphSAGE embedding for the news source and the Bi-LSTM output are concatenated together and fed to a fully-connected layer. The approach has been demonstrated to be highly effective compared to other graph-aware approaches. Furthermore, the Attention layer indicates that for fake news, the model

attends to responses mostly within only 12h after publication, while for real news, the attention is spread out much more evenly over a two week period. This intuitively makes sense, as fake news is designed to cause immediate outrage, and thus propagates quite quickly, while real news circulates and remains relevant for longer spans of time.

Other approaches focus on injecting more information into the graph embeddings. The ideological slant of the publisher of a given news item on a left-right spectrum has been shown to improve the accuracy of graph-based fake news detection approaches [213], when modelled jointly with user interactions and a user adjacency matrix. Other approaches, such as GCAN (Graph-aware Co-Attention Network) [165], seek to improve interpretability of DL approaches for fake news detection. GCAN integrates several different feature extractors together into one network: an RNN (GRU) layer that encodes the source tweet (the news headline), the propagation pattern as an input to a convolutional layer, the propagation pattern as an input to a GRU layer, and finally, a Graph Convolutional Network (GCN) layer to generate an embedding for a given source tweet by modelling the neighborhoods of users who interacted with the tweet in a constructed graph. The propagation pattern inputs to the GRU and convolutional layers are simply a set of hand-crafted user features (number of followers, length of description, etc.) fed in sequence of interaction with the original tweet to the layers. GCAN was demonstrated to improve the SoTA scores significantly, and does not rely on the actual replies of the users (who frequently repost without adding commentary), but rather simply the users themselves and the order of interaction. GCAN also integrates a "dual co-attention" mechanism to learn attention maps between both the graph-aware (GCN) representation of the interactions and the source tweet encoding, and the source tweet encoding and convolutional propagation encoding. This confirms the results of previous research ([51] [168]) that use of certain emotionally charged words in the source tweet correlates with fake news, as well as certain user characteristics like short account descriptions and recent account creation time. The latter implies malicious actors use bots and/or create fake accounts specifically for the promotion of fake news (and thus these accounts are young).

Multimodal Approaches: Incorporating Visual Information

In recent years, multimodal approaches that aim to incorporate information from multiple modalities into the predictive model have gained more and more popularity. Most multimodal approaches concern the fusion of visual and textual information. It has been demonstrated that the use of images related to real events present far more variability in terms of content than those related to fake news [125]. The use of hand-crafted visual features to quantify the distinctiveness of images has been demonstrate to boost baseline accuracy by up to 7% [125]. However, it is very time-consuming to craft features by hand, and so other, more recent approaches seek to incorporate visual information using existing feature extractor convolutional networks like VGG. Early research incorporating visual information via a VGG component in conjunction with word embeddings for actual post content and hand-crafted social (user characteristics) features demonstrated a marked improvement compared to approaches without the visual features [124].

One markedly different approach is known as EANN (Event Adversarial Neural Net-

works) [238]. This approach utilizes a GAN-like training process to create a model which is hypothesized to be more generalizable compared to existing approaches. Based on a hypothesis that existing approaches rely on characteristics specific to the events present in the dataset they are trained on (are ineffective at generalizing), EANN is trained to extract event-agnostic features with which to perform classification. The architecture is similar to other approaches which incorporate VGG and textual features, however, the training is modelled as a minimax game between the feature extractor (textual + VGG) and an event discriminator. The event discriminator is trained to determine which event is being fed to the feature extractor from the concatenated visual and textual representation the feature extractor emits. In a similar fashion to GAN training, the feature extractor is forced to learn a representation which betrays as little event-specific information as possible, to fool the discriminator. This approach was demonstrated to exhibit a marked improvement in accuracy compared to the SoTA of the time.

Another relatively different approach is known as MVAE (Multimodal Variational Autoencoder) [139]. MVAE uses a bimodal variational autoencoder to incorporate both text and image information into the predictive process. Under the hypothesis that existing approaches do not find correlations across modalities, MVAE was designed to incorporate a single shared representation over both textual and visual modalities for the reconstructive process. Bidirectional LSTMs are used to generate the textual component of the shared representation, while VGG-19 is used for the visual feature extractor. These representations are concatenated together to create the shared latent vector, which is then used to separately reconstruct both the original text and the VGG-19 extracted features. This approach was demonstrated to significantly outperform other multimodal approaches like att-RNN ([124]) and EANN ([238]), while not utilizing any propagation information at all, and operating only using the content of the news item itself (the orignal text and associated image(s)). This makes MVAE an especially good candidate for early detection.

Potential Research Direction # 2 (Fake News Detection)

There is no existing research covering multimodal detectors that incorporate explicit graph representations (not just sequences of user responses). It is conceivable that a model that incorporates VGG-based image features, content text, user response text, *and* relationships between users (via DeepWalk or other similar graph embeddings) would be even more effective at detecting fake news than the approaches outlined here. Furthermore, most approaches that utilise profile features typically hand-craft them based on what the authors believe may correlate with users who propagate fake news. However, a potential new direction may be to attempt to create profile-based "user embeddings" (in contrast to FANG's GraphSAGE-based user embeddings) via passing user information (e.g. recent tweets) through a language model like Sentence-BERT. The intuition behind this would be that a user is likely best defined through their most recent tweets, given that opinions (and by extension the propensity for a given user to spread misinformation) change over time.

4 Topic and Sentiment Modelling for Social Media

4.1 Introduction

This chapter presents an overview of topic and sentiment analysis approaches, as applied to social media posts (such as on Facebook or Twitter). We outline certain challenges relating to sentiment analysis as a whole that cause it to be a somewhat challenging problem, as well as challenges specific to social media platforms that make both topic and sentiment analysis more challenging than in the usual cases. An overview of classical topic modelling approaches, including Latent Dirichlet Allocation (LDA), as well as newer, more modern approaches for topic modelling that incorporate deep learning is provided. Various sentiment analysis methods are also discussed, split into two major categories: unsupervised rule-based methods, involving the creation of hand-crafted rules for modelling sentiment, and the current SoTA in semi-supervised (transfer-learning based) approaches, which typically leverage massive pre-trained language models like BERT, RoBERTa, and various other Transformer-based architectures. These approaches involve unsupervised pre-training on massive amounts of unlabelled data (or, more typically, simply using an existing publicly available pre-trained model), followed by few-shot prediction after fine-tuning on a small amount of labelled examples. Finally, a brief overview of two new and difficult dimensions of sentiment analysis is given: multimodal sentiment analysis, which pertains to the analysis of the sentiment of multiple modalities at once (e.g. image, audio, and video data, especially relevant to social media posts, which frequently include images and video along with text), and target-based sentiment analysis, which involves detecting both the sentiment and the target of said sentiment in a piece of text. Methods for analyzing sentiment over time are briefly discussed.

4.2 Introduction to Topic Modelling

The goal of topic modelling is to discover a set of "topics" (abstract concepts/categories) that exist in a document corpus. Specifically, the goal of a topic model is to a) discover the existing topics in a corpus, typically modelled as latent variables, and b) to assign each document some specific mixture of topics, generally modelled as a proportion in relation to each of the discovered topics (X% topic A, Y% topic B, and so on and so forth). Similar to most clustering algorithms, where the user must pick the number of clusters before applying the algorithm, with most topic models the user must set a specific number of topics to be discovered.

4.3 Overview of Classical Approaches to Topic Modelling

LDA

Latent Dirichlet Allocation (LDA) is a common generative approach to topic modelling that involves unsupervised discovery of latent topic variables based on a corpus of documents. Specifically, each topic is defined by a collection of words that represent it. Each word belonging to each topic has an associated probability (posterior probability), which

represents the likelihood of that word being selected in the generative process given the topic's presence in a document. For the LDA model to be utilized, the posterior distribution first needs to be learned from the training corpus. The number of topics to be discovered is preset at the beginning of the process, similar to the selection of a number of clusters for an unsupervised clustering algorithm.

The generative process involves several steps:

1. For each document generated, a topic distribution is drawn based on the Dirichlet distribution (parametrized by vector α)

2. For each word in the document to be generated (excluding stop words and common words), a topic is sampled from the topic distribution

3. Finally, a word is sampled from the probability distribution of words belonging to the previously sampled topic

The problem with LDA is that the true posterior distribution is intractable, and thus sampling methods are used (collapsed Gibbs sampling, Monte Carlo EM), but these are are computationally very expensive. This makes it difficult to perform interactive topic discovery, as each change to the modelling assumptions or even changes to the document corpus itself require a costly recalculation of the posterior distribution.

Another challenge with topic modelling is the difficulty in evaluating the generated topics. Most papers use a variety of different methods for evaluation: unfortunately, there is no "standardized" metric, nor a single standardized dataset for evaluating topic models. Human evaluation is effective but costly. One way that is relatively frequently used is normalized (pointwise) mutual information (NPMI). NPMI essentially models the "coherence" of topics, typically averaged across all topic words. A higher average NPMI means that the words within the topic category co-occur at a higher rate, and thus generally indicates high topic coherence. NPMI has been demonstrated to be correlated with human evaluation of topic quality [6].

4.4 Neural Topic Modelling

Variational Topic Modelling

Neural topic models (NTMs) aim to perform unsupervised topic discovery from a corpus of documents using neural networks. One approach that aims to combine LDA and the autoencoding variational Bayes (AEVB) [140] architecture is known as Autoencoded Variational Inference For Topic Model (AVITM) [220]. AEVB avoids the expensive computation of traditional methods for learning the posterior distribution by training a neural encoder network to learn the distribution directly. AVITM tackles two issues regarding the use of AEVB with a Dirichlet prior: a) the common problem in the AEVB of component collapse, where the network gets stuck in a bad local optimum, and b) the issue of reparameterizing the LDA prior (which is necessary to use the AEVB approach), which is challenging because the Dirichlet family of distributions are not a location-scale family.

AVITM solves the issue of component collapse by using a combination of the Adam optimizer with a high learning rate and high momentum, batch normalization, and dropout units to avoid getting stuck in local optima. The reparameterization of the Dirichlet prior is solved by approximating it via a logistic-normal distribution. The result is a black box inference method that learns the prior distributions (via the variational parameters) necessary for LDA via neural network (the "inference network"), accepting as input the documents of the corpus (the "observed data") for training. The "autoencoding" nature of AVITM arises from the fact that the inference network is composed of two parts: an encoder and a decoder, where the encoder takes a document and produces a continuous latent representation for it, and the decoder takes the latent representation and attempts to reconstruct the original words from the BoW representation of the document (essentially performing the LDA generative process). The proposed approach is far faster than traditional sampling approaches, and is feasible to run on a one million document dataset in under 80 minutes on one GPU. Furthermore, AVITM provides much faster inference on new data compared to other approaches (like standard mean field), as it is simply a matter of passing the new data point through the neural network. Furthermore, in the same paper as AVITM an approach called ProdLDA is proposed, which produces better and more coherent topics than the standard LDA approach, by replacing the word-level mixture model of LDA with a weighted product of experts model.

A similar approach to AVITM is known a the Neural Variational Document Model (NVDM) [174]. The primary differences with AVITM is that instead of reparameterizing the Dirichlet prior, a latent Gaussian distribution for the topics is assumed instead, which allows for more straightforward reparameterization, but carries a certain bias. As well, the high momentum Adam training is not used to avoid component collapse.

LDA2Vec

LDA2Vec [177] represents a different approach to NTMs, seeking to augment the Word2Vec model by adding a learned document representation to Word2Vec's Skipgram Negative-Sampling (SGNS) objective, in addition to the context-independent word vectors that are learned. These document vectors are the product of a document weight vector, which represents the proportion of each topic present in a given document, and the topic matrix, which represents topics as points in the word embedding space (though they do not necessarily correspond to actual words). The fact that both topics and words exist within the same embedding space allows topics to be easily visualized and understood via the words in their neighborhood in the embedding space. The existing SGNS objective is not modified: the pre-training task is simply predicting for a given pair of words whether one follows the other. The idea of the modified SGNS objective of LDA2Vec is that document information can help in the pair classification task, and therefore result in more robust word vectors. In regular Word2Vec, summing the vector for "Germany" and the vector for "airline" may result in a similar vector for "Lufthansa". In LDA2Vec, a document about airlines would have a similar document vector to the word vector for "airline". As the negative sampling loss is calculated in relation to a "context vector", which is the sum of the word vector and the document vector for the current document, the classification

of the word pair is rendered more effective, with this additional document-level information (see Figure 14 for a visual representation). These three representations (word, document, topic) are jointly discovered during the pre-training process via the modified SGNS objective.

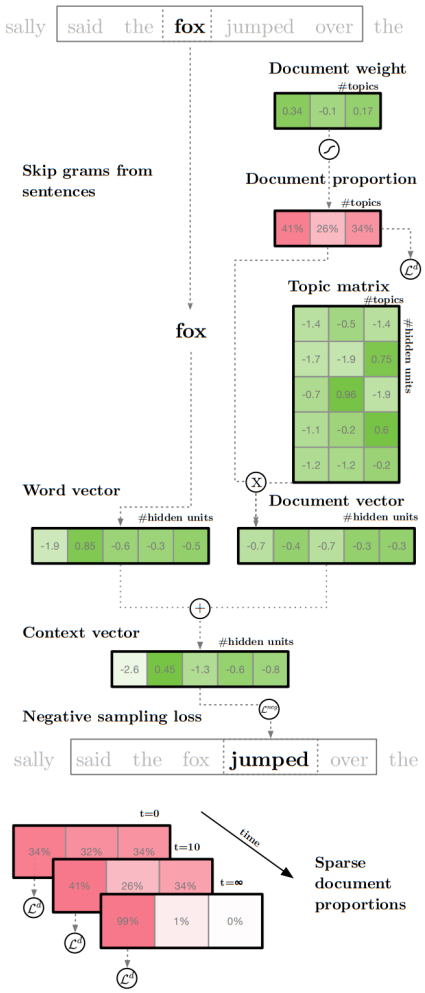

Figure 14: The pre-training process for LDA2Vec. The addition of the document vector and its contribution to the pre-training task (the negative sampling loss) is clear. [177]

Top2Vec

Top2Vec [8] is a similar approach to LDA2Vec, that however differs in a few key ways. Both LDA2Vec and Top2Vec seek to model documents as vectors within a shared word/document embedding space. LDA2Vec jointly generates interpretable document topic-mixture representations (a vector with the proportions of each topic that compose the document) via the modified pre-training objective in addition to the final document vector in the word embedding space. In contrast, Top2Vec opts for the comparatively simpler and

more straightforward approach of performing clustering directly on the word/document embedding space via the HDBSCAN algorithm after applying dimensionality reduction. The word/document embeddings can be generated by any algorithm that produces such joint embeddings, as long as both documents and words are jointly embedded in a shared semantic embedding space. Some examples are Doc2Vec, Universal Sentence Encoder, and BERT Sentence Transformer, the implication being that the quality of these shared embeddings will determine the performance of the approach to a large extent. The centroid of each cluster in the shared embedding space as produced by HDBSCAN is considered to be the "topic vector", and the 5 nearest word vectors to the topic vectors are taken as the topic descriptors. The distance from each topic centroid to each document vector in the embedding space could hypothetically be used to produce a similar topic-mixture representation as LDA2Vec, however this is not a key part of the algorithm in the same way as it is in LDA2Vec.

