Title

Multilinear (Tensor) Algebra Framework for Misinformation Detection With Limited Supervision

Author

Abdali, Sara

Publication Date

2021

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Multilinear (Tensor) Algebra Framework for Misinformation Detection With Limited Supervision

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Sara Abdali

December 2021

Dissertation Committee:

Dr. Evangelos Papalexakis, Chairperson
Dr. Tamar Shinar
Dr. Eamonn Keogh
Dr. Michalis Faloutsos
Dr. Huan Liu

The Dissertation of Sara Abdali is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

First and foremost, I would like to thank my advisor Professor Evangelos Papalexakis for all the support during the past four years. I would like to thank him for understanding students, not pressuring them and specially for having the freedom to collaborate with many researchers in both industry and academia, without which I would not have been able to broaden my knowledge in different areas of research.

I thank the other members of my Ph.D. thesis committee. Professors Tamar Shinar, Eamonn Keogh, Michalis Faloutsos from UCR and Professor Huan Liu from Arizona State University, for their valuable comments and feedback which improved the quality of this dissertation.

I would like to thank my undergraduate professor Dr. Amin Hassanzadeh and my friend Faegheh Negini for their invaluable help and support during the Ph.D. application process.

During my Ph.D. studies, I was extremely lucky to collaborate with other talented researchers and faculty members at University of California, Riverside (UCR), University of California, Los Angeles (UCLA), Lenovo Research and Microsoft Corporation.

At UCR, I am extremely grateful to Professor Henry Tucker at department of mathematics, for giving me his precious time and for helpful discussions on K-Nearest Hyperplane Graph.

At UCLA, I would like to thank Dr. Alex Vasilescu for multiple discussions on multilinear projection and Deepfake video detection. I will always treasure all I have learned from her.

I am extremely grateful to Dr. Subhabrata Mukherjee a senior researcher at Microsoft Research (MSR) for his genuine help, when I needed it the most. I will always be thankful to him for accepting my invitation to collaborate while I did not have a single paper, scheduling biweekly meetings, and mentoring me on Vec2Node project. He sincerely helped me without expecting

anything in return. I hope I could follow his way of helping others in future.

During the Fall 2020, I was interning at Lenovo research where I had the chance to meet many amazing and talented researchers. Specially, I would like to thank Dr. Hong Zhao for all the support and encouragements during and after my internship. I would like to also thank Dr. Grant LIoyd for all the support and Rachel Graham who surprised me over the phone with the least expected internship offer. I am extremely thankful to Dr. John Pincenti for the fun and educational interview process, all I learned about image processing and for the amazing virtual tour of Chicago! I would like to thank all other group members of Motorola Mobility at Chicago and PC camera group members at North Carolina.

During the last two years of my Ph.D., I was so honored to receive Grace Hopper Celebration Scholarships (GHC) in September 2020 and 2021. I am thankful to AnitaB organization for connecting students like me who do not have any connections with industry. Specially, I am grateful for establishing an industry connection with Microsoft Corporation.

At Microsoft, I am extremely grateful to Anjali Parikh, Hans Wolter and John Mauceri for all their support during and after my internship term. I would like to express my sincere appreciation to Brett Bloomquis for being such a passionate and amazing mentor and friend, for our memorable and fruitful daily brainstorming meetings, for setting me up for success, and for all I have learned from him. I would like to also thank my other teammates at CSX data team, specially I am grateful to Peter Houston, Prachur Bhargava, Ryan Bae, Daniel Huang, Lukas Dauterman, Ankit Srivastava and Ryan Serrao.

During my Ph.D. years, I was so lucky to work with many talented undergraduate and graduate students at UCR. I would like to thank Siddharth Menon, Daniel Fonseca, Baixi Sun and specially Lauren Flemmer for the great experience‘ of working and learning together.

In the last year of my Ph.D., I was so honored to receive a Computing Innovation Fellowship (CIFellowship). I am extremely grateful to the Computing Research Association (CRA), the Computing Community Consortium (CCC) and National Science Foundation (NSF) for all the support I have received from them to pursue fake news detection research as a postdoctoral research associate at Georgia Institute of Technology (Georgia Tech). I am also grateful to professor Srijan Kumar from Georgia Tech for introducing me to this great opportunity and supporting me as a postdoctoral mentor during and after the application process.

I would like to also thank University of Washington (UW) specially Suzzallo and Allen libraries for hosting me and granting me access to UW resources during the Fall 2021. Graduate reading room in the Suzzallo library will always be one of my favorite academic places in the world.

I would like to mention the late Maryam Mirzakhani for inspiring me and being my female role model in science. Whenever I fail in any project, I whisper her quote: *"Of course, the most rewarding part is the* ***Aha moment****, the excitement of discovery and enjoyment of understanding something new—the feeling of being on top of a hill and having a clear view. But most of the time, doing mathematics for me is like being on a long hike with no trail and no end in sight".*

I would like to also mention and remember the late "Dr." Vivian Thomas for inspiring me to stay passionate when going through hardship and to fight discrimination and injustice with grace and dignity.

I would like to thank all my labmates at UCR: Yorgos Tsitsikas, Uday Singh Saini, Rutuja Gurav, Ravdeep Pasricha, Ekta Gujral, William Shiao and Negin Entezari for all the joyful moments we had together. I would like to also thank all the other friends at Riverside, specially Samridhi

Singla and Sara Alaee for always being there for me.

Last but not least, I would like to thank my family, my father Mohammad Abdali for being my first and forever academic role model. He planted the seed of passion for learning in my early childhood. It has been always my dream to become a passionate reader and faculty like him. He has been always the one who calms me down with life lessons in a single sentence. I am also wholeheartedly grateful to my lifelong teacher and friend, my mother Soraya Selahvarzi. My greatest motivator, supporter, and companion during the hardest moments of my Ph.D. and in the entire life. I would like to also thank my dearest sister, my classmate, teacher, friend (you name it!) Masoumeh Abdali and her husband Mojtaba Hosseinzadeh for all their support and love.

The last two years of my Ph.D. coincided with COVID-19 pandemic. while being a worldwide crises and tragedy, it became a period of introspection and growth for me, a turning point in my life to take a leap forward in becoming a better version of myself both in academic and personal life. I would like to thank everyone, specially my family, who supported me during this tough time. The journey of learning and self improvement is indeed a lifelong journey, with no graduation date until the very last day that I am alive.

To my first teacher and my best friend, my mother.

ABSTRACT OF THE DISSERTATION

Multilinear (Tensor) Algebra for Misinformation Detection With Limited Supervision

by

Sara Abdali

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2021
Dr. Evangelos Papalexakis, Chairperson

Identifying misinformation is one of the most challenging problems in today's interconnected world. The vast majority of the state-of-the-art in detecting misinformation are fully supervised, requiring a large number of high-quality human annotations. However, the availability of such annotations cannot be taken for granted, since it is very costly, time-consuming, and challenging to do so in a way that keeps up with the proliferation of misinformation. In this thesis, we are interested in exploring scenarios where the number of annotations is limited. In such scenarios, we leverage a multilinear framework a.k.a. ”tensor” for a variety of modalities which is shown to be interpretable and a proper tool for semi-supervised and unsupervised settings where there is a few or no annotation. In this dissertation, We propose a number of tensor-based techniques, organized in the following parts:

Content-based techniques of misinformation detection. We propose a novel strategy mixing tensor-based content modeling and semi-supervised learning on article embeddings which requires very few labels. Driven by the effectiveness of our tensor embeddings, we propose a novel text augmentation framework i.e., `Vec2Node` leveraging tensor decomposition to generate synthetic samples by exploiting local and global information in text and reducing concept drift. Vec2Node leverages self-training from in-domain unlabeled data augmented with tensorized word embeddings. Finally, we propose a hybrid summarization framework that incorporates both extractive and abstractive techniques for capturing misinformative key phrases.

Ensemble techniques for multi-aspect detection of misinformation. We investigate how to tap into a diverse number of aspects that characterize a news article, can compensate for the lack of labels. We propose two tensor-based techniques for ensemble learning: HiJoD, a 2-level decomposition framework that leverages article content, context of social sharing behaviors, and host website/domain features; and K-Nearest Hyperplane Graph (KNH) which merges the aforementioned aspects to create a higher order graphical representation of articles.

Vision-based techniques for misinformation detection. We propose to use a promising yet neglected feature: the overall look of the domain web page. We propose VizFake which takes screenshots of news articles and leverages a tensor decomposition based semi-supervised classification technique to classify them. Finally, we propose a modified multilinear (tensor) method, a combination of linear and multilinear regressions for presenting manipulated videos. Our method leverages only a handful of frames per video to detect Deepfakes.

Overall, this dissertation is innovating in the field of misinformation detection, empowering work in label scarce settings while leveraging multiple modalities. We envision that the body of work contained in this dissertation will serve as a blueprint for further research in multi-modal label-scarce misinformation detection, in research and in practice.

Contents

List of Figures

Part I

Introduction and Background

1

Introduction

”Misinformation” is false information that spreads unintentionally whereas the term ”Disinformation” refers to false information that malicious users share intentionally and often strategically to affect other users’ behaviours toward social, political, and economic events. False information includes variety of misleading content, anything from rumors and junk science to hate, conspiracy theories and so on and so forth. [1] Regardless of users’ intention, news outlets have been always

[1] In this thesis, regardless of users’ intention, for the simplicity purposes, we refer to all sorts of false news i.e., misinformation and disinformation as ”Misinformation” or ”Fake News” interchangeably.

susceptible to the spread of fake news.

Nowadays, traditional news outlets have been vastly replaced by web-based technologies like social media. In fact, we may consider web-based platforms as the primary news outlets for many users. Spreading fake news on traditional outlets like TV channels or newspapers is extremely risky as it creates lots of legal and public consequences for the publishers. However, unlike the traditional news outlets, spreading misinformation throughout the web is much easier and usually spreads faster by a wider range of users. Thus, due to their public accessibility and ease of use, web-based outlets are immensely vulnerable to spread of fake news.

With that said, misinformation propagation on the web, and especially via social media, is one of the most challenging problems. The spread of misinformation on Twitter during Hurricane Sandy in 2012 [56], the Boston Marathon blasts in 2013 [54] and US Presidential Elections on Facebook in 2016 [142] are some real world examples of misinformation propagation and its consequences which brings the necessity of effective and robust misinformation detection techniques into light more than ever. In next sections, we discuss different types of fake news and then present different areas of fake news research.

1.1 Different types of fake news

In what follows, we briefly describe some of the false information categories proposed by B.S. Detector [22] crowd source fact checkers:

- **Fake News:** Fabricated stories that are intended to prank the public.
- **Satire:** Humorous commentary on current events in the form of fake news.

- **Extreme Bias:** Trafficking in political propaganda and gross distortions of fact.
- **Conspiracy Theory:** Promoting conspiracy theories.
- **Rumor:** Spreading rumors, innuendo, and unverified claims.
- **State News:** Repressive states operating under government sanction.
- **Junk Science:** Promoting pseudoscience, metaphysics, naturalistic fallacies, and other scientifically dubious claims.
- **Hate Group:** Promoting racism, misogyny, homophobia, and other forms of discrimination.
- **Clickbait:** Generating online advertising revenue and rely on sensationalist headlines or eye-catching pictures.

In this thesis, we consider all of the aforementioned categories as misinformative content.

1.2 Fake news in different modalities

Fake news could be spread in any shape or form e.g., text, image or video. In this section, we describe some of the modalities that are commonly targeted by fake news spreaders.

1.2.1 Fake news in text modality

Text modality i.e., article textual content has been always the main target for fake news spreaders. Misinformative textual information could be generated by human or AI based text generators such as deep transformers [191]. To detect misinformative text, researchers usually leverage Natural Language Processing (NLP) techniques to extract linguistic based information like lexical features i.e., character and word level information [77, 101, 123], content-based features such as stylistic, complexity and psychological features [70] and other content based information such as number of nouns, proportion of positive/negative words, article length etc. [62, 70, 131].

In chapter 3, we propose a novel, content-based technique to capture both word level and contextual level information using higher order co-occurrences of the words which is leveraged to extract patterns that disseminate fake news from credible ones.

1.2.2 Fake news in image modality

The majority of work on misinformation detection focus on text modality. However, there are few studies on visual information of articles. For instance, user image is considered as a feature to investigate the credibility of the tweets [59, 124] or visual clustering scores are used to verify microblogs posts [74]. Another example is to use outdated images as a cue for the detection of unmatched text and pictures of rumors [150]. Moreover, there are existing work that classify fake images on Twitter with a characterization analysis to understand the temporal and social reputation of images [57].

In chapter 8., we propose a novel visual cue i.e., overall look of the webpages and develop an image classification technique for classification of unreliable web domains.

1.2.3 Fake news in video modality

Technologies like Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs) embedded in applications like Zao[2], DeepFakes web β[3], Face Swap by Microsoft[4], Deep-

[2] https://www.zaoapp.net/

[3] https://deepfakesweb.com/

[4] https://www.microsoft.com/en-us/garage/profiles/face-swap/

FaceLab[5] etc. have resulted in a broad usage of synthetic media a.k.a. ”Deepfakes”. Due to the potential misuses of such media e.g., fake pornography, fake news, and financial or political fraud, they have become a major public concern. The term Deepfakes has been widely used for deep learning generated media, but it is also the name of a specific manipulation technique in which face of one person is replaced by another one.[6] Other automated manipulation techniques are Face2Face, FaceSwap, NeuralTextures, and more recently FaceShifter [127]. Interested reader is refered to [127] for more details on the aforementioned methods.

Prior Deepfakes detection can be categorized as [152] approaches that classify based on (a) physical or physiological causal factors which are not well presented in Deepfakes e.g., eye blinking [98] and heart rate [67], or (b) artifacts in imaging factors e.g., relative head pose to the camera position [188], and (c) data-driven techniques that do not leverage specific cues and directly train a deep learning model on a large set of real and Deepfake videos [5, 29].

From the first category, we can mention [188], where Yeng et al. propose using the inconsistencies in head poses to detect the Deepfakes. More precisely, 3D head poses cue is leveraged to estimate errors introduced by splicing process which synthesizes source face region into the target one. The eye blinking cue is anther physiological signal which is not well presented in Deepfakes and Li et al. take advantage of it for discriminating the Deepfakes [98]. More recently, a novel cue has been introduced that considers the heart rate measured by remote photoplethysmography (rPPG) to analyze color changes in the human skin, which is a signal for the presence of blood under the tissues [67].

As an example of the second category, we can refer to the work in [99] where the dis-

[5] https://awesomeopensource.com/project/iperov/DeepFaceLab

[6] To distinguish these, we denote said method by DeepFakes in the entire thesis.

tinctive feature is the introduced face warping artifacts. In this work, Li et al. discuss limitation of early Deepfake generators which produce images of limited resolutions and transformation of this images leaves certain distinctive artifacts in the Deepfake videos. In addition, in [107], McCloske et al. analyze the structure of the generator network of a GAN and show how the network's treatment of exposure is markedly different from a real camera. They propose leveraging frequency of over-exposed pixels as a feature for this cue to distinguish GAN-generated media from camera imagery.

However, the vast majority of proposed methods for Deepfake detection fall in the third category, i.e., data-driven approaches. For instance, in [14] a hybrid Long Short Term Memory Network (LSTM) and Encoder-Decoder architecture is introduced to detect forgeries in images. In another work [29], a novel CNN network inspired by inception is introduced, where inception modules have been replaced with depth wise separable convolutions. Another example of this category is the work proposed in [5], where two networks are presented, both with a low number of layers to focus on the mesoscopic properties of the images. [53], [113] and [112] are other instances of data-driven approaches which leverage (Recurrent Neural Networks (RNNs), capsule networks and CNN networks for detecting Deepfakes. Lastly, there are works that take advantage of CNNs and RNNs simultaneously to capture both frame level and sequence level information [14, 53, 114].

In chapter 9., we leverage a novel cue i.e., facial outer ring that we hypothesize to capture the majority of synthesizing artifacts and develop a multilinear framework to classify them.

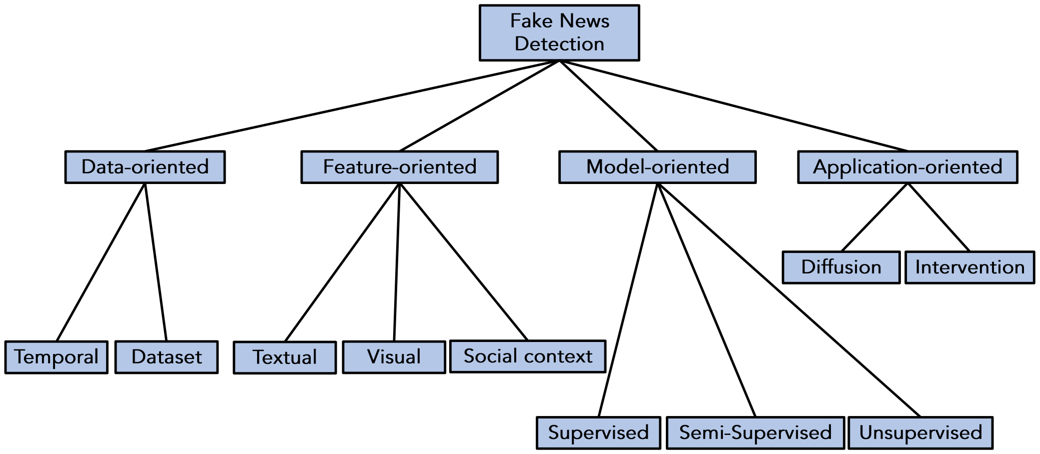


Figure 1.1: An overview of research directions in fake detection detection [137]

1.3 Research directions in fake news detection

As fake news detection is an extremely important and crucial task, researchers have put a lot of effort into studding and developing methods and techniques to address the fake news issue. Fig. 1.1 illustrates different directions of fake news research proposed by [137]. We briefly describe each direction.

Data-oriented Data-oriented fake news detection study refers to techniques that aim to create a large-scale and comprehensive benchmark dataset for fake news detection task which can be leveraged by researchers to push the boundaries of the research in this area. From a temporal perspective, fake news detection research demonstrates unique temporal patterns that distinguish fake news from reliable content.