Use of Pre-trained Embeddings for Neural Topic Modelling

Recently the use of pre-trained embeddings leading to SoTA results for many NLP tasks has prompted researchers to begin to look for ways to incorporate them into unsupervised topic modelling. One approach to doing so is to extend the AVITM approach by augmenting the traditional BoW representation with contextualized Sentence-BERT embeddings [26]. The sentence embeddings of the sentences in the document are concatenated to the original BoW representation of the document before passing the result of the concatenation to the autoencoder, allowing the autoencoder to use the BERT embeddings to aid in the reconstruction of the BoW representation of the document. The BERT embeddings are first passed through a single hidden layer prior to concatenation. Higher topic coherence is demonstrated compared to the original ProdLDA, as well as both NVDM and traditional LDA, indicating that the injection of external information is beneficial to the topic discovery process.

Another approach to topic modelling using pre-trained embeddings eschews the use of probabilistic models entirely, instead clustering vectors in the embedding space directly from pre-trained embeddings like BERT, GPT-2 and RoBERTa, with the intuition that proximity in the embedding space between two word embedding vectors suggests a common theme [231]. Specifically, k-means can be used to produce word clusters (topics) that resemble the prior distribution produced by LDA. The use of contextualized BERT embeddings allows the clustering approach to better account for polysemy (different meanings of the same word) compared to the topics formed by traditional LDA, which does not have access to contextual information (as it uses a BoW representation). The contextualized embeddings also capture more varied parts-of-speech to represent each topic, unlike LDA which generally prefers nouns. By using contextualized embedding clustering, analyses over partitions of the corpus can be performed, for example estimating the prevalence of certain topics over time, in the case of a corpus with temporal information associated with documents. Furthermore, it is demonstrated that the clustering in the embedding space is meaningful to produce topics even when PCA dimensionality reduction is applied to reduce the 768-dimension BERT embeddings down to 100 dimensions.

A somewhat different approach is improving existing neural topic models with pre-trained transformer networks via the process of knowledge distillation [115]. This approach is named "BERT-based Autoencoder as Teacher" (BAT), and involves fine-tuning a pre-trained BERT model with a document reconstruction objective, to act as an autoencoder. The BAT representation of a document, unlike the BoW representation, incorporates related but unseen terms into the representation distribution (vector), informed by BERT's pre-trained knowledge (see Figure 15). The BAT representation is incorporated as a KD term in the loss function for the NTM, which guides it to mimic the BAT representation. This allows the NTM to incorporate the knowledge contained within the the pre-trained BERT embeddings while maintaining the interpretability inherent to the NTM approach.

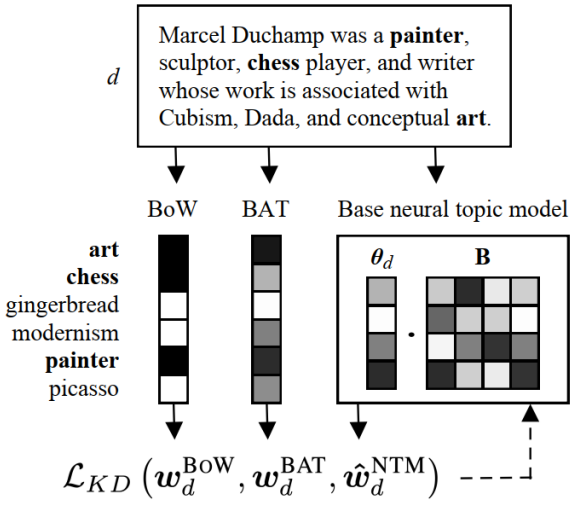

Figure 15: An illustration of the BoW representation of a document alongside the BAT representation. Weight is given to related terms that are not actually present in the document, improving the topic model. [115]

Neural Topic Modelling for Social Media

Social media creates additional challenges compared to corpora with longer documents when it comes to the application of traditional LDA approaches. LDA produces topics based on the co-occurence of topic words within documents. However, short documents (social media posts) create much sparser co-occurence statistics, making topic discovery more challenging.

One approach that uses BERT pre-trained embeddings for topic classification (not discovery) for social media posts specifically is known as TopicBERT [47]. TopicBERT leverages the NVDM architecture for document classification, rather than discovery. TopicBERT concatenates the NVDM latent representation with the BERT word embeddings to perform more effective document classification, and achieves higher scores in several

benchmark document classification datasets (Reuters8, 20 Newsgroups, and IMDB), compared to a set of baseline architectures (plain CNN, BERT, DistilBERT). This demonstrates that the NVDM latent representation can provide additional information to boost document classification accuracy in a social media context.

Another approach that more closely aligns with traditional LDA is that of the Latent Concept Topic Model (LCTM) [117]. Unlike LDA, which models each topic as a distribution over words in the document corpus, the LCTM instead seeks to model each topic as a distribution of latent concepts. These concepts are modelled as Gaussian distributions in the pre-trained word embedding space (in the LCTM case, GloVe vectors). The use of latent concepts (modelled as "concept vectors" in the word embedding space) instead of using words directly helps to ameliorate the sparsity problem for short texts, as the number of distinct concepts in a corpus is far fewer than the number of distinct words. The co-occurence statistics of latent concepts are therefore much less sparse. A collapsed Gibbs sampler approach is used for inference. The LCTM is also able to take advantage of OOV terms during inference on new data, by mapping them to the existing latent concepts via the embedding space. The LCTM has been demonstrated to be far more effective than traditional approaches at modelling topics in short texts (e.g. on the 20 Newsgroups dataset).

A similar approach to LCTM is known as GPU-DMM [158]. GPU-DMM uses a Dirichlet Multinomial Mixture (DMM) architecture augmented with pre-trained Word2Vec embeddings to group words with similar meanings in a similar way to LCTM. DMM is a similar approach to LDA, but instead assumes that a document is generated from a single topic, rather than a mixture of topics. For short texts, this is generally a fair assumption to make. The second component of GPU-DMM is the Generalized Polya Urn Model (GPU). The GPU allows the DMM model to incorporate semantically related words into the sampling process (which is similar to collapsed Gibbs sampling), where these semantically related words are based on proximity in the word embedding space. This allows it to achieve better results on short texts compared to traditional LDA, by in effect increasing the size of the texts being learned from.

The Context Reinforced Neural Topic Model (CRNTM) approach modifies the NVDM architecture to be more effective for short texts [76]. It does this by modelling topics as Gaussian mixture distributions within the pre-trained word embedding space, to alleviate the issue of feature sparsity with short texts. In a similar way as LCTM, CRNTM seeks to enrich the contextual information in short texts via nearby words in the embedding space. Unlike LCTM, this approach is used in combination with a more modern VAE-NTM approach, rather than a traditional sampling process.

Potential Research Direction # 3 (Extending NTMs)

Generally speaking not as much research has been done on topic modelling for short texts, ergo this seems like the area with the most potential for novel research. Bidirectional transformer representations (e.g. BERT) have been used to perform topic modelling via clustering the learned document representations (e.g. [115]). However, the potential for large autoregressive models to perform one-shot or zero-shot topic modelling has been

relatively unexplored. One novel line of research could be using GPT-3 and in-prompt learning to perform topic modelling on a short text corpus without gradient descent. One limitation would be that a representative sample of the corpus would have to fit into the prompt-length of the model, however there is much potential for the use of proper prompt tuning (a newly emergent field relating to optimizing in-prompt learning for large autoregressive models by carefully selecting prompts) to aid in effective topic modeling.

4.5 Sentiment Analysis

Sentiment analysis is not a single problem with well defined boundaries, but rather a family of problems relating to the detection of sentiment (emotion/attitude) in text. This family of problems can be roughly split into three categories:

1. Document-based, which relates to the detection of the prevailing sentiment in an entire document (a series of sentences),
2. Sentence-based, which relates to the detection of sentiment in a single sentence,
3. and Aspect-based:
    (a) Simple aspect-based, where the goal is to detect sentiment relating to a certain "aspect" of the topic of the text (e.g. in a product review context, sentiment regarding price, quality, etc.)
    (b) Target aspect-based, where the goal is to detect both the aspect, and the target of the sentiment as well

Of these three cases, the most challenging and least-developed is aspect-based sentiment detection, which also involves the most moving parts, as both sentiment has to be detected, and potential targets of the sentiment have to be determined. These two subproblems are known as Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). AE is a problem of sequence labelling, while ASC is a regular classification problem. These fundamentally different subproblems make it difficult to create a single uniform architeture that can perform both tasks.

Sentiment analysis can furthermore be split into supervised and unsupervised cases. The supervised case can be modelled in several different ways. One of the most simple ways to model the problem is to use a simple polarity axis, to predict how positive or negative the sentiment is. More complex modelling strategies include trying to predict a specific emotion in a set of standard emotions (anger, sadness, disgust, etc.). The former can be set up as a standard binary classification problem, the latter as a multi-class classification problem where each emotion is a distinct class. When trying to model complex emotions as a multi-class problem, the Plutchik wheel of emotions is typically used (for more detail see Chapter 2).

Sentiment Analysis and Stance Detection Standardized Datasets

In terms of standardized datasets for sentiment analysis, one common source is the International Workshop on Semantic Evaluation (commonly known as SemEval) [10]. The datasets of SemEval cover a wide range of NLP problems, including sentiment analysis, as well as stance detection. Stance detection refers to a related problem to sentiment detection, which instead has as a goal to detect the user's stance with regards to a certain issue/topic (in favor, against, neutral). Stance detection is most similar to target-based sentiment detection in this way, in that a target of the emotion/stance is needed. SemEval 2016 specifically offered a dataset for stance detection in a set of 2900 tweets, where the goal is to detect the stance of a particular tweet with regards to 5 distinct topics ("Atheism", "Climate Change is a Real Concern", "Feminist Movement", "Hillary Clinton", and "Legalization of Abortion"), and 3 stances ("in favour", "against", and "none"). This dataset, despite having been used in several papers, is currently out of date, and the size of the dataset is not sufficient for more complex newer ML systems. A better, more recent option is the stance detection dataset of the Fake News Challenge [11], which consists of 49k rows. The dataset consists of headlines and body text from online articles. The goal is, for a given headline and body text (not necessarily from the same article), to classify the pair into one of 4 classes: "agrees" (the body text agrees with the headline topic), "disagrees" (the body text disagrees with the headline topic), "discusses" (the body text discusses but does not express opinion on the headline topic), and "unrelated" (the two are about unrelated topics).

Traditional Supervised Sentiment Analysis

The supervised cases (excluding aspect-based detection) are relatively straightforward. BERT, RoBERTa, and other similar contextualized embeddings that have been pre-trained on massive amounts of unlabelled data provide a strong foundation for any supervised sentiment detection approach, through the use of transfer learning. Pre-trained weights for these architectures are available online. By using these pre-trained weights, these architectures can achieve SoTA accuracies via fine-tuning on fairly small labelled datasets, by leveraging their acquired knowledge through the pre-training process. At the time of writing, the current SoTA pre-trained embedding architecture is known as SMART-RoBERTa [123], and consists of a pre-trained RoBERTa model trained via a new learning framework that provides better generalizability to new data when fine tuning on small labelled datasets. A library known as HuggingFace [12] provides an easy-to-use high-level API for fine-tuning large Transformer-based NLP models for downstream tasks.

GPT-3 has been used for few-shot sentiment classification, leveraging the in-context training abilities of the model (mentioned in Section 3.4). It has been demonstrated that the choice of in-context examples is a crucial factor in how effective GPT-3 will be for a given few-shot learning task, with highly variable performance depending on this

[10] https://semeval.github.io/
[11] http://fakenewschallenge.org
[12] https://huggingface.co/

choice [161] [261]. Specifically, it has been demonstrated that the more similar the in-context examples given are to the testing set used, the higher the effectiveness of the model, which intuitively makes sense. The KATE system [161] uses a KNN-based sampling method for in-context examples, based on sentence embeddings produced by other models (e.g. RoBERTa), and demonstrates a consistent increase in accuracy compared to randomly selected in-context examples from the training set (which is the approach used for constructing in-context examples in the original OpenAI GPT-3 paper). KATE was evaluated on a sentiment analysis task (SST-2 [216]) and demonstrated competitive accuracy, despite being trained in a few-shot context. Models like SMART-RoBERTa (current SoTA at the time of writing) achieve higher accuracy, but are fine-tuned with gradient updates and far larger training sets than can fit in GPT-3's model prompt for in-context (few-shot) training.

A popular example of a rule-based approach for entirely unsupervised sentiment analysis of text is known as VADER [119]. VADER consists of hand-tuned word-sentiment correspondences (including sentiment intensity), in combination with rules that model how the presence of certain syntactical features modify the intensities (features like negation, degree modifiers, etc.). VADER is specifically tuned to the type of language used in social media contexts, and is aware of things like capitalization and punctuation, and how they modify sentiment intensity. At the time of publication of the VADER paper, VADER outperformed all ML sentiment analysis approaches, as well as human raters in certain cases. Most other fully unsupervised approaches use a variation of this lexicon-based strategy augmented with rules. One set of datasets that is often used to underpin such approaches is the NRC emotion and sentiment lexicons [176].

Multimodal Sentiment Analysis

Multimodal sentiment analysis refers to the use of multiple modalities for input for a sentiment (typically) classification model. There are two different ways that multiple modalities can be used:

1. Classifying sentiment based on not only the textual content of the post on social media, but also any images that have been uploaded with it, typically relating to the content which the post discusses

2. Classifying sentiment based on multiple directly correlated modalities, typically speech utterances: given a source video of someone speaking, the audio of their voice, and a transcript of what they're saying, classifying their sentiment.

Standard datasets for multimodal sentiment analysis primarily use the second problem structure, rather than the first. Standard datasets used include the Multimodal Opinion Utterances Dataset (MOUD) [189] and the Multimodal Corpus of Sentiment Intensity (MOSI) [255]. The first problem in the list above can be considered an area of open research, although it is also potentially a simpler problem, as correlations across modalities likely play a much smaller role than with utterance datasets. Nevertheless, it would be interesting to adapt one of the below models for the first task.

One approach to the second problem is known as MISA [108]. MISA uses separate feature extractors for each modality (BERT for text, bidirectional LSTM for image and speech) and projects each set of features per modality onto a distinct modality-specific subspace, as well as a single modality-invariant subspace. MISA is designed for synchronized multimodal classification: classifying video, speech, and text that all relate to the same context (a person speaking, where the video is of them speaking, the speech is the audio of them speaking, and the text is a subtitle). A Transformer model (with Multi-head attention) is used to fuse the different modalities together in the final stage to make a prediction.