Feature-oriented Feature-oriented fake news study aim to recognize effective features for discriminating fake news from credible information. These features include news content, social context, user information, linguistic based and visual-based cues etc. NLP baesd features such as word co-occurrences embedding as well as deep neural networks that extract textual or visual features [74, 176, 177] are some popular examples of this category. A category of Deepfake video detection methods that leverage physiological cues [67, 98, 188] or artifacts in imaging factors [99] are also under the umbrella of this category.

Model-oriented Model-oriented fake news research focuses on developing effective and robust machine learning models for fake news detection. This category includes, supervised, semi-supervised and unsupervised approaches.

The majority of existing misinformation detection work leverage supervised classifiers [70, 131]. The main issue with the supervised methods is that they often require a considerable amount of labeled data known as ground truth for training. In reality, these labels are very limited and insufficient. Although there are a couple of fact checking websites such as PolitiFact, FactCheck, Snopes and so on and so forth, they all require human expert for fact checking which is often a costly and time consuming process. There are fewer existing work that leverage semi-supervised or unsupervised techniques for misinformation detection [52, 71].

In contrast to the aforementioned works, in this thesis, we leverage techniques such as semi-supervised propagation, few shot learning, self-training etc. and develop novel augmentation techniques, ensemble and semi-supervised methods to address the problem of label scarcity.

Application-oriented Application-oriented fake news research mostly covers research areas such as fake news diffusion and intervention research. Fake news diffusion research aims to show that credible information and misinformation follow different patterns of social, life cycle, spreader identification etc. when propagate on social media [26,135]. Fake news intervention research, refers to reducing the effects of fake news by proactive intervention methods that try to minimize the scope of spread or reactive intervention methods which are leveraged after fake news goes viral [1].

1.4 Contributions

In this thesis, we leverage tensor algebra to develop multiple solutions for misinformation detection in label scare settings i.e., unsupervised and semi-supervised scenarios. Tensor provides interpretebility which is an urgent focus in AI-based research. Leveraging tensor algebra, we propose multiple solutions including semi-supervised, few-shot learning, ensemble methods and other NLP based techniques such as data augmentation to address the problem of label scarcity.

In this thesis, we contribute to different directions of fake news research i.e., data-oriented, feature-oriented, model-oriented and application-oriented studies. Moreover, we propose novel approaches for fake news detection in different modalities i.e., text, image and video. Summarily, the main contributions of this thesis are as follows:

- We contribute to the data-oriented fake news research by creating a multi-class and large-scale dataset comprising more than a 1M tweets and more than 400K articles each of which labeled as one of the categories of fake news we introduced earlier . We also create another dataset including the screenshots of these articles for visual-based studies. Interested reader is referred to chapter 3 and chapter 8 respectively.

- We contribute to the feature-oriented study by proposing novel features for different modalities. In chapter 3, we propose higher order co-occurrences of the words as a textual feature that discriminates fake news from real ones. In chapter 8, we propose overall look of the domain webpage, as a cue for evaluating the credibility of news articles shared by them. In chapter 9, we propose the facial outer-ring as a reign that comprises high concentration of synthesizing artifacts.

- We extensively contribute to the model-oriented study by developing novel semi-supervised content-based, ensemble-based and visual-based models as well as a supervised video-based method. More specifically,in part II, chapter 3, we propose a novel semi-supervised technique for content-based detection of misinformation. Moreover, to compensate for the lack of labels, we propose a novel augmentation method that leverages few-shot leaning and self training techniques. In part III, we propose two novel semi-supervised ensemble techniques for misinformation detection. In part IV, we propose a novel semi-supervised technique for image classification task which achieves the state-of-the-art accuracy, while being an order of magnitude faster. In chapter 9 of this part, we propose a novel supervised technique which is fast, interpretable and achieves admissible accuracy while using only a handful of frames per video.

- We contribute to the diffusion application-oriented research by proposing to decompose different features i.e., co-occurrences, overall visual looks, facial outer etc. into latent patterns that discriminate fake news from reliable information. As far as the intervention fake news research is concerned, in chapter 5, we leverage summarization and sentence ranking techniques to extract key phrases related to each category of fake news which could be leveraged by fact checkers to annotate similar content e.g., hate or bias languages and highlight misinformative parts to bring awareness to the users.

The rest of this thesis consists of four parts and 9 chapters and is organized as follows: Part I, chapter 2, describes the mathematical and machine learning backgrounds we leverage throughout the thesis. In part II, we introduce our novel content-based methods and novel NLP based techniques. More precisely, in chapter 3, we propose our tensor-based semi-supervised framework and in chapters 4 and 5, we introduce a tensor-based augmentation technique and a hybrid summarization approach for key phrases extraction respectively. In part III, we propose two tensor-based ensemble techniques i.e., `HiJoD` in chapter 6 and `KNH` in chapter 7. In part IV, we discuss vision based misinformation and our proposed techniques for detecting them using visual cues. To this end, first in chapter 8, we propose `VizFake`, an image-based technique for finding misinformation and then in chapter 9, we describe our multilinear franework for detecting Deepfake videos. Finally, we present conclusions and future work.

2

Background

In this chapter we present the mathematical background i.e., multilinear algebra, Euclidean geometry and other machine learning techniques we leverage in the proposed methods of this thesis. In the entire thesis we follow the notation of Table 2.1.

2.1 Relevant linear algebra

Definition 1 ***(Matrix Rank)****The rank of a matrix* $\mathbf{X} \in \mathbb{R}^{\mathrm{I_1 \times I_2}}$ *denoted by rank(A) is the maximum number of linearly independent columns (rows). The rank is bounded by the minimum of the matrix dimensions i.e., rank(*$\mathbf{X}$*)* $\leq$ *min(*I_1, I_2*).*

Definition 2 ***(Rank-1 Matrix)****$\mathbf{A}$ is a rank-1 matrix, i.e., rank(A) = 1, if we can decompose it into an outer product, of two vectors* $\mathbf{u} = [u_1, u_2, \ldots, u_{I_1}]$ *and* $\mathbf{v} = [v_1, v_2, \ldots, v_{I_2}]^T$*:*

Definition 3 ***(Rank-R Decomposition)*** *The rank-R decomposition of a matrix* $\mathbf{X}$ *is the minimum number of rank-1 matrices whose linear combination results in* $\mathbf{X}$ *as follows:*

$$\mathbf{X} \simeq \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r \tag{2.1}$$

2.1.1 Singular Value Decomposition (SVD) and Principle Components Analysis (PCA)

In linear algebra, we factorize a matrix $\mathbf{D} \in \mathbb{R}^{I_1 \times I_2}$ using Singular Value Decomposition (SVD) as follows:

$$\mathbf{X} = \mathbf{U\Sigma V}^{\mathrm{T}} \tag{2.2}$$

where the columns of $\mathbf{U} \in \mathbb{R}^{I_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{I_2 \times r}$ are orthonormal and $\mathbf{\Sigma} \in \mathbb{R}^{\mathbf{r \times r}}$ is a diagonal matrix with positive real entries know as singular values. where the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_R > 0$.

Rewriting equation 2.2 in conventional linear algebra, the Principal Components Analysis (PCA) is

$$\mathbf{X} = \underbrace{\mathbf{U}}_{\text{Basis}} \underbrace{\mathbf{\Sigma V}^T}_{\text{Coefficient}} \tag{2.3}$$

Symbol	Definition
$\mathcal{X}$,$\mathbf{X}$,$\mathbf{x}$,x	Tensor,Matrix,Vector,Scaler
$\circ$	Outer product
$\times$	Cross product
$\otimes$	Kronecker product
$\odot$	Khatri-Rao
$\circledast$	Hadamard product
$\mathcal{X}^{\dagger_m}$	Mode-m tensor pseudo-inverse of $\mathcal{X}$
$\mathbf{X}_{[m]}$	Mode-m tensor matrixizing
$\times_\mathrm{m}$	Mode-m product
$\mathrm{Cov}(\mathbf{x},\mathbf{y})$	Covariance $\mathbf{x}$ and $\mathbf{y}$
$\mathrm{E}(\mathbf{x})$	Mean $\mathbf{x}$
ρ,$\mathrm{Corr}(\mathbf{x},\mathbf{y})$	Correlation between $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{C_{xx}}$	Variance matrix of vector$\mathbf{x}$
$\mathbf{C_{xy}}$	Covariance matrix of and $\mathbf{y}$
$\mathcal{C}_{\mathbf{12\cdots m}}$	Covariance Tensor
$\mathbf{h}_x$	Canonical vector
$\mathbf{z}_x$	Canonical variable

Table 2.1: Symbols and Definitions

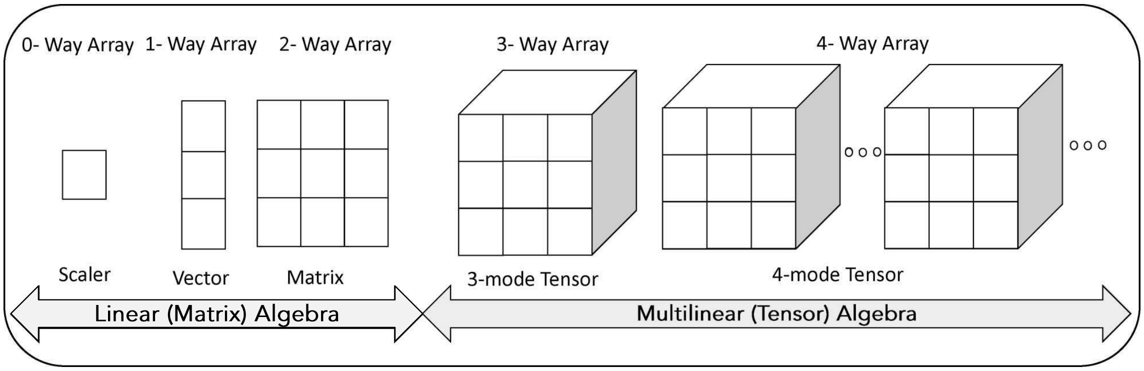


Figure 2.1: Linear algebra vs. multilinear algebra.

2.2 Multilinear (tensor) algebra

2.2.1 Multilinear (tensor) framework

Definition 4 *(**Tensor**) A tensor $\mathcal{X} \in \mathbb{R}^{\mathbf{I_1 \times I_2 \times \ldots \times I_M}}$ is a multi-way array. In other words, a tensor is an array with three or more than three dimensions. The dimensions of a tensor are usually referred to as modes [33, 87]. Figure 2.1 demonstrates matrix and tensor algebra.*

Definition 5 *(**Rank-1 Tensor**) A mode-M tensor $\mathcal{X} \in \mathbb{R}^{\mathbf{I_1 \times I_2 \times \ldots \times I_M}}$ is a rank-1 tensor when it is* *decomposable into outer product of M vectors as follows:*

$$\mathcal{X} = \mathbf{u_1} \circ \mathbf{u_2} \circ \ldots \mathbf{u_M} \tag{2.4}$$

2.2.2 Mode-M matrixizing a tensor

The Mode-m matrixizing of tensor $\mathcal{X} \in \mathbb{R}^{\mathbf{I_1 \times I_2 \times \ldots \times I_M}}$ is defined as the matrix $\mathbf{X}_{[m]} \in \mathbb{R}^{I_m \times (I_1 \ldots I_{m-1} I_{m+1} \ldots I_M)}$ where the parenthetical ordering indicates that column vectors are ordered by sweeping indices of all other modes through their ranges. Therefore:

$$\begin{gathered}[\mathbf{X}]_{jk} = a_{i_1 \ldots i_m \ldots i_M} \text{ where} \\ j = i_m \text{ and } k = 1 + \sum_{n=0, n \neq m}^{M} (i_n - 1) \prod_{l=0, l \neq m}^{n-1} I_l\end{gathered} \tag{2.5}$$

A 3-mode tensor may be matrixized threee different ways by stacking first, second and third mode slices which are illustrated in Figure.2.2 respectively [87, 118, 141].

2.2.3 Mode-M product of a matrix and a tensor

The mode-m product of tensor $\mathcal{X} \in \mathbb{R}^{\mathbf{I_1 \times I_2 \times \ldots I_m \times \ldots \times I_M}}$ and matrix $\mathbf{A} \in \mathbb{R}^{J_m \times I_m}$ denoted by $\mathcal{X} \times_{\mathrm{m}} \mathbf{A}$ is a tensor of size $\mathbb{R}^{I_1 \times I_2 \times \ldots J_m \times \ldots \times I_M}$ where the entries are calculated as: [87, 118, 141]:

$$[\mathcal{X} \times_{\mathrm{m}} \mathbf{A}]_{\mathbf{i_1 \ldots i_{m-1} j_m i_{m+1} \ldots i_M}} = \sum_{\mathbf{i_m}} \mathbf{d_{i_1 i_2 \ldots i_{m-1} i_m i_{m+1} \ldots i_M} a_{j_m i_m}} \tag{2.6}$$

The mode-M product is interchangeably denoted by matrix multiplication and tensor multiplication as follows

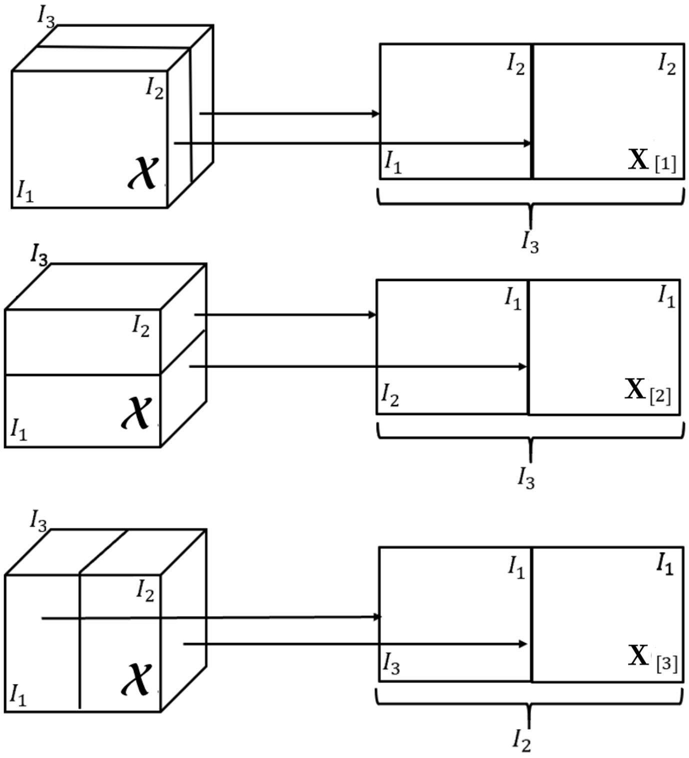


Figure 2.2: Matrixizing a 3-mode tensor

$$\mathcal{B} = \mathcal{X} \times_{\mathrm{m}} \mathbf{A} \underset{\text{tensorizing}}{\overset{\text{matrixizing}}{\rightleftarrows}} \mathbf{B}_{[m]} = \mathbf{A}\mathbf{X}_{[m]} \tag{2.7}$$

Definition 6 *(**Kronecker Product,** $\otimes$) The Kronecker product of* $\mathbf{U} \in \mathbb{R}^{I \times J}$ *and* $\mathbf{V} \in \mathbb{R}^{K \times L}$ *and is the following matrix of size* $IJ \times KL$*:*

$$\mathbf{U} \otimes \mathbf{V} = \begin{bmatrix} \mathbf{u}_{11}\mathbf{V} & \dots & \mathbf{u}_{1J}\mathbf{V} \\ \vdots & \ddots & \vdots \\ \mathbf{u}_{I1}\mathbf{V} & \dots & \mathbf{u}_{IJ}\mathbf{V} \end{bmatrix} \tag{2.8}$$

Definition 7 *(**Khatri-Rao Product,** $\odot$) The Khatri-Rao product of* $\mathbf{U} \in \mathbb{R}^{I \times J}$ *and* $\mathbf{V} \in \mathbb{R}^{K \times L}$ *is a* *columnwise Kronecker product. In fact, the entries of* $\mathbf{U} \odot \mathbf{V}$ *are expressed as follows:*

$$\mathbf{U} \odot \mathbf{V} = [(\mathbf{u}^{(1)} \otimes \mathbf{v}^{(1)}) \dots (\mathbf{u}^{(l)} \otimes \mathbf{v}^{(l)}) \dots (\mathbf{u}^{(L)} \otimes \mathbf{v}^{(L)})] \tag{2.9}$$

In other words, $[\mathbf{U} \odot \mathbf{V}]_{ik,l} = \mathbf{u}_{il}\mathbf{v}_{kl}$. *The Kahtri-Rao product of a set of matrices* $\mathbf{U}_m \in \mathbb{R}^{I_m \times L}$ *for* $1 \leq m \leq M$ *is denoted as:*

$$\mathbf{U}_1 \odot \dots \mathbf{U}_M = [(\mathbf{u}_1^{(1)} \otimes \dots \otimes u_1^{(1)}) \dots (\mathbf{u}_M^{(L)} \otimes \dots \otimes \mathbf{u}_M^{(L)})] \tag{2.10}$$

Where $\mathbf{u}_m^{(l)}$ *is the* l^{th} *column of* $\mathbf{U}_m$ *for* $1 \leq l \leq L$.

Definition 8 ***(Hadamard Product,*** $\circledast$***)*** *Hadamard product of* $\mathbf{U} and \mathbf{V} \in \mathbb{R}^{I \times J}$ *is an element-wise product of* $\mathbf{U}$ *and* $\mathbf{V}$. *In other words,* $[\mathbf{U} \circledast \mathbf{V}]_{ij} = \mathbf{u}_{ij}\mathbf{v}_{ij}$.