Another new approach for the same speech-utterance task is known as TransModality [239]. TransModality extends the Transformer architecture to include "modality-fusion cells", which jointly learns the correlations across modalities by "translating" from one modality to the other, leveraging the Transformer's original design for textual translation. The TransModality model achieves SoTA results on the two standard datasets mentioned earlier.

Potential Research Direction # 4 (Textual Sentiment Analysis over Time)

There is very little existing research examining the problem of analysing sentiment over time in a textual corpus. One can envision two different approaches: one approach would be to leverage the transfer learning strategies outlined earlier, using a pre-trained model like RoBERTa to segment the dataset into set time periods, and then performing few-shot classification on each period to analyse sentiment in a corpus or dataset over time. A different approach might be to use time series based algorithms to attempt to capture the sentiment over time in a single model. The aforementioned speech utterance task does include a temporal component, as the dataset is composed of videos with per-frame labels of sentiment, so the models mentioned in the previous section can also be thought of as doing sentiment analysis over time, for the duration of the video. Analysis of entire textual corpora over time is a comparatively less explored research area.

Aspect-based Sentiment Analysis

Targeted aspect-based sentiment analysis (TABSA) is an especially challenging subproblem of sentiment analysis, composed of the separate AE and ASC subproblems mentioned earlier. One approach to ASC involves the use of an auxiliary sentence to reframe the problem as a sentence-pair classification task (similar to question answering) [223]. When compared to directly fine-tuning BERT on the task, the construction of an auxiliary sentence presents a substantial improvement to accuracy on the standard SentiHood dataset. Directly using BERT on the SentiHood task constitutes fine-tuning a separate BERT model for each aspect. The auxiliary sentence that is constructed takes the form of a question or statement relating one of the aspects and targets together (e.g. "what do you think of the safety of location x"). It is hypothesized that the increase in accuracy compared to applying BERT directly to the task is due to the pre-training strategy of BERT, specifically the next sentence prediction component, that give BERT an edge when

it comes to sentence-pair-based tasks.

A different approach to performing TABSA, instead of modelling the problem as a sentence-pair classification task, is to use conditional random fields (CRFs) in conjunction with the direct application of BERT without fine-tuning via layer aggregation [138]. The CRFs inject aspect information (performing sequence labelling) into the training process which aids the network in performing the sentiment analysis. The last four layers of the BERT network are used for the aggregation, under the assumption that sentiment classification is a fairly high-level task, and thus the information required would be contained in these later layers. This is consistent with typical use in existing literature using BERT as simple embeddings for downstream tasks, where the output of the last 4 layers of BERT is used as input to regular linear layers trained on top of them, without gradient updates (i.e. fine-tuning) to the BERT layers themselves.

The success of BERT with regards to ABSA has been analyzed through the visualization of the fine-tuned BERT model self-attention heads, to see what features are being used in this classification task, and to interpret the latent space generally [247]. Through this analysis, it has been shown that there is no single dimension or small set of dimensions responsible for the success of BERT when it comes to ABSA tasks. It has been demonstrated that the majority of the dimensions of BERT when fine-tuning on ABSA relate to the semantics of the aspect itself, rather than the opinions learned from the fine-tuning dataset. Nevertheless, a deeper understanding of why BERT is so effective at ABSA (and NLP tasks in general) remains somewhat elusive.

ASBA in a Unified Framework

Most existing research in ABSA is focused on the classification subtask, with the assumption that targets have already been extracted. Despite this, there is some research that aims to focus on both subtasks simultaneously. GRACE (GRadient hArmonized and CascadEd labeling model) [167] is an end-to-end framework for performing both AE and ASC. It does so through a "aspect term-polarity co-extraction" framework, essentially performing both the AE and ASC task simultaneously. That is to say, provide two labels per token in a given sentence: one indicating if it is an aspect or not, and the second to provide the sentiment if it is. It seeks to solve the issues of a) class imbalance in the AE training set, in other words that the majority of the words in any given text are not aspects, and b) interplay between aspect terms in a given sentence, e.g. by coordinating conjunctions ("nice `a` and `b`" should label both `a` and `b` as positive). The model uses a cascaded labelling approach, where the aspect labels are first generated from the pre-trained BERT embeddings, then are fed into a Transformer-Decoder block as the key and value parameters to generate the sentiment labels. The use of the Transformer-Decoder block (built on Multi-Head Attention) mitigates the issue of sentiment not propagating properly through coordinating conjunctions. The issue of class imbalance in the AE task is mitigated by using a gradient harmonized loss, borrowed from the problem space of object detection.

Another approach that aims to perform both AE and ASC in a unified framework is known as SpanABSA [116]. Most existing solutions to joint AE/ASC use a sequence

tagging approach, where each word is given a term/not term label, as well as a positive/negative sentiment label (as mentioned previously). SpanABSA uses a different representation instead, where the target span is represented directly as a span (a start and end point in the sequence), rather than as a stream of tokens. As with GRACE, BERT is used as the DNN underpinning the framework. For the AE component, BERT is used to predict the start and ending positions (the span) directly, rather than predicting a label for each token in the input (as with most other sequence labelling approaches to AE). A heuristic approach is used to narrow down the suggested target spans based on several criteria, e.g. removing overlapping spans or softmax scores for the start and end positions below a certain threshold. These spans are then fed to the polarity classifier module (also BERT) to create the sentiment prediction. SpanABSA outperforms existing unified AE/ASC training frameworks, especially in the case of targets being more than 2 words long.

Potential Research Direction # 5 (Aspect-based Multimodal Sentiment Analysis)

The most active areas of SA with regards to new research are that of aspect-based sentiment analysis, and multimodal sentiment analysis. A natural path for future research could be the union of these two areas, that is to say: given a video of someone speaking, classify not only the sentiment displayed, but also the target of the sentiment. This task promises to be quite challenging, as it would require leveraging both span-based AE/ASC models (e.g. SpanABSA [116]), and modality fusion techniques from existing multimodal models. Despite being a natural extension of existing research areas, this problem is almost entirely unexplored as of the time of writing, perhaps due to its obviously challenging nature.

5 Mining and Modelling Complex Networks

Mining complex networks represented as graphs is now a well-established sub-field within data science. Analyzing social networks by investigating metadata as well as the content produced by the users can be powerful but it does not capture the dynamics and various types of interactions between users. Such additional dimension can be naturally modelled as graphs in which nodes (associated with user accounts in the case of social networks) are connected to each other by edges (relations based on follower requests, similar user's behaviour, age, geographic location, etc.). Such networks are often large-scale, decentralized, and evolve dynamically over time.

Modelling complex networks using random graphs is a very active and popular area of research in pure mathematics, starting with a pioneering work of Paul Erdős and Alfréd Rényi [71] from 1959. The initial interest was mostly concentrated on investigating properties of the binomial random graph $\mathcal{G}(n,p)$ and random d-regular graphs but now the family of known and studied models include more realistic models of complex networks such as the Chung-Lu model [55] or various geometric graphs such as the spatial preferential attachment model [3] or the hyperbolic geometric graph [143].

Currently, we experience a rapid growth of research done in the intersection of mining and modelling of social networks. There are two main reasons to include random graph models in mining complex networks:

- **synthetic models**: Many important algorithms (such as community detection algorithms) are unsupervised in nature. Moreover, despite the fact that the research community gets better with exchanging datasets (see, for example, SNAP—Stanford Large Network Dataset Collection [157]), there are still very few publicly available networks with known underlying structure, the so-called ground truth. Hence, in order to test, benchmark, and properly tune unsupervised algorithms, one may use random graphs to produce synthetic "playground": graphs with known ground truth (such as the community structure in the context of community detection algorithms).

- **null-models**: Null-model is a random object that matches one specific property $\mathcal{P}$ of a given object but is otherwise taken, unbiasedly, at random from the family of objects that have property $\mathcal{P}$. As a result, the null-models can be used to test whether a given object exhibits some "surprising" property that is not expected on the basis of chance alone or as an implication of the fact that the object has property $\mathcal{P}$.

It is expected that both applications of random graphs will continue to gain their importance in the context of mining complex networks. There is a need for synthetic models of more general structures such as hypergraph as well as models that are dynamic (that is, they produce a sequence of graphs with both edge and node additions/deletions). Null-models are successfully used in designing clustering algorithms but they are expected to also play an important role in other aspects of mining networks, especially, as a predictive tool, when dynamics is captured in the form of a sequence of graphs or times when edges/nodes were added/deleted from the network.

In the following subsections, we identify a few areas in which more research seems to be needed. The outcome of such research projects may potentially be of interest to the industrial world as well as academia at large. We refer the reader to the recent book [133] for more standard applications of mining complex networks.

In each subsection, we briefly summarize current "state of the art" tools and approaches but also discuss various possible directions for future applied research. These parts will be clearly marked.

5.1 Node Embeddings

An embedding is a function from the set of nodes $V(G)$ of some graph G to $\mathbb{R}^k$, where k is typically much smaller than n. In other words, the embedding represents each node as a low-dimensional feature vector. The goal of this function is not only to decrease the dimension but to also preserve pairwise proximity between nodes as best as possible. Embedding-based representations emerged and quickly increased attention over the last decade. As reported in [49], the ratio between the number of papers published in top 3 conferences (ACL, WWW, KDD) closely related to Computational Social Science (CSS) applying symbol-based representations and the number of papers using embeddings decreased from 10 in 2011 to 1/5 in 2020.

A closely related way to achieve this is to conduct relational learning by propositionalization. In this approach, relational information is first captured and stored as propositions, according to some predefined declarative bias. Propositional learning algorithms can then be applied to learn using these extracted features. Such algorithms, often probabilistic, are well studied and available to use. Finally, one may enrich the embedding algorithms by including some domain-specific relations constructed using Inductive Logic Programming (ILP). It seems that incorporating symbolic domain knowledge might improve the quality of algorithms and ILP can play an important role in providing high-level relationships that are not easily discovered by other means [59].

We present a toy example in Figure 16: an embedding of the Zachary's karate club graph, a popular example of a network with community structure, first used in [89]. The graph represents social interactions between 34 members of a karate club; nodes of this graph are partitioned into two group as a result of a conflict between the club's president and the instructor.

There are many ways the proximity can be measured: first, second, and in general kth-order proximities, Katz Index, Personalized PageRank, Common Neighbours, Adamic Adar, SimRank (see, for example, [133] for the corresponding definitions). There are various known applications of node embeddings and many potential new directions are currently being explored, some of them will be mentioned later in this document.

- Node classification is an example of a semi-supervised learning algorithm where labels are only available for a small fraction of nodes and the goal is to label the remaining set of nodes based on this small initial seed set. Since embedding algorithms can be viewed as the process of extracting features of the nodes from the structure of the graph, one may reduce the problem to a classical machine learning

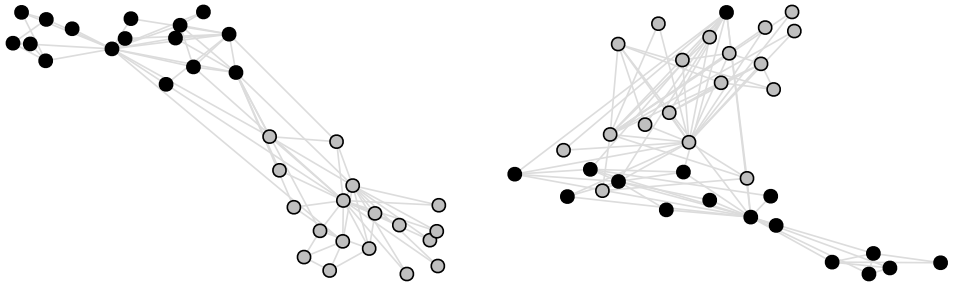

Figure 16: Two-dimensional projections of two sample embeddings of the Zachary karate club graph.

predictive modelling classification problem for the set of vectors.

- Community detection can be reduced to (or, at least, supported by) clustering points in $\mathbb{R}^k$ which is a much easier task and is a well-studied area of research with many scalable algorithms, such as k-means or DB-scan, that are easily available for use. Some initial results (embedding + vanilla k-means or Gaussian mixture models) show a potential of this approach [228].

- Link prediction algorithms (including predicting missing links or links that are likely to be formed in the future) can also be successfully built using node embeddings. Indeed, once nodes are embedded in k-dimensional space, one may use the distance between the corresponding vectors, combined with additional information, to build the corresponding null-model and to make the prediction.

There are over 100 algorithms proposed in the literature for node embeddings. The techniques and possible approaches to construct the desired embedding can be broadly divided into the following families. For more details, the reader is directed to [104], [258], and [133].

- Linear algebra algorithms include Local Linear Embedding (LLE) [196], Laplacian Eigenmaps (LEM) [22] and High Order Proximity preserving Embedding (HOPE) [182], which is an interesting instance of this approach aimed at embedding nodes in directed graphs.

- Random walk based node embedding methods are derived from the Word2Vec [175] algorithm for word embedding commonly used in Natural Language Processing (NLP). Two representative algorithms from this family are Node2Vec [96] and Deep Walk [184]. The common and general idea is as follows. The "words" are simply the nodes of a graph, and one generates "sentences" (sequences of nodes) via random walks on a graph.

- Deep learning methods have also successfully been used to produce node embeddings. Structural Deep Network Embedding (SDNE) [235] is an example of an

autoencoder, a type of artificial neural network that is a commonly used in deep learning to represent complex objects such as images. Graph Convolution Network (GCN) [141] and GraphSAGE [106] recursively extract and aggregate some important features similarly to the approach of the Recursive Feature Extraction (ReFeX) [110]. See [263, 245] for two sample comprehensive surveys (from the many available) on Graph Neural Networks (GNN).

Most of the algorithms fall into one or more of the categories defined above but some propose a different approach. One such algorithm is LINE [229], which explicitly defines two functions to encode the first and the second order proximity.

Hyperbolic Spaces

Some recent research show that, when embedding real-world graphs with scale-free or hierarchical structure, graph distances between pairs of nodes could not be accurately estimated based on the Euclidean distances between the corresponding pairs of embeddings. However, it turns out that embeddings in hyperbolic geometries seem to produce smaller distortion and so they offer a possible alternative for such families of graphs [31]. Currently, they seem to be slower than other methods [260] and so not suitable for large-scale networks but it might change in the near future.

In order to see the power of hyperbolic spaces, note that one may approximate tree (discrete) distances arbitrarily well while still being in a continuous space. One cannot hope to approximate this behaviour in Euclidean spaces, but this can be done in the hyperbolic ones, and with only two dimensions [198]. Moreover, it is possible to generalize important algorithms such as multidimensional scaling (MDS) and principal components analysis (PCA) to hyperbolic spaces.

The Hyperbolic Graph Convolutional Neural Network (HGCN) [46] is the first inductive hyperbolic counterpart of GCN that leverages both the expressiveness of GCNs and hyperbolic geometry to learn inductive node representations for hierarchical and scale-free graphs. Their flexibility (in particular, the curvature of the space) have been successfully leveraged in various areas such as computer vision, NLP, and computational biology. In the context of social networks, they are currently used in collaborative filtering where the goal is to use past user-item/advertisement interactions to build the recommender systems (see, for example, [224][13]).

Signed Networks

The vast majority of existing node embedding algorithms are designed for social networks without sign, typically only with positive links. However, in many social media platforms one may extract both positive and negative links, yielding signed networks. These additional information can be given explicitly; for example, a social news website `Slashdot.org` allows their users to specify other users as friends or foes, `Epinions.com` (currently `Shopping.com`) allows users to mark their trust or distrust to other users on

[13] `https://github.com/layer6ai-labs/HGCF`

product reviews. Alternatively, one may try to infer whether a given link is positive or negative by investigating the interaction between the two users (comments they produce, likes/dislikes of similar products or posts, etc.). This brings both challenges and opportunities for signed network embedding. While extracting such additional information seems challenging, some initial approaches and experiments indicate some potential, in particular, in link prediction algorithms. A few algorithms have been already proposed: SiNE [237], SNE [253], SNEA [236], ROSE [121]. Finally, let us mention about a hyperbolic node embedding algorithm for signed networks that was recently explored in [219].

Potential Research Direction # 6 (Embedding Sequences of Graphs)

Node embeddings extract important features of the nodes of the graph. In the context of social networks, they may be representing user's interests, beliefs, emotions, geographic location, age, gender, and many other demographic characteristics. In dynamic scenarios, with users joining/leaving the network, establishing new/deleting old connections, one may want to use embeddings to better understand how communities form/split, how/when/why users become segregated/polarized, who/when/why becomes central and powerful, etc. As a result, temporal graphs become an increasingly important object of study [104] and it is expected that extending node embedding techniques to include dynamic aspects will open up various exciting applications.

Suppose that we have snapshots of some network available at various times. Of course, one may embed each of them independently. Unfortunately, most of the embedding algorithms are randomized and so even if the very same graph is embedded twice, the two resulting embeddings will be completely different despite the fact that they try to preserve information from the same graph. Deterministic algorithms also do not guarantee that two similar graphs (say, measured via the edit distance, a measure of similarity, or dissimilarity, between two graphs) yield similar embeddings. In order to be able to use embeddings to better understand/predict dynamics, one needs to couple embeddings of any two consecutive snapshots such that they are similar to each other. For example, one may insist that vectors associated with nodes are not shifted by more than $\delta = \delta(\varepsilon)$ away between the two consecutive snapshots G_i, G_{i+1}, provided that the edit distance between the two graphs is equal to $\delta|E(G_i)|$.

Unfortunately, almost all existing node embedding algorithms are inherently transductive. As a result, when a new data point is added to the dataset, then one has to re-run the algorithm from the beginning to train the model. Some of them may be adjusted to inductive setting but they require additional rounds of gradient descent before prediction for new nodes can be used and so such adjustments are computationally expensive. GraphSAGE [106] is a rare example of graph convolutional networks (GCN) extended to the task of inductive unsupervised learning. For a few more recent attempts to deal with dynamic graphs see Section 4.3 in [263] and [118].

Potential Research Direction # 7 (Multi-Layered Graphs)

In some applications, we are provided with ℓ graphs $G_i = (V_i, E_i)$, $1 \leq i \leq \ell$, with overlapping sets of nodes. We may view it as one graph on the set of nodes $V = \bigcup_{i=1}^{\ell} V_i$ consisting of multiple "layers". This is a typical situation in the context of social networks where we often have access to datasets from more than one social platform or there are some auxiliary graphs constructed based on, for example, similarly between users implied by the posts they write and/or read.

The easiest approach would be to learn features in such multi-layered networks either by treating each layer independently of other layers, or by aggregating the layers into a single (weighted) network on the set of nodes V. However, as expected, ignoring the existence of multi-layered structure affects the topological and structural properties of such complex networks. As a result, the importance of individual nodes is altered and cannot be recovered from the simplified picture of the network, leading to wrong identification of versatile nodes and overestimating the importance of marginal nodes [63, 64, 62]—see also the book [21] which "provides a summary of the research done during one of the largest and most multidisciplinary projects in network science and complex systems" (Mathematical Review Clippings).

Despite the fact that we know more about processes shaping multi-layered networks, there are still only a few known dedicated embedding algorithms. One of them is OhmNet algorithm [266] that builds on recent success of unsupervised representation learning methods based on neural architectures. This algorithm uses a form of structured regularization suitable for multi-layer networks. It was tested for the human protein–protein interaction (PPI) network but the ideas and approaches might be potentially useful for multi-layered social networks.

5.2 Evaluating Node Embeddings

As mentioned in the previous section, there are 100+ node embedding algorithms, most of them have a number of hyper-parameters that one needs to carefully tune for a given task and a given network at hand. Moreover, most algorithms are randomized and not so stable which means that even if the algorithm is run twice on the same network and with the same set of parameters, the resulting embeddings might be substantially different. Some embedding algorithms seem to be performing better than others (for example, in the experiments performed recently in [65], Node2Vec worked relatively well for both real world networks as well as synthetically generated ones) but there is no universal choice for all potential applications. As a result, evaluating graph embedding algorithms is a challenging task, typically requiring ad-hoc experiments and tests performed by the domain experts.

However, in [128], local and global scores are proposed that can be assigned to outcomes of the embedding algorithms to help distinguish good ones from bad ones. This general *framework* provides an unsupervised tool for graph embedding comparison based on *global* and *local* properties of the graph based on the fact that a good embedding should be able to recover (to certain degree, of course) the structure of the graph. Scalable code

suitable for directed as well as undirected graphs, weighted or unweighted, is available on the associated GitHub repository[14].

This is a generalization of the original work presented in [131] where the authors generalized the classical Chung-Lu model [55] to incorporate geometry. Their Geometric Chung-Lu model is then used as the null-model to evaluate the quality of the competing embeddings. For the *global* score, the framework compares edge density between and within the communities that are found in the graph with the corresponding expected edge density in the associated random null-model via a divergence score. This global (divergence) score is designed to identify embeddings that should perform well in tasks requiring global knowledge of the graph such as node classification or community detection.

On the other hand, the *local* score is based on the ability of the embedding to predict directed or undirected adjacency between nodes in the graph. For example, it is natural to expect that if two vertices are embedded at points that are far away from each other, then the chance that they are adjacent in the graph is smaller compared to another pair of vertices that are close to each other. Embeddings with such property are naturally suitable for link-prediction algorithms.

The best embedding can be selected in an unsupervised way using one or both scores, depending on the context, or can be used to identify a few embeddings that are worth further investigation.

In order to illustrate the power of the framework, we present an experiment from [65] on another well-known real-world network with known community structure, namely, the College Football graph. This graph represents the schedule of United States football games between Division IA colleges during the regular season in Fall 2000 [89]. The teams are divided into conferences containing 8–12 teams each. In general, games are more frequent between members of the same conference than between members of different conferences, with teams playing an average of about seven intra-conference games and four inter-conference games in the 2000 season. There are a few exceptions to this rule, as detailed in [166]: one of the conferences is really a group of independent teams, one conference is really broken into two groups, and 3 other teams play mainly against teams from other conferences. We refer to those as *outlying* nodes, which we represent with a distinctive triangular shape. In Figure 17, we show the best and worst scoring embeddings based on the global (divergence) score. The colours of nodes correspond to the conferences, and the triangular shaped nodes correspond to outlying nodes as observed earlier. The communities are very clear in the left plot while in the right plot, only a few communities are clearly grouped together. In order to produce a low dimensional representation of high dimensional data that preserves relevant structure, the Uniform Manifold Approximation and Projection (UMAP[15]) [172] was used, a novel manifold learning technique for dimension reduction.

[14] `https://github.com/KrainskiL/CGE.jl`

[15] `https://github.com/lmcinnes/umap`

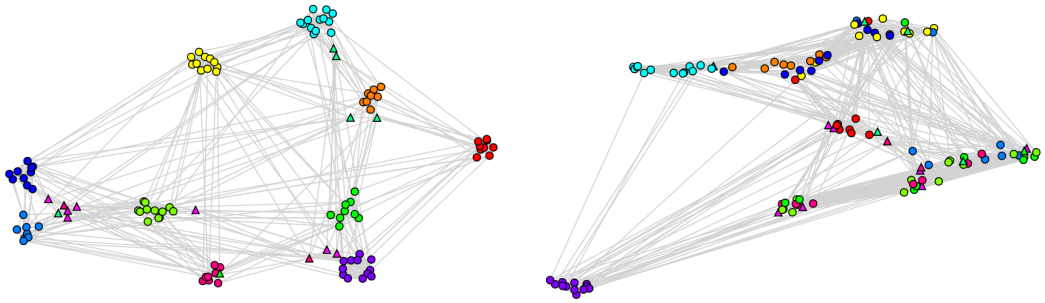

Figure 17: The best (left) and the worst (right) scoring embedding of the College Football Graph.

Potential Research Direction # 8 (Selecting an Appropriate Embedding for a Given Task at Hand—Supervised vs. Unsupervised Approach)

Selecting the best embedding can be done in unsupervised or supervised way. In order for the unsupervised approach to perform well, it is crucial to select an appropriate null-model for a given task. The framework mentioned above allows for tuning the selection process toward community detection algorithms or link prediction algorithms. The geometric Chung-Lu model is used as the null-model since it incorporates two important aspects, geometry as well as the degree distribution, but is simple enough so that one can easily compute the probability of two nodes to be adjacent in the model. Moreover, all edges are generated independently which allows one to investigate the desired properties and use them to benchmark embeddings. However, it is clearly not the only reasonable model to use. For a specific task at hand, or if some additional information about the graph or the embedding is provided, one may choose to use some other, more appropriate, model. In particular, it might be good to experiment with various random hyperbolic geometric graphs, especially if hierarchical networks are under investigation. Despite the fact that typically null-models are easy, let us mention that it is not a problem if the model is too complex for the desired properties to be theoretically analyzed—indeed, one may always use simulations to investigate them. Finally, when labelled training data becomes available, one may alternatively use (or complement the unsupervised "divergence score") supervised learning tools to benchmark the embeddings available and select the one that scores the best for a given task at hand.

5.3 Community Detection

A network has community structure if its set of nodes can be split into subsets that are densely internally connected. For example, in social networks, communities may represent user interests, occupation, age, etc. This is a very important (unsupervised) task when analyzing complex networks. Indeed, after finding communities one can better understand the role of each node, their interactions, and it allows one to focus on relevant portions of the graph, to name only a few possible applications. The first problem we consider

is node partitioning, where we seek to divide the set of n nodes into k non-overlapping subsets, where each subset yields a dense community subgraph. Note that the number of communities (that is, the value of k) is generally unknown, and the number of possible partitions is enormous even for small graphs.

There are several natural ways to define "dense subgraph" to make the task of finding communities well-defined but the general idea is that the number of internal connections (edges) should be larger than the expected number of connections based on some global statistics of the whole graph. One commonly used measure that is used to find communities is the modularity function [179], which is based on the comparison between the actual density of edges inside a community and the density one would expect to have if the nodes of the graph were attached at random, without any community structure, while respecting the nodes' degrees.

Several graph clustering algorithm aim at maximizing the modularity function, one of the best ones being the well-known Louvain algorithm [30]. In this algorithm, small communities are first found by optimizing modularity locally on all nodes. This yields the level-1 partition of the nodes. Then, each small community is grouped into one node and the original step is repeated on this smaller graph, yielding the next level of partition. The process is repeated until no improvement on the modularity function can be achieved. As a result, a hierarchy of partitions is obtained with decreasing granularity.

The Louvain algorithm offers good trade-off between the quality of the clusters it produces and its speed but it has some stability and resolution issues. Instability is due to the randomization of the node ordering when locally optimizing the modularity, which can lead to very different partitions when running the algorithm multiple times on the same graph. Resolution issue is an issue with modularity itself [82] and it can lead to the merger of small communities. Such behaviour can be addressed by trying to "break up" the communities individually or by increasing aversion of nodes to form communities by adding self-loops or modifying the modularity function directly.

Another way to address the above issues is by using an ensemble of partitions, as proposed with the Ensemble Clustering for Graphs (ECG) algorithm [187]. With ECG, we start by running the level-1 of the Louvain algorithm several times, thus building an ensemble of highly granular partitions. For each edge, the number of times it is internal (that is, both nodes are within the same community) for the partitions in the ensemble is used as edge weight to obtain the final partition using the Louvain algorithm. Thus, not only are the obtained partitions more stable and often of better quality than with the "vanilla" Louvain, but the ECG derived edge weights are useful to assess the quality of the clusters we get. We illustrate this is Figure 18, where we ran ECG on a small network with two communities: red nodes form a weakly connected community (40% of pairs have an edge) and blue nodes form a tight community (90% of pairs have an edge). We also report the edges with high ECG weights with thicker black lines. All edges in the blue community have maximal ECG weight of 1, which is not the case for the red community, but in this case, we do see that the tight sub-structures (triangles) do have higher ECG weights.

There are several other graph clustering algorithms (see for example [81], or [249]). For example, the Leiden algorithm [232] is a proposed improvement to the Louvain algorithm

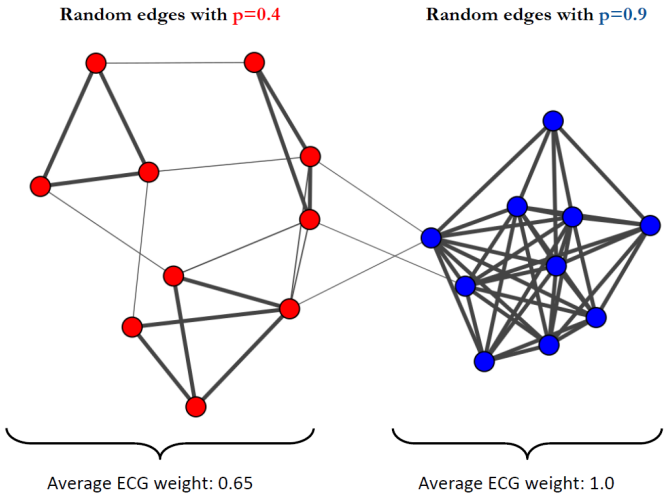

Figure 18: Clustering a small graph with two communities: weak (red, edge density 0.4) and strong (blue, edge density 0.9) using ECG. We see that ECG derived weights (thick lines have high weights) are useful in identifying tight communities and sub-structures.

fixing issues such as disconnected communities; note that Leiden can be used within the ECG code[16]. CNM [56] is a hierarchical, agglomerative algorithm based on the modularity function. Other hierarchical algorithms include Ravasz agglomerative algorithm [193] based on the topological overlap, and the Girvan-Newman divisive algorithm [90] which is based on edge betweenness centrality. Other algorithms include label propagation [191], spectral bisection [80] and Infomap [195], to name a few. For more details, we refer the reader to, for example, [133].