The Hadamard product is useful in the computation of the rank-K CP/PARAFAC decomposition of a tensor. The Hadamard product is related to the Khatri-Rao product as follows:

$$(\mathbf{U} \odot \mathbf{V})^T(\mathbf{U} \odot \mathbf{V}) = \begin{bmatrix} \mathbf{U_1}^T \otimes \mathbf{V_1}^T \\ \vdots \\ \mathbf{U_L}^T \otimes \mathbf{V_L}^T \end{bmatrix} \begin{bmatrix} \mathbf{u_1} \otimes \mathbf{v_1} & \dots & \mathbf{u_L} \otimes \mathbf{v_L} \end{bmatrix} \tag{2.11}$$

$$= \begin{bmatrix} \mathbf{u_1}^T\mathbf{u_1} \otimes \mathbf{v_1}^T\mathbf{v_1} & \dots & \mathbf{u_L}^T\mathbf{u_L} \otimes \mathbf{v_1}^T\mathbf{v_L} \\ \vdots & \ddots & \vdots \\ \mathbf{u_L}^T\mathbf{u_1} \otimes \mathbf{v_L}^T\mathbf{v_1} & \dots & \mathbf{u_L}^T\mathbf{u_L} \otimes \mathbf{v_L}^T\mathbf{v_L} \end{bmatrix} \tag{2.12}$$

$$= (\mathbf{U}^T\mathbf{U} \circledast \mathbf{V}^T\mathbf{V}) \tag{2.13}$$

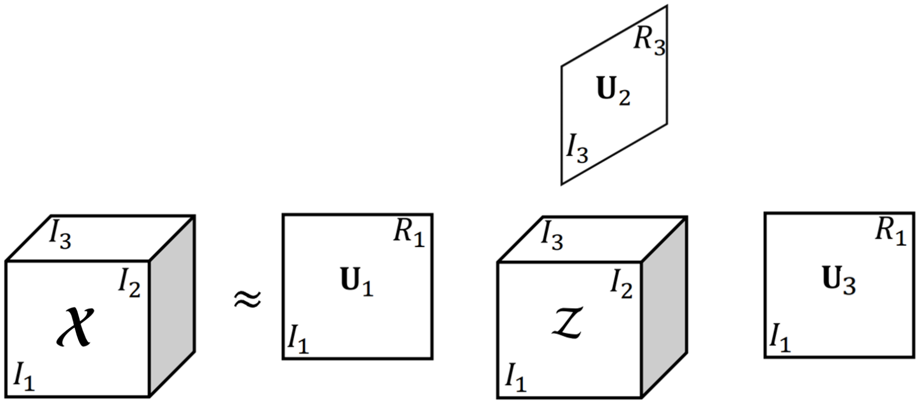


Figure 2.3: M-mode SVD decomposition of a 3-mode tensor

The Kahtri-Rao product of a set of matrices $\mathbf{U}_m \in \mathbb{R}^{I \times J}$ for $1 \leq m \leq M$ could be expressed as:

$$(\mathbf{U}_1 \odot \ldots \odot \mathbf{U}_M)^T (\mathbf{U}_1 \odot \ldots \odot \mathbf{U}_M) = (\mathbf{U}_1^T \mathbf{U}_1) \circledast \ldots \circledast (\mathbf{U}_M^T \mathbf{U}_M) \tag{2.14}$$

2.2.4 Multilinear SVD

In linear algebra, we can define the SVD in terms of n-mode product as follows:

$$\mathbf{D} = \mathbf{\Sigma} \times_1 \mathbf{U} \times_2 \mathbf{V} \tag{2.15}$$

In multilinear algebra there is a generalization of SVD know as multilinear SVD or M-mode SVD which decomposes an M-mode tensor $\mathcal{X}$ into the M-mode product of orthonormal spaces as discussed in

$$\mathcal{X} \simeq \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_M \mathbf{U}_M \tag{2.16}$$

where $\mathcal{Z}$ is the core tensor that governs the interaction between the orthonormal mode matrices, $\mathbf{U}_m$.

Algorithm 1 M-mode SVD

```
Input :$\mathcal{X} \in \mathbb{R}^{\mathbf{I_1 \times I_2 \times \ldots \times I_M}}$
Output: Mode matrices $\mathbf{U}_1 \times \mathbf{U}_M$ and the core tensor $\mathcal{Z}$.
for all $m, m = 1, \ldots, M$ do
    Let $\mathbf{U_m}$ be the left orthonormal matrix of the SVD of $\mathbf{X}_{[m]}$ , the mode-m matrixized $\mathcal{X}$.
    $\mathcal{Z} = \mathcal{X} \times_1 \mathbf{U_1^T} \times_2 \mathbf{U_2^T} \ldots \times_M \mathbf{U_M^T}$
end
```

The core tensor is analogues to the singular value matrix Σ but unlike the Σ the core tensor is not always diagonal [87, 118, 141].

The M-mode SVD of a 3-mode tensor is demonstrated in Figure 9.3. $\mathbf{U}_i$ is approximated by left singular vectors of truncated SVD decomposition of $\mathbf{D}_{[i]}$. In addition, since $\mathbf{U}_i$ is orthonoramal, we have $\mathbf{U}_m^{-1} = \mathbf{U}_m^T$ and the core tensor $\mathcal{Z}$ is estimated as follows:

$$\mathcal{Z} = \mathcal{X} \times_1 \mathbf{U_1^{-1}} \times_2 \mathbf{U_2^{-1}} \ldots \times_M \mathbf{U_M^{-1}} \tag{2.17}$$

$$= \mathcal{X} \times_1 \mathbf{U_1^T} \times_2 \mathbf{U_2^T} \ldots \times_M \mathbf{U_M^T} \tag{2.18}$$

More details on M-mode SVD is demonstrated in Algorithm 1.

2.3 Canonical Polyadic (CP) or PARAFAC Decomposition

The Canonical Polyadic (CP) or PARAFAC decomposition is an extension of SVD for higher multiway arrays i.e. tensors [118, 141] where the core tensor in this case is an identity tensor i.e., a diagonal tensor of ones along its main diagonal. Indeed, CP/PARAFAC factorizes a tensor into sum of R rank-1 tensors. For instance, decomposition of a 3-mode tensor is as follows:

$$\mathcal{X} \simeq \Sigma_{r=1}^{R} \mathbf{u}_{1r} \circ \mathbf{u}_{2r} \circ \mathbf{u}_{3r} \tag{2.19}$$

Algorithm 2 Canonical Polyadic(CP/PARAFAC) Decomposition

```
Input : $\mathcal{X} \in \mathbb{R}^{\mathbf{I_1 \times I_2 \times \ldots \times I_M}}$ and a desired R.
Output : Converged factor matrices $\mathbf{U_1}, \ldots, \mathbf{U_M}$.
// Random initialization:
for all $m, m = 1, \ldots, M$ do
    Set $\mathbf{U_i}$ to a random $I_m \times R$ matrix.
end
// Local optimization:
for all $n, n = 1, \ldots, N$ until convergence do
    for all $m, m = 1, \ldots, M$ do
        $\mathbf{U_m} = \mathbf{X}_{[m]}(\mathbf{U_M} \odot \ldots \odot \mathbf{U_{m+1}} \odot \mathbf{U_{m-1}} \odot \ldots \odot \mathbf{U_1})(\mathbf{U_1}^T\mathbf{U_1} \circledast \ldots \circledast \mathbf{U_{m-1}}^T\mathbf{U_{m-1}} \circledast \mathbf{U_{m+1}}^T\mathbf{U_{m+1}} \circledast \ldots \circledast \mathbf{U_M}^T\mathbf{U_M})^{-1}$
    end
end
```

where $\mathbf{u}_{1r} \in \mathbb{R}^I$, $\mathbf{u}_{2r} \in \mathbb{R}^J$, $\mathbf{u}_{3r} \in \mathbb{R}^K$ and factor matrices are defined as $\mathbf{U}_1 = [\mathbf{u}_{11}\ \mathbf{u}_{12} \ldots \mathbf{u}_{1R}]$, $\mathbf{U}_2 = [\mathbf{u}_{21}\ \mathbf{u}_{22} \ldots \mathbf{u}_{2R}]$, and $\mathbf{U}_3 = [\mathbf{u}_{31}\ \mathbf{u}_{32} \ldots \mathbf{u}_{3R}]$ where R is the rank of decomposition or the number of columns in the factor matrices. The optimization problem for finding factor matrices is as follows:

$$\min_{\mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3} = \|\mathcal{X} - \Sigma_{r=1}^{R}\mathbf{u}_{1r} \circ \mathbf{u}_{2r} \circ \mathbf{u}_{3r}\|^2 \tag{2.20}$$

To solve the optimization problem above we can use Alternating Least Squares (ALS) method which solves for any of each factor matrix by fixing the others [118, 141]. Algorithm 2 illustrates the details of ALS algorithm.