Potential Research Direction # 9 (More General Community Detection and Using Several Sources of Information)

Graph partitioning to find communities is a very useful tool in network science but, depending on the problem at hand, one or several of the following generalizations may be of relevance.

- Some nodes could be allowed to be excluded from communities (so-called "noise").
- Nodes could be part of several communities; this is the overlapping communities problem.
- The same nodes could interact over several distinct networks, for example, Twitter, Facebook, and Reddit.

[16] github.com/ftheberge/graph-partition-and-measures

Ideas based on ensemble clustering as in ECG could be useful to address the generalizations above. For example, ECG derived weights could be used to identify "noise" nodes after removing weak edges. Similarly, nodes having strong edges spanning several communities could indicate overlapping clusters with these nodes belonging to more than one cluster. There are some known methods for finding overlapping communities such as clique percolation [66], which looks for overlapping cliques (small fully connected subgraphs such as triangles), ego-splitting [70], where nodes can be duplicated to properly reflect their multiple community membership, and edge clustering [2].

Independently, one may try to use some external information to improve the quality or the stability of the clustering algorithm.

- Extra information could be available for the nodes; for example if those represent people interacting over social networks, we may know about the country of origin, language(s) used, interests, text generated, etc.

- Extra information could be available for the edges; for example the topic of a message sent, the sentiment (such as like or dislike), the timestamp(s), etc.

Such metadata could, for example, be used to obtain an embedding of the nodes, followed by some clustering of the points to obtain several tight clusters. This, in turn, can be used as starting point for a Louvain-like algorithm to improve its quality. If several sources of metadata are available, then an ensemble methods could be tried. Finally, another possibility is to modify the modularity function itself to favour clustering of nodes such that the corresponding points exhibit high level of similarity derived from the external information available for the nodes and edges.

5.4 Hypergraphs

Real-world complex networks are usually being modelled as graphs. The concept of graphs assumes that the relations between nodes within the network are binary; however, this is not always true for many real-life scenarios, including social network interactions. For example, a group of users commenting on the same post or liking/disliking a given picture typically consists of more than two users.

Hypergraph $H = (V, E)$ consists of the set of nodes V and the set of hyperedges E; each hyperedge $e \in E$ is a subset of V, but not necessarily of size 2 as in the case of graphs. As a result, it is a natural generalization of graphs. More importantly, many complex networks that are currently modelled as graphs would be more accurately modelled as hypergraphs. This includes the collaboration network in which nodes correspond to researchers and hyperedges correspond to papers that consist of nodes associated with researchers that co-authorship a given paper. Similarly, social networks can be modelled as hypergraphs with hyperedges consisting of all users that interact with each other by commenting on the same post/article. However, here the situation is even more complex. Indeed, this type of data can be more accurately modelled by a dynamic tree structure with the root associated with a given post/article followed by a tree of comments/likes/dislikes.

Unfortunately, the theory and tools are still not sufficiently developed to allow most problems (including community detection algorithms and centrality measures) to be tackled directly within this context. As a result, researchers and practitioners often create the 2-section graph of a hypergraph of interest, that is, replace each hyperedge with a clique—see Figure 19. After moving to the 2-section graph, one clearly loses some information about hyperedges of size greater than two and so there is a common believe that one can do better by using the knowledge of the original hypergraph. Having said that, there has been a recent surge of interest in higher-order methods, especially in the context of hypergraph clustering which we will concentrate on from now on. See [20] for a recent survey on the higher-order architecture of real complex systems, and [78] for a hypergraph neural networks framework (HGNN) for data representation learning.

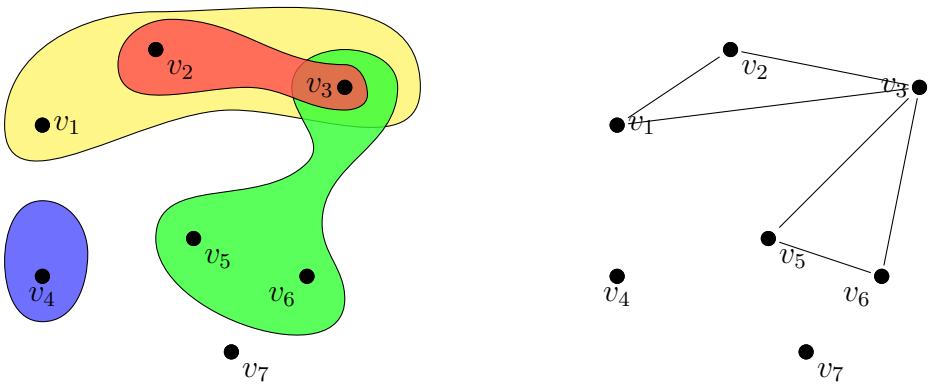

Figure 19: Hypergraph (left) and its 2-section (right).

Graph clustering is the task of partitioning the set of nodes of a given graph into clusters such that clusters are substantially denser than the global density of a network (see the two surveys, by Fortunato and Hric [83], and by Schaeffer [203]). Since there are various ways to partition the nodes of a hyperedge, there are numerous generalizations of a graph-based objective function to hypergraphs. Various notions of hyperedge cuts have been considered in the past (see references in [234]). Recently, a hypergraph clustering objective was proposed in [234] that differently treats hyperedges and pairwise edges in a parametric fashion.

Potential Research Direction # 10 (Hypergraph Modularity Function)

There are some recent attempts to deal with hypergraphs in the context of graph clustering. Kumar *et al.* [145, 146] still reduce the problem to graphs but use the original hypergraphs to iteratively adjust weights to encourage some hyperedges to be included in some cluster but discourage other ones. Moreover, in [129] a number of extensions of the classic null model for graphs are proposed that can potentially be used by true hypergraph algorithms.

Unfortunately, there are many ways such extensions can be done depending on how often nodes in one community share hyperedges with nodes from other communities. Fortunately, it is possible to unify all variants of the graph modularity function to hypergraphs and put them into one framework. Two prototype algorithms investigated in [130] show the potential of the framework but its true power needs to be confirmed by experiments on larger networks. The authors of that paper still work on designing a scalable algorithm.

Finally, let us stress it again that hypergraphs are just the beginning and there are many ways to generalize hypergraphs even further. One important generalization is the one in which nodes are assigned roles within each hyperedge. Many complex networks provide such additional information: research articles have junior and senior authors, political bills have sponsors and supporters, movies have starring and supporting actors, e-mails have senders, receivers, and carbon copies, etc. In order to model these networks, annotated hypergraphs were introduced in [53] as natural polyadic generalizations of directed graphs. To facilitate data analysis with annotated hypergraphs, the authors of that paper construct a role-aware configuration null-model for these structures and prove an efficient Markov Chain Monte Carlo scheme for sampling from it. In order to take advantage of this additional information, one should adjust the modularity function once again to incorporate this new null-model.

5.5 Understanding the Dynamics of Networks

One of the very first model of dynamic network is the preferential attachment model introduced in the paper of Barabási and Albert [17] who observed a power law degree sequence for a subgraph of the World Wide Web; soon after the same property was observed for the internet graph [72]. It is an important model as it explains why power-law degree distribution occurs in many real-world complex networks. Indeed, this model was rigorously analyzed in [34, 33] and the conclusion is as follows. The existence of a power-law degree distribution is a consequence of a rich-get-richer phenomenon: the preferential attachment rule incorporated in the model makes new nodes to be more likely to connect to the more connected nodes than to the smaller nodes, creating as a result power-law degree distribution.

One drawback of the above mentioned model is that there is a strong correlation between the age of a node and its degree, making it impossible for young nodes to gain a substantial number of neighbours. The fitness model introduced in [27] assigns a fitness to new born nodes that corresponds to some intrinsic property that propels them ahead of the pack. Another problem of the original model is that it generates power-law degree distribution but the exponent cannot be easily tuned. This issue was addressed in [181] and in the very recent work, a preferential rule was adjusted to mimic any empirical degree distribution [88].

A model is said to have accelerating growth if the number of edges grows non-linearly with the number of vertices. Such models became important as there is an evidence of increasing edge density (densification) and decreasing diameter in existing networks [155, 156]. Among networks found by the authors to exhibit increasing average out-degree over

time were: ArXiv citation, patent citation and autonomous systems (internet routers). The authors of that paper were concerned to find causal models of densification, and propose explanatory mechanisms for this, such as community guided attachment, and the forest fire model. In particular, simulations of the forest fire model show that densification itself does not necessarily cause the diameter to shrink. Another accelerating growth model is the preferential attachment model in which the degrees of newly added nodes increase over time [58].

There are many other models that explain the occurrence of various typical properties of complex networks. The Watts–Strogatz model produces graphs with short average path lengths and large clustering coefficient [240]. There are also many geometric models of interest, one of them is the Spatial Preferential Attachment model that combines the rich-get-richer phenomenon with geometrical aspects [3, 57]. For more examples of such models see, for example, [68].

Another type of models have the purpose to represent network dynamics on the basis of observed longitudinal data, and evaluate these according to the paradigm of statistical inference. Simulation Investigation for Empirical Network Analysis (SIENA)[17] is a software to preform the statistical estimation of models for repeated measures of social networks according to the Stochastic Actor-oriented Model. For more details, see a review article [215] or the manual for SIENA 4.0[18] compiled recently (May 11, 2021). This model has an "actor-oriented" nature which means that it models change from the perspective of the actors (nodes). According to the manual, the model is suitable for the analysis of:

- the evolution of a directed or non-directed one-mode network (e.g., friendships in a classroom),
- the evolution of a two-mode network (e.g., club memberships in a classroom: the first mode is constituted by the students, the second mode by the clubs),
- the evolution of an individual behaviour (e.g., smoking), and
- the co-evolution of one-mode networks, two-mode networks and individual behaviours (e.g., the joint evolution friendship and smoking; or of friendship and club membership).

An important drawback is that this method is applicable to only very small graphs consisting of approximately 10 to 1,000 nodes. However, some ideas and approaches might be useful for large graphs.

Another general modeling framework for temporal network data analysis is to extend Exponential Random Graph Models (ERGMs) [107] to Temporal Exponential family Random Graph Models (TERGMs) [153][19]. For example, Separable Temporal ERGMs (STERGMs) are an extension of ERGMs for modeling dynamic networks in discrete time,

[17] `https://www.stats.ox.ac.uk/~snijders/siena/`
[18] `https://www.stats.ox.ac.uk/~snijders/siena/RSiena_Manual.pdf`
[19] `http://statnet.org/Workshops/tergm_tutorial.html`

introduced in [144][20]. Within this framework, one may obtain maximum-likehood estimates for the parameters of a specified model for a given data set; simulate additional networks with the underlying probability distribution implied by that model; test individual models for goodness-of-fit, and perform various types of model comparison.

Let us now briefly discuss a few aspects that seems to be important in the context of dynamics of social networks.

Human-bot Interaction and Spread of Misinformation

Social networks created an information platform in which automated accounts (both human as well as bots—software-assisted accounts) can try to take advantage of the system for various opportunistic reasons: trigger collective attention [152, 61], gain status [45, 222], monetize public attention [43], diffuse disinformation [15, 84], or seed discord [242].

It is known that a high fraction of active Twitter accounts are bots and that they are responsible for much disinformation. We better understand the role they play in the diffusion of false information. In particular, it seems that they play an important role at the initial stage of diffusion by amplifying low-credibility content but they cannot distinguish between true and false, that is, real human accounts are more likely to spread false news. (See [93] and the references there.) Indeed, it is possible to characterize the peculiar behaviour of certain individuals who massively use bots to enhance their online visibility and influence. The term *cyborg* or *augmented human* has been used in this context to identify, indistinctly, bot-assisted human or human-assisted bot accounts. (See [222] and the references there.) It is also worth mentioning that spreading information and banning strategies depend on whether social platform is moderated or not [12]. In a recent paper, a framework for learning/opinion formation of an individual from signed network data is proposed [173]. This framework can be used to understand, for example, why people end up trusting misinformation sources. Finally, let us mention that various countries have different levels of infodemic risk [87], adding another level of complexity for a person trying to model these processes.

Despite the fact that we understand social networks better, it is clear that the exact mechanisms responsible for spreading false information (for example, during political events) are still far from being well understood. In fact, there are some recent results that suggest that spreading mechanisms might have indistinguishable population-level dynamics [109].

Social Bursts in Collective Attention

We are all flooded with a large amount of information, impossible to consume. Indeed, human attention is a limited resource and a way a given individual reacts to a given information is a complex interplay between individual interests and social interaction. It is known that a collective attention is typically characterized by a quickly growing

[20] `https://cran.r-project.org/web/packages/tergm/vignettes/STERGM.pdf`

accumulated focus on a specific topic (for example, presidential elections) until a well identified peak of collective attention is reached. This first phase is followed by the second phase in which one observes a slow decay of interest. (See [152] or [61] and the references there.)

Some initial research neglected the effects of the underlying social structure [243] but it is clear that underlying network structure plays an important role in the process [91]. In particular, the authors of [61] combine two simple mechanisms to explain the dynamic of the collective attention: a preferential attachment process shaping the network topology and a preferential attention process responsible for individual's attention bias towards specific users of the network.

Social Learning (Segregation, Polarization)

Social learning is a term that refers broadly to the processes by which a person's social environment shapes their actions (how a person behaves) and cognitions (how a person thinks). For example, in a recent paper [207] the authors show that asset discussions on WallStreetBets (WSB) are self-perpetuating: an initial set of investors attracts a larger and larger group of excited followers. Based on that, a model for how social contagion impacts prices is developed. Similarly, in the context of social networks a user may adopt the cognitions or behaviours from those they have an opportunity to interact with directly. At the same time, products of learning also shape the social environments, since individuals also exercise control over their social environment and potentially select network partners as a function of individual attributes, including their behaviours and cognitions [10]. As a result, social learning is a complex process; here we only concentrate on segregation and polarization.

There are various models that try to explain why and how segregation occurs, starting from the classic model of residential segregation of Schelling [204]. Schelling's results also apply to the structure of networks; namely, segregated networks always emerge even if the users are assumed to have only a small aversion from being connected to others who are dissimilar to themselves, and yet no actor strictly prefers a segregated network [111]. The power of aversion is often amplified by homophily, a tendency for people to have ties with people who are similar to themselves [41]. For example, [24] develops and tests empirical models of how social networks evolve over time; particularly, how people in a social network choose links on the basis of their own attributes, and how individual attributes are in turn shaped by network structure. In [160], the authors try to investigate how homophily shapes the topology of adaptive networks (see also a long list of other related models referenced in that paper).

An extreme situation occurs when a group is divided into two opposing sub-groups having conflicting and contrasting positions, goals and points of view, with only a few individuals remaining neutral. A domain where polarization typically occurs is politics but there are other domains that often experience it, such as global warming, gun control, same-sex marriage or abortion. The modularity function mentioned earlier identifies communities but it is known that it is not a direct measure of antagonism between groups [99]. As a result, other metrics being able to identify and to analyze the boundary

of a pair of polarized communities were proposed, which better captures the notions of antagonism and polarization. The presence of polarization changes the dynamics of a network. Indeed, for example it is known that Twitter users are unlikely to be exposed to cross-ideological content from the clusters of users they followed, as these were usually politically homogeneous [112]. (See also a recent survey of Twitter research [9].)

Potential Research Direction # 11 (Tools Based on the Null-models)

Apart from the theoretical interest, models of complex networks can have significant impact on the design of practical tools. Indeed, understanding the principles driving the organization and behaviour of such networks proved to be helpful to design various important tools: sampling algorithms [154], recommendation systems [208], defence systems against attacks from bots [79] and spam campaigns [25], and measuring of users' influence [178]. Having said that, there is clearly a room for more research in this area. Below we mention a few potential directions.

Community Detection: The Chung-Lu random graph model is currently used to define the modularity function which guides the Louvain algorithm to find communities. It tries to find a partition that induces "surprisingly" dense communities in comparison to the underlying null-model. It could also be used to identify overlapping clusters.

Anomalies Detection: Going down to the level of nodes, the same Chung-Lu random null-model should also be able to identify leaders, followers, and other "unusual" members of the community. Initial experiments on the College Football network presented in [133] show that this approach has a potential in identifying anomalies. There are several other methods for graph-based anomaly detection that could be possibly investigated; see for example [4] and [188].

Link Prediction: Combining the null-model (in this case, random geometric graph) with a carefully selected embedding of a graph can be a useful tool in link prediction algorithms. Such algorithms should take into account not only positions of the nodes in the embedded space but complement it with some additional information such as community membership and triadic closure (a measure of the tendency of edges to form triangles), to make a better, density based, link prediction.

Many social networks are heterogeneous by nature, that is, there are various types of relationships that are present in these networks. One may combine all of these relationships into a single weighted network but some important information is lost during this process. A few algorithms try to take advantage of heterogeneous information—see, for example, [225, 226] for link prediction algorithms—but more work in this area is clearly needed.

Even less tools try to take advantage of dynamics aspects. On the other hand, for example, understanding how communities are formed should allow us not only to identify current communities but also to predict the future shape of a network. Moreover, simple principles such as rich-get-richer or fitness might be enough to build a model which can predict which nodes are going to be central and influential in the future, and which nodes are losing their power.

Finally, a good tool should try to combine all possible aspects (dynamics, heterogeneity, higher-order structures, community distribution, metadata, etc.) to "squeeze the last drop" from the dataset. There are very few techniques that combine at least 2 of such aspects. One such example is the recent paper [75] that combines heterogeneity and temporal evolution to make better link predictions.

5.6 Generating Synthetic Networks

Many machine learning algorithms and tools are unsupervised in nature, including community detection and anomalies detection algorithms. Unfortunately, these algorithms are often quite sensitive and so they cannot be fine-tuned for a given family of networks we want these algorithms to work on. For example, some algorithms perform well on networks with strong communities but perform poorly on graphs with weak communities; often density or degree distribution affect both the speed as well as the quality of a given algorithm, etc. Because of that it is important to be able to test these algorithms for various scenarios that can only be done using synthetic graphs that have built-in community structure, power-law degree distribution, and other typical properties observed in complex networks.

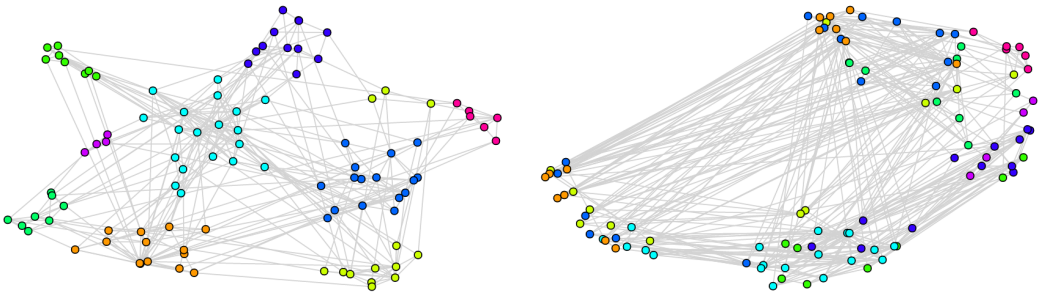

Figure 20: Two embeddings of the same LFR graph with ground-truth communities shown in colour, respectively with good (left) and bad (right) global (divergence) scores obtained by the framework presented in Section 5.2.

Classical random graphs, such as binomial random graphs or random regular graphs, are very interesting from theoretical point of view [32, 120, 85] but are not suitable for practical applications. Hence, in order to model the community structure, the Stochastic Block Model (SBM) [113] was introduced (see [86] for an overview of various generalizations). On the other hand, Chung-Lu model [55] was introduced to generate graphs with non-homogeneous degree distributions, including power-law degree distribution that is commonly present in complex networks. The LFR (Lancichinetti, Fortunato, Radicchi) model [151, 150] generates networks with communities and at the same time it allows for the heterogeneity in the distributions of both vertex degrees and of community sizes—see Figure 20. As a result, it became a standard and extensively used method for generating artificial networks.

An alternative, "LFR-like" random graph model, the Artificial Benchmark for Community Detection (ABCD graph) [132] was recently introduced and implemented[21], including an implementation that uses multiple threads (ABCDe)[22]. LFR and ABCD produce graphs with comparable properties but ABCD/ABCDe are faster than LFR and can be easily tuned to allow the user to make a smooth transition between the two extremes: pure (independent) communities and random graph with no community structure. Moreover, the simplicity of the ABCD/ABCDe models allows for the derivation of several theoretical properties as shown in [136] and [135]. Finally, the building blocks in the model are flexible and may be adjusted to satisfy different needs. For example, the original ABCD model was recently adjusted to include potential outliers in [134] resulting in ABCD+o model.

Potential Research Direction # 12 (Generating Synthetic Higher-order Structures)

As mentioned a few times earlier, many complex networks (including social networks) are better modelled with higher-order structures such as hypergraphs. Synthetic graph models are available but there is a need for more scalable hypergraph models that mimimc the properties of real world networks. (Let us again mention the annotated hypergraphs that were recently introduced in [53] but such models are still rare.) Good models of graphs/hypergraphs with communities (including overlapping communities) and/or anomalous nodes are also in need for a purpose of testing and tuning potential community/anomaly detection algorithms. Finally, it would be good to have a synthetic model that generates a sequence of graphs/hypergraphs to be able to train and benchmark the algorithms that try to capture the dynamic of networks.

[21] `https://github.com/bkamins/ABCDGraphGenerator.jl/`
[22] `https://github.com/tolcz/ABCDeGraphGenerator.jl/tree/CFGparallel`

6 Conclusions

We have presented above a comprehensive survey of generative methods for social media analysis. Its topic is timely and needed, given the recent interest in social media and their role in communications, community building, fake news etc. The survey covers four fundamental areas of social media analytics: relevant ontologies and data management aspects, natural language text generation and social media, sentiment analysis in the social media context, and the network approaches to social media understanding. Each of the chapters is rounded up by a discussion of the current limitations and potential future work in the area of the chapter.

References

[1] A. Abid, M. Farooqi, and J. Zou. Persistent Anti-Muslim Bias in Large Language Models. *arXiv:2101.05783 [cs]*, Jan. 2021. arXiv: 2101.05783.

[2] Y. Ahn, J. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.

[3] W. Aiello, A. Bonato, C. Cooper, J. Janssen, and P. Prałat. A spatial web graph model with local influence regions. *Internet Mathematics*, 5(1-2):175–196, 2008.

[4] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data Min Knowl Disc*, 29:626–688, 2015.

[5] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[6] N. Aletras and M. Stevenson. Evaluating Topic Coherence Using Distributional Semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 13–22, Potsdam, Germany, Mar. 2013. Association for Computational Linguistics.

[7] G. Alexandridis, I. Varlamis, K. Korovesis, G. Caridakis, and P. Tsantilas. A survey on sentiment analysis and opinion mining in greek social media. *Information*, 12(8):331, 2021.

[8] D. Angelov. Top2vec: Distributed representations of topics, 2020.

[9] D. Antonakaki, P. Fragopoulou, and S. Ioannidis. A survey of twitter research: Data model, graph structure, sentiment analysis and attacks. *Expert Systems with Applications*, 164:114006, 2021.

[10] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[11] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*, Dec. 2017. arXiv: 1701.07875.

[12] O. Artime, V. d'Andrea, R. Gallotti, P. L. Sacco, and M. De Domenico. Effectiveness of dismantling strategies on moderated vs. unmoderated online social platforms. *Scientific reports*, 10(1):1–11, 2020.

[13] S. K. B, A. Chandrabose, and B. R. Chakravarthi. An Overview of Fairness in Data – Illuminating the Bias in Data Pipeline. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 34–45, Kyiv, Apr. 2021. Association for Computational Linguistics.

[14] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, May 2016. arXiv: 1409.0473.

[15] C. A. Bail, B. Guay, E. Maloney, A. Combs, D. S. Hillygus, F. Merhout, D. Freelon, and A. Volfovsky. Assessing the russian internet research agency's impact on the political attitudes and behaviors of american twitter users in late 2017. *Proceedings of the national academy of sciences*, 117(1):243–250, 2020.

[16] M. Bao, J. Li, J. Zhang, H. Peng, and X. Liu. Learning Semantic Coherence for Machine Generated Spam Text Detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2019. ISSN: 2161-4407.

[17] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[18] N. Barbieri, F. Bonchi, and G. Manco. Who to follow and why: link prediction with explanations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1266–1275, 2014.

[19] B. Batrinca and P. C. Treleaven. Social media analytics: a survey of techniques, tools and platforms. *Ai & Society*, 30(1):89–116, 2015.

[20] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 2020.

[21] S. Battiston, G. Caldarelli, and A. Garas, editors. *Multiplex and Multilevel Networks*. Oxford University Press, 2019.

[22] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Nips*, volume 14, pages 585–591, 2001.

[23] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623, New York, NY, USA, Mar. 2021. Association for Computing Machinery.

[24] A. B. Bener, B. Çağlayan, A. D. Henry, and P. Prałat. Empirical models of social learning in a large, evolving network. *PloS one*, 11(10):e0160307, 2016.

[25] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[26] F. Bianchi, S. Terragni, and D. Hovy. Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence. *arXiv:2004.03974 [cs]*, Apr. 2020. arXiv: 2004.03974.

[27] G. Bianconi and A.-L. Barabási. Competition and multiscaling m evolving networks. In *The Structure and Dynamics of Networks*, pages 361–367. Princeton University Press, 2011.

[28] M. Bilal, A. Gani, M. I. U. Lali, M. Marjani, and N. Malik. Social profiling: A review, taxonomy, and challenges. *Cyberpsychology, Behavior, and Social Networking*, 22(7):433–450, 2019.

[29] B. Bischke, D. Borth, C. Schulze, and A. Dengel. Contextual enrichment of remote-sensed events with social media streams. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1077–1081, 2016.

[30] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics Theory and Experiment*, 2008, 04 2008.

[31] M. Boguna, I. Bonamassa, M. De Domenico, S. Havlin, D. Krioukov, and M. Á. Serrano. Network geometry. *Nature Reviews Physics*, pages 1–22, 2021.

[32] B. Bollobás. *Random graphs*. Number 73. Cambridge university press, 2001.

[33] B. Bollobás and O. Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24(1):5–34, 2004.

[34] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. In *The Structure and Dynamics of Networks*, pages 384–395. Princeton University Press, 2011.

[35] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *arXiv:1607.06520 [cs, stat]*, July 2016. arXiv: 1607.06520.

[36] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[37] K. Bontcheva, H. Cunningham, I. Roberts, A. Roberts, V. Tablan, N. Aswani, and G. Gorrell. Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029, 2013.

[38] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 223–232, 2013.

[39] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020. arXiv: 2005.14165.

[40] J. Bullock and M. Luengo-Oroz. Automated Speech Generation from UN General Assembly Statements: Mapping Risks in AI Generated Texts. *arXiv:1906.01946 [cs]*, June 2019. arXiv: 1906.01946.

[41] D. Byrne. An overview (and underview) of research and theory within the attraction paradigm. *Journal of Social and Personal Relationships*, 14(3):417–431, 1997.

[42] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel. Extracting Training Data from Large Language Models. *arXiv:2012.07805 [cs]*, Dec. 2020. arXiv: 2012.07805.

[43] D. Carter. Hustle and brand: The sociotechnical shaping of influence. *Social Media+ Society*, 2(3):2056305116666305, 2016.

[44] P. R. Center. Social media update 2016, 2016.

[45] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 4, 2010.

[46] I. Chami, Z. Ying, C. Ré, and J. Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[47] Y. Chaudhary, P. Gupta, K. Saxena, V. Kulkarni, T. Runkler, and H. Schütze. TopicBERT for Energy Efficient Document Classification. *arXiv:2010.16407 [cs]*, Oct. 2020. arXiv: 2010.16407.

[48] T. Che, Y. Li, R. Zhang, R. D. Hjelm, W. Li, Y. Song, and Y. Bengio. Maximum-Likelihood Augmented Discrete Generative Adversarial Networks. *arXiv:1702.07983 [cs]*, Feb. 2017. arXiv: 1702.07983.

[49] H. Chen, C. Yang, X. Zhang, Z. Liu, M. Sun, and J. Jin. From symbols to embeddings: A tale of two representations in computational social science. *arXiv preprint arXiv:2106.14198*, 2021.

[50] T. Chen, D. Borth, T. Darrell, and S.-F. Chang. Deepsentibank: Visual sentiment concept classification with deep convolutional neural networks. *arXiv preprint arXiv:1410.8586*, 2014.

[51] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang. Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection. *arXiv:1704.05973 [cs]*, Apr. 2017. arXiv: 1704.05973.

[52] K.-L. Chiu and R. Alexander. Detecting Hate Speech with GPT-3. *arXiv:2103.12407 [cs]*, Mar. 2021. arXiv: 2103.12407.

[53] P. Chodrow and A. Mellor. Annotated hypergraphs: models and applications. *Applied Network Science*, 5(1):1–25, 2020.

[54] A. Chouldechova and A. Roth. The frontiers of fairness in machine learning, 2018.

[55] F. Chung and L. Lu. *Complex Graphs and Networks*. American Mathematical Society, 2006.

[56] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, Dec 2004.

[57] C. Cooper, A. Frieze, and P. Prałat. Some typical properties of the spatial preferred attachment model. *Internet Mathematics*, 10(1-2):116–136, 2014.