2.4 Canonical Correlation Analysis (CCA)

In 2-dimensional space the correlation between two vectors x, y is defined as follows [96, 186]:

$$\rho = Corr(\mathbf{x}, \mathbf{y}) = \frac{cov(\mathbf{x}, \mathbf{y})}{\sigma_x \sigma_y} \tag{2.21}$$

Since $Cov(\mathbf{x}, \mathbf{y}) = E(\mathbf{xy}) - E(\mathbf{x})E(\mathbf{y}) = E(\mathbf{xy})$ [31], if the vectors are centered around the mean, then $E(\mathbf{x})$ and $E(\mathbf{y})$ are equal to zero and ρ is going to be [186]:

$$\rho = \frac{E(\mathbf{xy})}{\sqrt{E(\mathbf{x}^2)E(\mathbf{y}^2)}} \tag{2.22}$$

There is a technique known as Canonical Correlation Analysis or CCA where canonical vectors $\mathbf{h_x}, \mathbf{h_y}$ are found such that by projecting vectors $\mathbf{x}$ and $\mathbf{y}$ into canonical variables $\mathbf{z_x}, \mathbf{z_y}$ using these two vectors, the correlation between $\mathbf{z_x}$ and $\mathbf{z_y}$ is maximized [96, 186]:

$$\text{argmax}_{\rho_{\mathbf{z_1},\mathbf{z_2}}} = \frac{E(\mathbf{h_x}^T \mathbf{xy}^{\mathbf{T}} \mathbf{h_y})}{\sqrt{E(\mathbf{h_x}^T \mathbf{xx}^{\mathbf{T}} \mathbf{h_x}) E(\mathbf{h_y}^T \mathbf{yy}^T \mathbf{h_y})}} = \frac{\mathbf{h_x}^T \mathbf{C_{xy}}}{\sqrt{\mathbf{h_x}^T \mathbf{C}_{\mathbf{xx}}^T \mathbf{h_x} \mathbf{h_y}^T \mathbf{C_{yy}} \mathbf{h_y}}} \tag{2.23}$$

Where $\mathbf{C_{xx}} = \mathbf{xx}^T$,$\mathbf{C_{yy}} = \mathbf{YY}^T$ are variance matrices and $\mathbf{C_{xy}} = \mathbf{XY}^T$ is covariance matrix of $\mathbf{x}$ and $\mathbf{y}$.

2.4.1 Tensor Canonical Correlation Analysis (TCCA)

When there are more than two variables, we can define the optimization problem above as a minimization problem where we aim at minimizing the pairwise distance between the variables. So, the generalized form of the CCA is defined as follows [186]:

$$\text{argmin}_{\mathbf{h_p},\mathbf{h_q}} \frac{1}{2m(m-1)} \Sigma_{p,q=1}^{m} \|\mathbf{x_p}^T \mathbf{h_p} - \mathbf{x_q}^T \mathbf{h_q}\|^2 \tag{2.24}$$

Since $\mathbf{C_{xy}} = \mathbf{XY}^T$ is the covariance matrix of $\mathbf{x}$ and $\mathbf{y}$, in higher dimensional space the variance matrix $\mathbf{C}_{pp}$ and covariance tensor $\mathcal{C}_{\mathbf{1}\cdots\mathbf{m}}$ can be defined as [186]:

$$\mathbf{C}_{pp} = \frac{1}{m} \Sigma_{n=1}^{m} \mathbf{x}_{pn} \mathbf{x}_{pn}^T \quad p \in 1 \ldots m \tag{2.25}$$

$$\mathcal{C}_{\mathbf{12}\cdots\mathbf{m}} = \frac{\mathbf{1}}{\mathbf{m}} \boldsymbol{\Sigma}_{\mathbf{n=1}}^{\mathbf{m}} \mathbf{x_{1n}} \circ \mathbf{x_{2n}} \circ \cdots \circ \mathbf{x_{mn}} \tag{2.26}$$

In [186], it has been shown that higher order canonical correlation can be solved by CP/ALS.

2.5 Hyperplanes and flats in n-dimensional space

A hyperplane in an n-dimensional space V is an $n-1$ dimensional subspace which is defined by following linear equation [16]:

$$a_1(x_1 - x_1{}') + a_2(x_2 - x_2{}') + \cdots + a_n(x_n - x_n{}') = 0 \tag{2.27}$$

Where the vector $(a_1, a_2, \ldots, a_n)$ is a normal vector perpendicular to the hyperplane and $(x_1{}', x_2{}', \ldots, x_n{}')$ is a point on the hyperplane. Therefore, we can rewrite the linear equation of hyperplane as [16]:

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = d \tag{2.28}$$

Given n datapoints a hyperplane is uniquely defined in an n-dimensional space. The distance from a point $(x_1{}', x_2{}', \ldots, x_n{}')$ to a hyperplane is equal to [16]:

$$d_{point-hyperplane} = \frac{|a_1 x_1{}' + a_2 x_2{}' + \cdots + a_n x_n{}' + d|}{\sqrt{a_1{}^2 + a_2{}^2 + \cdots + a_n{}^2}} \tag{2.29}$$

A flat or a Euclidean subspace is any lower dimension subspace in that space. For instance, flats in 4-dimensional space are points, lines, and planes. We can described a flat in n-dimensional space by a system of linear parametric equations. For example, the equation of a line in n-dimensional space is equal to:

$$x_1 = a_1 t + b_1, x_2 = a_2 t + b_2, \ldots, x_n = a_n t + b_n \tag{2.30}$$

We can use the Euclidean distance to calculate the distance from a point to a 2-flat (line). For instance, the distance from point P_0 to a 2-flat (line) defined by 2 points P_1 and P_2 in 3-dimensional space is:

$$d_{point-line} = \frac{|(p_2 - p_0) \times (p_1 - p_0)|}{|p_2 - p_1|} \tag{2.31}$$

And a 3-flat (plane) in n-dimensional space is equal to:

$$x_1 = a_1 t_1 + b_1 t_2 + c_1, x_2 = a_2 t_1 + b_2 t_2 + c_2, \ldots, x_n = a_n t_1 + b_n t_2 + c_n \tag{2.32}$$

2.6 Tensor-based KNN graph

A k-nearest-neighbor (KNN) graph is a model for representing the nodes in a given feature space such that the k most similar nodes are connected with edges, weighted by a similarity measure [61].

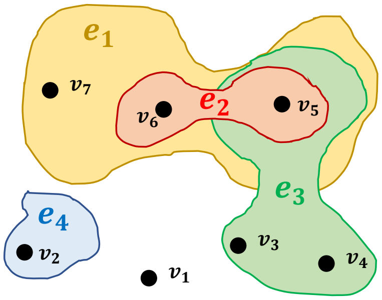


Figure 2.4: An example of a hypergraoh with vertices $v_i, 1 \leq i \leq 7$ and edges $e_j, 1 \leq j \leq 4$
.

In this work, we use tensor representations for each entities i.e., articles as nodes in the embedding space and then we measure the similarity of the nodes using the Euclidean distance between the corresponding vectors.

2.7 Hypergraph

Hypergraphs [50, 194] are an extension of graph models where an edge may connect more than two nodes to illustrate higher-order relationships between the nodes. In contrast to a single weighted connection in traditional graphs, an edge in a hypergraph is a subset of nodes that are similar in terms of features or distance. Figure 2.4 illustrates a hypergraph with vertices $v_i, 1 \leq i \leq 7$ and edges $e_j, 1 \leq j \leq 4$.

Hypergraph Learning

Hypergraph learning has been used for a variety of machine learning applications. For instance, in [190] Yu et al. propose to model an image as a hypergraph that uses hyperedges to capture the contextual features of the pixels. In another work, Lian et al. construct a hypergraph to exploit the correlation information among labels for multi label classification task [149]. In [95] Yu et al. propose an adaptive hypergraph based method for classification of images. Moreover, there are previous works that leverage hypergraphs for object detection tasks [95, 148]. In hypergraphs, the main goal is to define hyperedges and the weights to model high order relationships. Although there are some unsupervised work using affinities within the hyperedges, [72], hypergraphs, do not have exploratory capabilities to define a common "feature space" to predict behaviour of arriving data points (nodes) or predicting the missing data points using this common space. Moreover, finding the weights for the hyperedges is a challenging task and requires complicated optimization and regularization techniques like graph laplacian, $\mathbf{L}_1$ and $\mathbf{L}_2$ regularizers [95, 175].

2.8 Belief propagation

Belief propagation is a message passing algorithm usually applied for calculation of marginal distribution on graph based models such as Markov or Bayesian networks. Several different versions for belief propagation have been introduced each of which for a different graphical model. The iterative message passing mechanism throughout the network, is a common function used for all different versions of belief propagation because the operative intuition behind this algorithm is that the nodes which are "close" are more likely to have similar values known as "belief". Suppose $m_{j\hookrightarrow i}(x_i)$ denote the message passes from node i to node j. $m_{j\hookrightarrow i}(x_i)$ conveys the opinion of node i about the belief of node j. Each node of a given graph G uses the messages received from neighboring nodes to compute its belief iteratively as follows:

$$b_i(x_i) \propto \prod_{j\in(N_i)} m_{j\hookrightarrow i}(x_i) \tag{2.33}$$

Where N_i denotes all the neighboring nodes of node i [20, 189].