[58] C. Cooper and P. Prałat. Scale-free graphs of increasing degree. *Random Structures & Algorithms*, 38(4):396–421, 2011.

[59] T. Dash, A. Srinivasan, and L. Vig. Incorporating symbolic domain knowledge into graph neural networks. *Machine Learning*, pages 1–28, 2021.

[60] T. Daudert. A web-based collaborative annotation and consolidation tool. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7053–7059, 2020.

[61] M. De Domenico and E. G. Altmann. Unraveling the origin of social bursts in collective attention. *Scientific reports*, 10(1):1–9, 2020.

[62] M. De Domenico, C. Granell, M. A. Porter, and A. Arenas. The physics of spreading processes in multilayer networks. *Nature Phys*, 12:901–906, 2016.

[63] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas. Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences*, 111(23):8351–8356, 2014.

[64] M. De Domenico, A. Solé-Ribalta, E. Omodei, S. Gómez, and A. Arenas. The physics of spreading processes in multilayer networks. *Nat Commun*, 6:6868, 2015.

[65] A. Dehghan-Kooshkghazi, B. Kamiński, L. Kraiński, P. Prałat, and F. Théberge. Evaluating node embeddings of complex networks. *Journal of Complex Networks*, 10(4), August 2022. doi.org/10.1093/comnet/cnac030.

[66] I. Derényi, G. Palla, and T. Vicsek. Clique percolation in random networks. *Phys. Rev. Lett.*, 94:160202, Apr 2005.

[67] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

[68] S. N. Dorogovtsev and J. F. Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. OUP Oxford, 2013.

[69] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness Through Awareness. *arXiv:1104.3913 [cs]*, Nov. 2011. arXiv: 1104.3913.

[70] A. Epasto, S. Lattanzi, and R. Paes Leme. Ego-splitting framework: From non-overlapping to overlapping clusters. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 145–154, New York, NY, USA, 2017. Association for Computing Machinery.

[71] P. Erdős and A. Rényi. On random graphs i. *Publ. Math. Debrecen*, 6:290–297, 1959.

[72] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *The Structure and Dynamics of Networks*, pages 195–206. Princeton University Press, 2011.

[73] S. Faralli, G. Stilo, and P. Velardi. Large scale homophily analysis in twitter using a twixonomy. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[74] S. Faralli, G. Stilo, and P. Velardi. What women like: A gendered analysis of twitter users' interests based on a twixonomy. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, 2015.

[75] A. M. Fard, E. Bagheri, and K. Wang. Relationship prediction in dynamic heterogeneous information networks. In *European Conference on Information Retrieval*, pages 19–34. Springer, 2019.

[76] J. Feng, Z. Zhang, C. Ding, Y. Rao, and H. Xie. Context Reinforced Neural Topic Modeling over Short Texts. *arXiv:2008.04545 [cs]*, Aug. 2020. arXiv: 2008.04545.

[77] R. Feng, Y. Yang, Y. Lyu, C. Tan, Y. Sun, and C. Wang. Learning Fair Representations via an Adversarial Framework. *arXiv:1904.13341 [cs, stat]*, Apr. 2019. arXiv: 1904.13341.

[78] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019.

[79] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016.

[80] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.

[81] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[82] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[83] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[84] D. Freelon, M. Bossetta, C. Wells, J. Lukito, Y. Xia, and K. Adams. Black trolls matter: Racial and ideological asymmetries in social media disinformation. *Social Science Computer Review*, page 0894439320914853, 2020.

[85] A. Frieze and M. Karoński. *Introduction to random graphs*. Cambridge University Press, 2016.

[86] T. Funke and T. Becker. Stochastic block models: A comparison of variants and inference methods. *PloS one*, 14(4):e0215296, 2019.

[87] R. Gallotti, F. Valle, N. Castaldo, P. Sacco, and M. De Domenico. Assessing the risks of 'infodemics' in response to covid-19 epidemics. *Nature Human Behaviour*, 4(12):1285–1293, 2020.

[88] F. Giroire, S. Pérennes, and T. Trolliet. A random growth model with any real or theoretical degree distribution. In *COMPLEX NETWORKS 2020-The 9th International Conference on Complex Networks and their Applications*, 2020.

[89] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[90] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[91] J. P. Gleeson, K. P. O'Sullivan, R. A. Baños, and Y. Moreno. Effects of network structure, competition and memory time on social spreading phenomena. *Physical Review X*, 6(2):021019, 2016.

[92] J. Golbeck, C. Robles, M. Edmondson, and K. Turner. Predicting personality from twitter. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pages 149–156. IEEE, 2011.

[93] S. González-Bailón and M. De Domenico. Bots are less central than verified accounts during contentious political events. *Proceedings of the National Academy of Sciences*, 118(11), 2021.

[94] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. arXiv: 1406.2661.

[95] J. Greenberg. Big metadata, smart metadata, and metadata capital: Toward greater synergy between data science and metadata. *Journal of Data and Information Science*, 2(3):19, 2017.

[96] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[97] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1995.

[98] N. Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

[99] P. Guerra, W. Meira Jr, C. Cardie, and R. Kleinberg. A measure of polarization on social media networks based on community boundaries. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 7, 2013.

[100] S. C. Guntuku, D. Preotiuc-Pietro, J. C. Eichstaedt, and L. H. Ungar. What twitter profile and posted images reveal about depression and anxiety. In *Proceedings of the international AAAI conference on web and social media*, volume 13, pages 236–246, 2019.

[101] B. Guo, Y. Ding, L. Yao, Y. Liang, and Z. Yu. The Future of False Information Detection on Social Media: New Perspectives and Trends. *ACM Computing Surveys*, 53(4):68:1–68:36, July 2020.

[102] W. Guo and A. Caliskan. Detecting Emergent Intersectional Biases: Contextualized Word Embeddings Contain a Distribution of Human-like Biases. *arXiv:2006.03955 [cs]*, July 2020. arXiv: 2006.03955.

[103] M. A. Haidar and M. Rezagholizadeh. TextKD-GAN: Text Generation Using Knowledge Distillation and Generative Adversarial Networks. In M.-J. Meurs and F. Rudzicz, editors, *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 107–118, Cham, 2019. Springer International Publishing.

[104] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[105] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive Representation Learning on Large Graphs. *arXiv:1706.02216 [cs, stat]*, Sept. 2018. arXiv: 1706.02216.

[106] W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.

[107] J. K. Harris. *An introduction to exponential random graph modeling*, volume 173. Sage Publications, 2013.

[108] D. Hazarika, R. Zimmermann, and S. Poria. MISA: Modality-Invariant and -Specific Representations for Multimodal Sentiment Analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, pages 1122–1131, New York, NY, USA, Oct. 2020. Association for Computing Machinery.

[109] L. Hébert-Dufresne, S. V. Scarpino, and J.-G. Young. Macroscopic patterns of interacting contagions are indistinguishable from social reinforcement. *Nature Physics*, 16(4):426–431, 2020.

[110] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos. It's who you know: Graph mining using recursive structural features. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 663–671, New York, NY, USA, 2011. Association for Computing Machinery.

[111] A. D. Henry, P. Prałat, and C.-Q. Zhang. Emergence of segregation in evolving social networks. *Proceedings of the National Academy of Sciences*, 108(21):8605–8610, 2011.

[112] I. Himelboim, S. McCreery, and M. Smith. Birds of a feather tweet together: Integrating network and content analyses to examine cross-ideology exposure on twitter. *Journal of computer-mediated communication*, 18(2):154–174, 2013.

[113] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[114] C. Honey and S. C. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–10. Ieee, 2009.

[115] A. Hoyle, P. Goel, and P. Resnik. Improving Neural Topic Models using Knowledge Distillation. *arXiv:2010.02377 [cs]*, Oct. 2020. arXiv: 2010.02377.

[116] M. Hu, Y. Peng, Z. Huang, D. Li, and Y. Lv. Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification. *arXiv:1906.03820 [cs]*, June 2019. arXiv: 1906.03820.

[117] W. Hu and J. Tsujii. A Latent Concept Topic Model for Robust Topic Inference Using Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 380–386, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[118] Y. Huang, H. Xu, Z. Duan, A. Ren, J. Feng, Q. Zhang, and X. Wang. Modeling complex spatial patterns with temporal features via heterogenous graph embedding networks. *arXiv preprint arXiv:2008.08617*, 2020.

[119] C. Hutto and E. Gilbert. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1), May 2014. Number: 1.

[120] S. Janson, T. Luczak, and A. Rucinski. *Random graphs*, volume 45. John Wiley & Sons, 2011.

[121] A. Javari, T. Derr, P. Esmailian, J. Tang, and K. C.-C. Chang. Rose: Role-based signed network embedding. In *Proceedings of The Web Conference 2020*, pages 2782–2788, 2020.

[122] G. Jawahar, M. Abdul-Mageed, and L. V. S. Lakshmanan. Automatic Detection of Machine Generated Text: A Critical Survey. *arXiv:2011.01314 [cs]*, Nov. 2020. arXiv: 2011.01314.

[123] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao. SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, 2020. arXiv: 1911.03437.

[124] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo. Multimodal Fusion with Recurrent Neural Networks for Rumor Detection on Microblogs. In *Proceedings of the 25th ACM international conference on Multimedia*, MM '17, pages 795–816, New York, NY, USA, Oct. 2017. Association for Computing Machinery.

[125] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian. Novel Visual and Statistical Image Features for Microblogs News Verification. *IEEE Transactions on Multimedia*, 19(3):598–608, Mar. 2017. Conference Name: IEEE Transactions on Multimedia.

[126] N. Johnson, B. Turnbull, T. Maher, and M. Reisslein. Semantically modeling cyber influence campaigns (cics): Ontology model and case studies. *IEEE Access*, 2020.

[127] B. Jou, T. Chen, N. Pappas, M. Redi, M. Topkara, and S.-F. Chang. Visual affect around the world: A large-scale multilingual visual sentiment ontology. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 159–168, 2015.

[128] B. Kamiński, Ł. Kraiński, P. Prałat, and F. Théberge. A multi-purposed unsupervised framework for comparing embeddings of undirected and directed graphs. *Network Science*, 2022. doi.org/10.1017/nws.2022.27.

[129] B. Kamiński, V. Poulin, P. Prałat, P. Szufel, and F. Théberge. Clustering via hypergraph modularity. *PloS one*, 14(11):e0224307, 2019.

[130] B. Kamiński, P. Prałat, and F. Théberge. Community detection algorithm using hypergraph modularity. In *International Conference on Complex Networks and Their Applications*, pages 152–163. Springer, 2020.

[131] B. Kamiński, P. Prałat, and F. Théberge. An unsupervised framework for comparing graph embeddings. *Journal of Complex Networks*, 8(5):cnz043, 2020.

[132] B. Kamiński, P. Prałat, and F. Théberge. Artificial benchmark for community detection (abcd): Fast random graph model with community structure. *Network Science*, 9(2):153–178, 2021.

[133] B. Kamiński, P. Prałat, and F. Théberge. *Mining Complex Networks*. Chapman and Hall/CRC, 2021. doi.org/10.1201/9781003218869.

[134] B. Kamiński, P. Prałat, and F. Théberge. Outliers in the abcd random graph model with community structure (abcd+o). In *Proceedings of the 11th International Conference on Complex Networks and their Applications*, 2022 (in press).

[135] B. Kamiński, T. Olczak, B. Pankratz, P. Prałat, and F. Théberge. Properties and performance of the ABCDe random graph model with community structure. *Big Data Research*, 30, 2022. doi.org/10.1016/j.bdr.2022.100348.

[136] B. Kamiński, B. Pankratz, P. Prałat, and F. Théberge. Modularity of the ABCD random graph model with community structure. *pre-print, arXiv [cs:SI]*, 2022. arxiv.org/abs/2203.01480.

[137] M. Kaneko and D. Bollegala. Gender-preserving Debiasing for Pre-trained Word Embeddings. *arXiv:1906.00742 [cs]*, June 2019. arXiv: 1906.00742.

[138] A. Karimi, L. Rossi, and A. Prati. Improving BERT Performance for Aspect-Based Sentiment Analysis. *arXiv:2010.11731 [cs]*, Mar. 2021. arXiv: 2010.11731.

[139] D. Khattar, J. S. Goud, M. Gupta, and V. Varma. MVAE: Multimodal Variational Autoencoder for Fake News Detection. In *The World Wide Web Conference*, WWW '19, pages 2915–2921, New York, NY, USA, May 2019. Association for Computing Machinery.

[140] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014. arXiv: 1312.6114.

[141] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*, 2017.

[142] H. Kirk, Y. Jun, H. Iqbal, E. Benussi, F. Volpin, F. A. Dreyer, A. Shtedritski, and Y. M. Asano. How True is GPT-2? An Empirical Analysis of Intersectional Occupational Biases. *arXiv:2102.04130 [cs]*, Feb. 2021. arXiv: 2102.04130.

[143] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá. Hyperbolic geometry of complex networks. *Physical Review E.*, 82(3):036106, 2010.

[144] P. N. Krivitsky and M. S. Handcock. A separable model for dynamic networks. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 76(1):29, 2014.

[145] T. Kumar, S. Vaidyanathan, H. Ananthapadmanabhan, S. Parthasarathy, and B. Ravindran. A new measure of modularity in hypergraphs: Theoretical insights and implications for effective clustering. In *International Conference on Complex Networks and Their Applications*, pages 286–297. Springer, 2019.

[146] T. Kumar, S. Vaidyanathan, H. Ananthapadmanabhan, S. Parthasarathy, and B. Ravindran. Hypergraph clustering by iteratively reweighted modularity maximization. *Applied Network Science*, 5(1):1–22, 2020.

[147] M. J. Kusner and J. M. Hernández-Lobato. GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution. *arXiv:1611.04051 [cs, stat]*, Nov. 2016. arXiv: 1611.04051.

[148] S. Kwon, M. Cha, and K. Jung. Rumor Detection over Varying Time Windows. *PLOS ONE*, 12(1):e0168344, 2017. Publisher: Public Library of Science.

[149] A. Lamb, A. Goyal, Y. Zhang, S. Zhang, A. Courville, and Y. Bengio. Professor Forcing: A New Algorithm for Training Recurrent Networks. *arXiv:1610.09038 [cs, stat]*, Oct. 2016. arXiv: 1610.09038.

[150] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.

[151] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[152] J. Lehmann, B. Gonçalves, J. J. Ramasco, and C. Cattuto. Dynamical classes of collective attention in twitter. In *Proceedings of the 21st international conference on World Wide Web*, pages 251–260, 2012.

[153] P. Leifeld, S. J. Cranmer, and B. A. Desmarais. Temporal exponential random graph models with btergm: Estimation and bootstrap confidence intervals. *Journal of Statistical Software*, 83(6), 2018.

[154] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, 2006.

[155] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, 2005.

[156] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007.

[157] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, May 2021.

[158] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma. Topic Modeling for Short Texts with Auxiliary Word Embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, SIGIR '16, pages 165–174, New York, NY, USA, July 2016. Association for Computing Machinery.

[159] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky. Adversarial Learning for Neural Dialogue Generation. *arXiv:1701.06547 [cs]*, Sept. 2017. arXiv: 1701.06547.

[160] X. Li, M. Mobilia, A. M. Rucklidge, and R. Zia. How does homophily shape the topology of a dynamic network? *arXiv preprint arXiv:2106.15963*, 2021.

[161] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What Makes Good In-Context Examples for GPT-3? *arXiv:2101.06804 [cs]*, Jan. 2021. arXiv: 2101.06804.

[162] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 51–62, 2014.

[163] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, July 2019. arXiv: 1907.11692.

[164] Y. Liu and Y.-F. B. Wu. FNED: A Deep Network for Fake News Early Detection on Social Media. *ACM Transactions on Information Systems*, 38(3):25:1–25:33, May 2020.

[165] Y.-J. Lu and C.-T. Li. GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. *arXiv:2004.11648 [cs, stat]*, Apr. 2020. arXiv: 2004.11648.

[166] Z. Lu, J. Wahlström, and A. Nehorai. Community detection in complex networks via clique conductance. *Scientific reports*, 8(1):1–16, 2018.

[167] H. Luo, L. Ji, T. Li, N. Duan, and D. Jiang. Grace: Gradient harmonized and cascaded labeling for aspect-based sentiment analysis, 2020.

[168] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 3818–3824, New York, New York, USA, July 2016. AAAI Press.

[169] T. Manzini, Y. C. Lim, Y. Tsvetkov, and A. W. Black. Black is to Criminal as Caucasian is to Police: Detecting and Removing Multiclass Bias in Word Embeddings. *arXiv:1904.04047 [cs, stat]*, July 2019. arXiv: 1904.04047.

[170] C. J. Matheus, M. M. Kokar, and K. Baclawski. A core ontology for situation awareness. In *Proceedings of the Sixth International Conference on Information Fusion*, volume 1, pages 545–552, 2003.

[171] K. McGuffie and A. Newhouse. The Radicalization Risks of GPT-3 and Advanced Neural Language Models. *arXiv:2009.06807 [cs]*, Sept. 2020. arXiv: 2009.06807.

[172] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[173] F. Meng, M. Medo, and B. Buechel. Whom to trust in a signed network? optimal solution and two heuristic rules. 2021.

[174] Y. Miao, L. Yu, and P. Blunsom. Neural Variational Inference for Text Processing. *arXiv:1511.06038 [cs, stat]*, June 2016. arXiv: 1511.06038.

[175] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013.

[176] S. M. Mohammad and P. D. Turney. Crowdsourcing a word-emotion association lexicon. 29(3):436–465, 2013.

[177] C. E. Moody. Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec. *arXiv:1605.02019 [cs]*, May 2016. arXiv: 1605.02019 version: 1.

[178] A. J. Morales, J. Borondo, J. C. Losada, and R. M. Benito. Efficiency of human activity on information spreading on twitter. *Social networks*, 39:1–11, 2014.

[179] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69(2 Pt 2):026113, 2004.

[180] V.-H. Nguyen, K. Sugiyama, P. Nakov, and M.-Y. Kan. FANG: Leveraging Social Context for Fake News Detection Using Graph Representation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 1165–1174, New York, NY, USA, Oct. 2020. Association for Computing Machinery.

[181] L. Ostroumova, A. Ryabchenko, and E. Samosvat. Generalized preferential attachment: tunable power-law degree distribution and clustering coefficient. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 185–202. Springer, 2013.

[182] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1105–1114, New York, NY, USA, 2016. Association for Computing Machinery.

[183] J. W. Pennebaker, M. E. Francis, and R. J. Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[184] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery.

[185] J. C. Peterson. jcpeterson/openwebtext, June 2021. original-date: 2019-02-18T17:18:16Z.

[186] R. Plutchik. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier, 1980.

[187] V. Poulin and F. Théberge. Ensemble clustering for graphs: comparisons and applications. *Applied Network Science*, 4(51), 2019.

[188] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133:113303, 2020.

[189] V. Pérez-Rosas, R. Mihalcea, and L.-P. Morency. Utterance-Level Multimodal Sentiment Analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 973–982, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics.

[190] D. Quercia, M. Kosinski, D. Stillwell, and J. Crowcroft. Our twitter profiles, our selves: Predicting personality with twitter. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pages 180–185. IEEE, 2011.

[191] N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 76:036106, 10 2007.

[192] S. Rajeswar, S. Subramanian, F. Dutil, C. Pal, and A. Courville. Adversarial Generation of Natural Language. *arXiv:1705.10929 [cs, stat]*, May 2017. arXiv: 1705.10929.

[193] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, 2002.

[194] R. Rietveld, W. van Dolen, M. Mazloom, and M. Worring. What you feel, is what you like influence of message appeals on customer engagement on instagram. *Journal of Interactive Marketing*, 49:20–53, 2020.

[195] M. Rosvall, D. Axelsson, and C. Bergstrom. The map equation. *Eur. Phys. J. Spec. Top.*, 178:13—23, 2009.

[196] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[197] N. Ruchansky, S. Seo, and Y. Liu. CSI: A Hybrid Deep Model for Fake News Detection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806, Nov. 2017. arXiv: 1703.06959.

[198] F. Sala, C. De Sa, A. Gu, and C. Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018.

[199] A. Salfinger, W. Retschitzegger, W. Schwinger, and B. Pröll. Crowd sa—towards adaptive and situation-driven crowd-sensing for disaster situation awareness. In *2015 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision*, pages 14–20. IEEE, 2015.

[200] A. Salfinger, W. Schwinger, W. Retschitzegger, and B. Pröll. Mining the disaster hotspots-situation-adaptive crowd knowledge extraction for crisis management. In *2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pages 212–218. IEEE, 2016.

[201] Z. R. Samani, S. C. Guntuku, M. E. Moghaddam, D. Preoţiuc-Pietro, and L. H. Ungar. Cross-platform and cross-interaction study of user personality based on images on twitter and flickr. *PloS one*, 13(7):e0198660, 2018.

[202] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*, Feb. 2020. arXiv: 1910.01108.

[203] S. E. Schaeffer. Graph clustering. *Comput. Sci. Rev.*, 1(1):27–64, 2007.

[204] T. C. Schelling. Models of segregation. *The American Economic Review*, 59(2):488–493, 1969.

[205] P. Schramowski, C. Turan, N. Andersen, C. Rothkopf, and K. Kersting. Language Models have a Moral Dimension. *arXiv:2103.11790 [cs]*, Mar. 2021. arXiv: 2103.11790.

[206] T. Schuster, R. Schuster, D. J. Shah, and R. Barzilay. The Limitations of Stylometry for Detecting Machine-Generated Fake News. *arXiv:1908.09805 [cs]*, Feb. 2020. arXiv: 1908.09805.

[207] V. Semenova and J. Winkler. Reddit's self-organised bull runs: Social contagion and asset prices. *arXiv preprint arXiv:2104.01847*, 2021.

[208] Y.-D. Seo, Y.-G. Kim, E. Lee, and D.-K. Baik. Personalized recommender system based on friendship strength in social network services. *Expert Systems with Applications*, 69:135–148, 2017.

[209] J. Seymour and P. Tully. Generative Models for Spear Phishing Posts on Social Media. *arXiv:1802.05196 [cs, stat]*, Feb. 2018. arXiv: 1802.05196.

[210] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 92–104. Springer, 2018.

[211] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu. dEFEND: Explainable Fake News Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 395–405, New York, NY, USA, July 2019. Association for Computing Machinery.

[212] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. Fake News Detection on Social Media: A Data Mining Perspective. *arXiv:1708.01967 [cs]*, Sept. 2017. arXiv: 1708.01967.

[213] K. Shu, S. Wang, and H. Liu. Beyond News Contents: The Role of Social Context for Fake News Detection. *arXiv:1712.07709 [cs]*, Dec. 2018. arXiv: 1712.07709.

[214] K. Shu, S. Wang, J. Tang, R. Zafarani, and H. Liu. User identity linkage across online social networks: A review. *Acm Sigkdd Explorations Newsletter*, 18(2):5–17, 2017.

[215] T. A. Snijders. Stochastic actor-oriented models for network dynamics. *Annual Review of Statistics and Its Application*, 4:343–363, 2017.

[216] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics.

[217] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, and J. Wang. Release Strategies and the Social Impacts of Language Models. *arXiv:1908.09203 [cs]*, Nov. 2019. arXiv: 1908.09203.

[218] M. Soleymani, D. Garcia, B. Jou, B. Schuller, S.-F. Chang, and M. Pantic. A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65:3–14, 2017.

[219] W. Song, H. Chen, X. Liu, H. Jiang, and S. Wang. Hyperbolic node embedding for signed networks. *Neurocomputing*, 421:329–339, 2021.

[220] A. Srivastava and C. Sutton. Autoencoding Variational Inference For Topic Models. *arXiv:1703.01488 [stat]*, Mar. 2017. arXiv: 1703.01488.

[221] F. Stahl, M. M. Gaber, and M. Adedoyin-Olowe. A survey of data mining techniques for social media analysis. *Journal of Data Mining & Digital Humanities*, 2014, 2014.

[222] M. Stella, M. Cristoforetti, and M. De Domenico. Influence of augmented humans in online interactions during voting events. *PloS one*, 14(5):e0214210, 2019.

[223] C. Sun, L. Huang, and X. Qiu. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. *arXiv:1903.09588 [cs]*, Mar. 2019. arXiv: 1903.09588.

[224] J. Sun, Z. Cheng, S. Zuberi, F. Perez, and M. Volkovs. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the International World Wide Web Conference*, 2021.

[225] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 121–128. IEEE, 2011.

[226] Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla. When will it happen? relationship prediction in heterogeneous information networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 663–672, 2012.

[227] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models. *arXiv:2102.02503 [cs]*, Feb. 2021. arXiv: 2102.02503.

[228] A. Tandon, A. Albeshri, V. Thayananthan, W. Alhalabi, F. Radicchi, and S. Fortunato. Community detection in networks using graph embeddings. *Physical Review E*, 103(2):022316, 2021.

[229] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 1067–1077, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[230] L. Tang and H. Liu. Relational learning via latent social dimensions. pages 817–826, Jan. 2009.

[231] L. Thompson and D. Mimno. Topic Modeling with Contextualized Word Representation Clusters. *arXiv:2010.12626 [cs]*, Oct. 2020. arXiv: 2010.12626.

[232] V. Traag, L. Waltman, and N. van Eck. From louvain to leiden: guaranteeing well-connected communities. *Sci Rep*, 9, 5233, 2019.

[233] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, Dec. 2017. arXiv: 1706.03762.

[234] N. Veldt, A. Wirth, and D. F. Gleich. Parameterized objectives and algorithms for clustering bipartite graphs and hypergraphs. In *Proceeding of KDD2020*, KDD '20, pages 1868–1876, 2020. Accepted.

[235] D. Wang, P. Cui, and W. Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1225–1234, New York, NY, USA, 2016. Association for Computing Machinery.

[236] S. Wang, C. Aggarwal, J. Tang, and H. Liu. Attributed signed network embedding. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 137–146, 2017.

[237] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu. Signed network embedding in social media. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 327–335. SIAM, 2017.

[238] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao. EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 849–857, New York, NY, USA, July 2018. Association for Computing Machinery.

[239] Z. Wang, Z. Wan, and X. Wan. TransModality: An End2End Fusion Method with Transformer for Multimodal Sentiment Analysis. In *Proceedings of The Web*

Conference 2020, WWW '20, pages 2514–2520, New York, NY, USA, Apr. 2020. Association for Computing Machinery.

[240] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[241] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992.

[242] S. C. Woolley and P. N. Howard. *Computational propaganda: political parties, politicians, and political manipulation on social media*. Oxford University Press, 2018.

[243] F. Wu and B. A. Huberman. Novelty and collective attention. *Proceedings of the National Academy of Sciences*, 104(45):17599–17601, 2007.

[244] L. Wu and H. Liu. Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 637–645, New York, NY, USA, Feb. 2018. Association for Computing Machinery.

[245] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

[246] L. Xing, K. Deng, H. Wu, P. Xie, H. V. Zhao, and F. Gao. A survey of across social networks user identification. *IEEE Access*, 7:137472–137488, 2019.

[247] H. Xu, L. Shu, P. S. Yu, and B. Liu. Understanding Pre-trained BERT for Aspect-based Sentiment Analysis. *arXiv:2011.00169 [cs]*, Oct. 2020. arXiv: 2011.00169.

[248] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[249] Z. Yang, R. Algesheimer, and C. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Sci Rep*, 6, 30750, 2016.

[250] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[251] S. M. Yimam, I. Gurevych, R. E. de Castilho, and C. Biemann. Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, 2013.

[252] L. Yu, W. Zhang, J. Wang, and Y. Yu. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *arXiv:1609.05473 [cs]*, Aug. 2017. arXiv: 1609.05473.

[253] S. Yuan, X. Wu, and Y. Xiang. Sne: signed network embedding. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 183–195. Springer, 2017.

[254] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin. A survey of sentiment analysis in social media. *Knowledge and Information Systems*, 60(2):617–663, 2019.

[255] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency. MOSI: Multimodal Corpus of Sentiment Intensity and Subjectivity Analysis in Online Opinion Videos. *arXiv:1606.06259 [cs]*, Aug. 2016. arXiv: 1606.06259.

[256] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi. Defending Against Neural Fake News. *arXiv:1905.12616 [cs]*, Dec. 2020. arXiv: 1905.12616.

[257] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[258] D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 6(1):3–28, 2018.

[259] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin. Adversarial Feature Matching for Text Generation. *arXiv:1706.03850 [cs, stat]*, Nov. 2017. arXiv: 1706.03850.

[260] Y.-J. Zhang, K.-C. Yang, and F. Radicchi. Systematic comparison of graph embedding methods in practical tasks. *arXiv preprint arXiv:2106.10198*, 2021.

[261] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate Before Use: Improving Few-Shot Performance of Language Models. *arXiv:2102.09690 [cs]*, Feb. 2021. arXiv: 2102.09690.

[262] F. Zhou, L. Liu, K. Zhang, G. Trajcevski, J. Wu, and T. Zhong. Deeplink: A deep learning approach for user identity linkage. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1313–1321. IEEE, 2018.

[263] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[264] X. Zhou, A. Jain, V. V. Phoha, and R. Zafarani. Fake News Early Detection: A Theory-driven Model. *Digital Threats: Research and Practice*, 1(2):12:1–12:25, June 2020.

[265] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *arXiv:1506.06724 [cs]*, June 2015. arXiv: 1506.06724.

[266] M. Zitnik and J. Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):190–198, 2017.