In this work, we define the belief as the label of a news article and given a set of known labels, we use FaBP as a means to propagate label likelihood over the nearest neighbor graph. Fast Belief Propagation (FaBP) [89] is a fast and linearized guilt-by-association method, which improves the basic idea of belief propagation (BP) we discussed. The FaBP algorithm solves the following linear system:

$$[\mathbf{I} + a\mathbf{D} - c'\mathbf{A}]b_h = \phi_h \tag{2.34}$$

Where a and c' are defined as follows:

$$a = \frac{4h_h^2}{1 - 4h_h^2} \tag{2.35}$$

$$c' = \frac{2h_h}{(1 - 4h_h^2)} \tag{2.36}$$

ϕ_h and b_h denote prior and final beliefs, respectively. $\mathbf{A}$ denotes the $n \times n$ adjacency matrix of an underlying graph of n nodes, $\mathbf{I}$ denotes the $n \times n$ identity matrix, and $\mathbf{D}$ is a $n \times n$ diagonal matrix of degrees where $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ and $\mathbf{D}_{ij} = 0$ for $i \neq j$. h_h denotes the homophily factor between nodes (i.e. their "coupling strength" or association). More specifically, higher homophily means that close nodes tend to have more similar labels. The coefficient values are set as above for convergence reasons; we refer the interested reader to [89] for further discussion.

Part II

Content-based Techniques for Misinformation Detection

3

Content-based Detection of Misinformation Using Tensor Embedding

Up until now, different strategies have been implemented to extract insightful information from content of news articles. For instance, there are many works on extracting linguistic based information like lexical features that include character and word level information and syntactic features that leverage sentence level information. There are some other approaches that exploit other content based information rather than linguistic based features e.g. number of nouns, proportion of positive/negative words article length etc. The main drawback of lexical based techniques such as bag of words is that there is no consideration about the relationship between the words within the text and the latent patterns they may form when they co-occur.

Moreover, the majority of existing work for misinformation detection leverage supervised classifiers. For instance, Rubin et al. [131] use a linguistic features and a SVM-based classifier. In a similar way, Horne et al. [70] apply a SVM classifier on stylistic and psychological features. The main issue with this category of misinformation detection is that they mostly require a considerable amount of labeled data known as ground truth for training. However, in real world scenarios, these labels are extremely limited and sometimes unreliable. Although there are a couple of fact checking websites e.g., PolitiFact, FactCheck, Snopes and so on and so forth, they all require human expert and that is why the task of fact checking is often a costly and time consuming process.

In contrast to the aforementioned works, in this chapter, we propose a novel tensor-based approach which not only considers relationships between words within article text but also requires a limited amount of labeled data and human supervision.

More precisely, in this chapter we propose a novel strategy mixing tensor-based modeling of article content and semi-supervised learning on article embeddings for detecting misinformation. Our method requires very few labels to achieve state-of-the-art results. We propose and experiment with three different article content modeling variations which target article body text or title, and enable meaningful representations of word co-occurrences which are discriminative in the downstream news categorization task. We evaluate our proposed models on real world data and we show that our approach achieves 71% accuracy on a large dataset using only 2% of the labels. Additionally, our approach is able to classify articles into different fake news categories (clickbait, bias, rumor, hate, and junk science) by only using the titles of the articles, with roughly 70% accuracy and 30% of the labeled data.

Our main contributions are:

- We propose three different tensor-based embeddings to model content-based information of news articles which decomposition of these tensor-based models produce concise representations of spatial context and provide us with insightful patterns for classification of news articles.
- Introducing a tensor-based modeling approach which is not only applicable on body text but also capable of modeling news articles just using article title.
- We leverage a propagation based approach for semi-supervised classification of news articles which enables us to classify news articles when there is scarcity of labels.
- We create a large dataset of misinformation and real news articles out of publicly shared tweets on Twitter.
- We evaluate our method on real datasets. Experiments on two previously used datasets demonstrate that our method outperforms prior works since it requires a fewer number of known labels and achieves comparable performance.

3.1 Problem definition

In this work, we follow the definition used in [138] and consider articles that are *"intentionally and verifiably false,"* as fake news or misinformation. Based on this definition, we leverage content of news articles to discriminate fake news from real content. Henceforth, we refer to the body text or title of the articles as "content".

Suppose $N = \{n_1, n_2, n_3, ..., n_M\}$ is a collection of M news articles and $\mathbf{l}$ is a vector that comprises labels of articles in N. We define the following problems:

Problem 1: Given N and $\mathbf{l}$ with entries labeled as real, fake or unknown,

Problem 2: Given $\mathcal{N}$ and $\mathbf{l}$ with entries labeled as real, bias, clickbait, conspiracy, fake, hate, junck science, satire, and unreliable or unknown, and the majority of entries are unknown, **predict** the labels of the unknown articles.

Due to the scarcity of known labels Both problems are addressed by semi-supervised techniques.

3.2 Proposed method

In this section, we introduce a content-based method for semi-supervised classification of news articles. This method consists of three consecutive steps: step 1 refers to the modeling of articles' content as a tensor, and decomposition of resulted model into factor matrices. In step 2, we leverage the factor matrix corresponding to article mode to create a k-NN graph in order to represent the proximity of articles. In step 3, we use Fast belief propagation technique (FaBP) to propagate

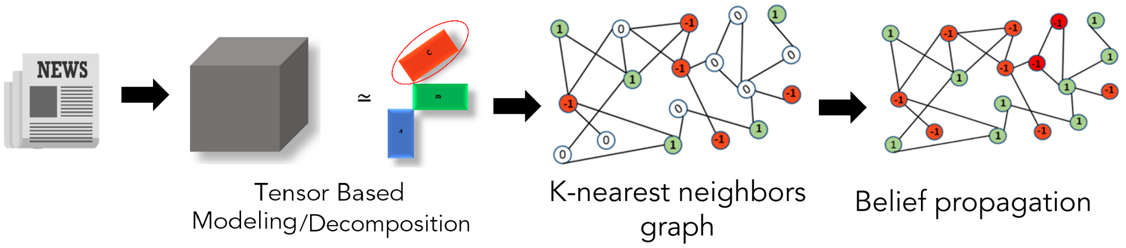


Figure 3.1: Our proposed method discerns real from misinformative news articles via leveraging tensor representation and semi-supervised learning in graphs.

very few known labels throughout the graph to predict the unknown ones. Fig. 3.1 illustrates our proposed method. In what follows, we discuss each step in detail.

Step 1: Tensor decomposition.

The very first step of our proposed method is to model news articles. We propose a novel tensor-based approach to model articles' content. We define 3 different models as follows:

Model 1: Term-Term-Article (TTA)

We define each news article in N as a matrix representing the co-occurrence of the words within a sliding window of size w. We create a tensor embedding by stacking co-occurrence matrices of all news article in N as proposed in [71].

In other words, in this three-mode tensor $\mathcal{X} \in \mathbb{R}^{\mathbf{I \times I \times M}}$ (Term, Term, Article) each news article is a co-occurrence matrix in which entry (t_1, t_2) of the matrix is a binary value representing the co-occurrence of terms t_1 and t_2 (for binary-based model) or the number of times this pair of terms co-occur across the text (for frequency-based model) within a window of size w (usually 5-10)

[1].

Model 2: Term-Term-Term-Article (3TA)

In previous model, we created a tensor embedding as a representative of all couples of co-occurred words within news articles. In this model, we aim to design a tensor-based embedding which demonstrates meaningful co-occurrence of larger set of words within an article, i.e., instead of pairwise co-occurrence we are interested in capturing all triple-way co-occurred words Fig. 3.2. Therefore, the resulted tensor embedding is going to be a 4-mode tensor $\mathcal{X} \in \mathbb{R}^{\mathbf{I \times I \times I \times M}}$ (Term, Term, Term, Article) created by stacking 3-mode tensors each of which representing triple-way co-occurrence within each news article. In other words, instead of a co-occurrence mat ices, we have co-occurrence tensors for each article.

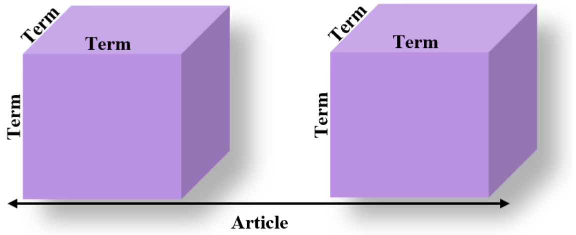


Figure 3.2: Modeling content based information of articles using a 4 mode Tensor (3TA).

Model 3: Term-Term-Article on news Title (`TTA` out of titles)

In some cases, the title of a news article is as informative as the body of article. For example, there is a category of news articles known as clickbait in which the headline of the article is written in such a way that persuade the reader to click on the link. The authors of this category of try to use

[1] We experimented with small values of this interval and results were qualitatively similar.

some persuasive words to tempt readers to follow their articles [28, 121].

The study of news article titles could be very interesting in situations where the webpage does not exist anymore but the tweet/post or a shared link still comprises the title. Moreover, leveraging only the title is very useful for early detection of misinformation because we would be able to predict trustworthiness of the content before browsing malicious webpages. Having this in mind, the main goal here is to investigate how a model created out of the words (terms) of in the title can capture nuance differences between categories of fake news. To this end, we create a `TTA` Tensor but this time we only use title words (terms).

Step 2: k-NN graph of news articles.

Decomposition of the tensor-based models we introduced in Step 1 results in three or four factor matrices (3 for `TTA` and 4 for 3TA) which are embeded representations of each mode of the tensor (Term, Term or News articles) and comprise corresponding latent patterns. Using the factor matrix corresponding to article mode (factor matrix **C** for `TTA` model and **D** for 3TA model), we can create a graphical representation of news articles where each row (article) of the factor matrix **C** or **D** corresponds to a node in k-NN graph G. In other words, factor matrix **C** or **D** which is a representation of the news articles in the latent topic space can be used to construct a k-NN graph which later on will be leveraged to find similar articles. We consider each row in **C** or $\mathbf{D} \in \mathbb{R}^{M \times R}$ as a point in R-dimensional space where R is the rank of decomposition. Then, we compute ℓ_2 distance among nodes (news) to find the k-closest points for each data point in **C** or **D**. In practice, the number of news articles is extremely large, so, in order to find the k-nearest-neighbors of each article efficiently, we propose to use a well-known kd-tree based optimizations as explained in [122].

Each node in G represents a news article and an edge illustrates the similarity of corresponding nodes in the article embedding space. In this step, we only leverage the distance as a means to measure similarity between news articles, without much concern for the actual order of proximity. Thus, we enforce symmetry in the neighborhood relations, that is, if n_1 is a k-nearest-neighbor of news n_2, the opposite should also hold. The result is an undirected, symmetric graph where each node is connected to at least k nodes. The graph can be compactly represented as an $M \times M$ adjacency matrix.

Step 3: Belief Propagation.

Using the graphical representation of the news articles, and considering that for a small set of those news articles we have ground truth labels, our problem becomes an instance of semi-supervised learning over graphs. We use a belief propagation algorithm which assumes homophily i.e., news articles that are connected in the k-NN graph are likely to be of the same type due to the construction method of the tensor embeddings; moreover, [71] demonstrates that such embeddings produces fairly homogeneous article groups. we use the fast and linearized FaBP variant proposed in background section. The algorithm is demonstrated to be insensitive to the magnitude of the known labels.

3.3 Implementation

In this section, we describe the implementation details of our proposed method and datasets we experiment on.

We implement our proposed method in MATLAB using the Tensor Toolbox [9] and for

FaBP we leveraged the implementation exists in [89]. We run each experiment 100 times and report the average and standard deviation of the results. The reminder of this section describes the datasets we experimented on.

Table 3.1: Dataset specifics.

Datasets	**# fake news**	**# real news**	**# total**
Dataset1 (Political)	75	75	150
Dataset2 (Bulgarian)	69	68	137
Our dataset	31,739	31,739	63,478

3.3.1 Dataset description

For evaluation of our proposed method, we use two public datasets each of which consist of hundreds of articles. We also create a new dataset that comprises more than 63k articles, as shown in Table 3.1.

Public datasets

The first public dataset i.e., Dataset1 provided by [70] consists of 150 political news articles and is balanced to have 75 articles of each class. Dataset2, the second public dataset, provided by [62], includes 68 real and 69 fake news articles.

Our dataset

For our dataset, we implemented a crawler in Python to crawl Twitter to collect news article URLs mentioned in some tweets during a 3-month period from June 2017 to August 2017. Our crawler extracts news content using the web API boilerpipe[2] and the Python library Newspaper3k [3]. For

[2] http://boilerpipe-web.appspot.com/

[3] http://newspaper.readthedocs.io/en/latest/

some few cases where these tools were not able to extract news content, we used Diffbot [4] which is another API to extract article text from web pages.

All real news articles were taken from Alexa [5] from 367 different domains, and fake news articles belong to 367 other domains identified by B.S. Detector (a crowd source tool box in form of browser extension to annotate fake news sites) [22]. Table 3.2 demonstrates different categories specified by B.S. Detector. For this work, we consider news from all of these categories as misinformative. With this in mind, our new dataset comprises 31,739 fake news and 409,076 real news articles. We randomly down-sampled the real class to create a balanced dataset. The distribution of different fake categories has been illustrated in Fig. 3.3.

It is worth mentioning that we remove stopwords and punctuations from both body and the title of news articles and for all three datasets we pre-processed the data using tokenization and stemming.

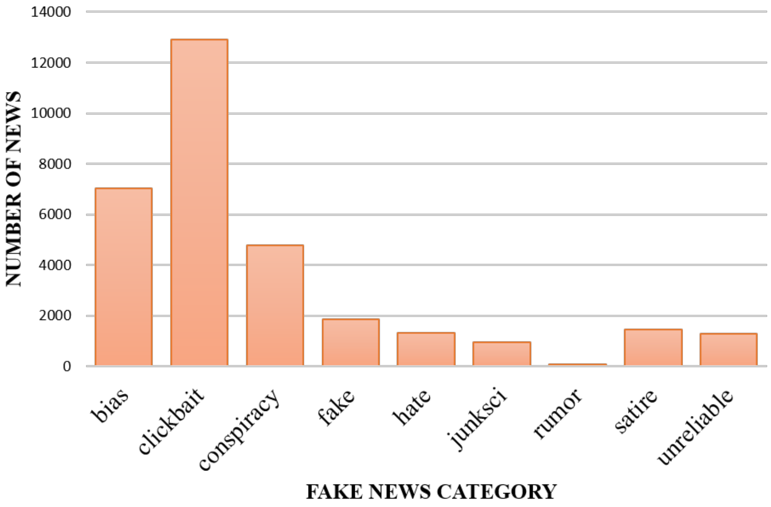


Figure 3.3: Distribution of misinformation per domain category in our collected dataset.

Table 3.2: Domain categories collected from BSDetector [22], as indicated in our dataset. The category descriptions are taken from [22].

Category	**Description**
Bias	"Sources that traffic in political propaganda and gross distortions of fact."
Clickbait	"Sources that are aimed at generating online advertising revenue and rely on sensationalist headlines or eye-catching pictures."
Conspiracy	"Sources that are well-known promoters of kooky conspiracy theories."
Fake	Sources that fabricate stories out of whole cloth with the intent of pranking the public.
Hate	"Sources that actively promote racism, misogyny, homophobia, and other forms of discrimination."
Junk Science	"Sources that promote pseudoscience, metaphysics, naturalistic fallacies, and other scientifically dubious claims."
Rumor	"Sources that traffic in rumors, innuendo, and unverified claims."
Satire	"Sources that provide humorous commentary on current events in the form of fake news."

3.4 Evaluation

In this section, first, we discuss experimental results of our basic tensor-based model i.e., model 1 (`TTA`) against state-of-the-art baselines and then we compare our model 2 (3TA) against model 1 (`TTA`) to show which one performs better in terms of classification accuracy. Finally, we state the experimental results of model 3 (`TTA` out of titles) to see how successful the `TTA` model is when the only available information about the articles is the title.

[4] https://www.diffbot.com/dev/docs/article/

[5] https://www.alexa.com/

3.4.1 Experimental results

Evaluation of basic tensor based model (`TTA`) against baselines

In this section, we discuss the experimental evaluation of model 1 or `TTA` on body text of the articles. For this section, we use cross-validation where we evaluate different settings with respect to R i.e., decomposition rank and k i.e., the number of nearest neighbors which controls the density of the k-NN graph G).

Practically, decomposition rank is often set to be low for time and space reasons [134], so, we grid searched for values of R in range 1 to 20 and values of k in range 1 to 100. In fact, by increasing k we trade off the greater bias for less variance.

As a result of this experiment, parameters R and k both are set to be 10. As illustrated in Fig. 3.4, performance for values of k and R greater than 10, are very close. Also, using a small k value (for example, 1 or 2), leads to a poor accuracy because building a k-NN graph with small k results in a highly sparse graph which means limited propagation capacity for FaBP step.

As mentioned in Section 8.2, we consider two different tensor embeddings: frequency-based and binary-based. Fig. 3.5 shows the performance of our proposed method using these two tensors. As shown, the binary-based tensor performs better than frequency-based tensor in classification task. Thus, we used binary-based representations in all of our experiments. Later on, in Section 3.5, we will discuss why binary-based tensor achieves better classification performance.

Evaluation of our method with different percentages p of known labels (5% to 30%) is reported in Table 3.3.

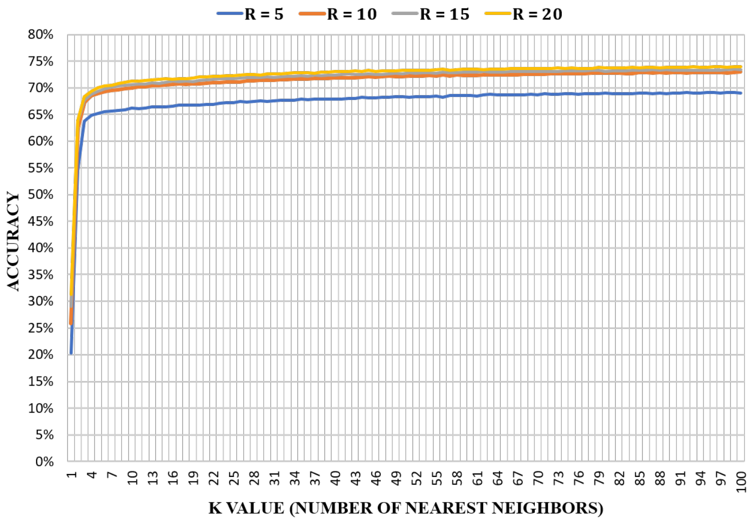


Figure 3.4: Performance using different parameter settings for decomposition rank (R) and number of nearest neighbors (k).

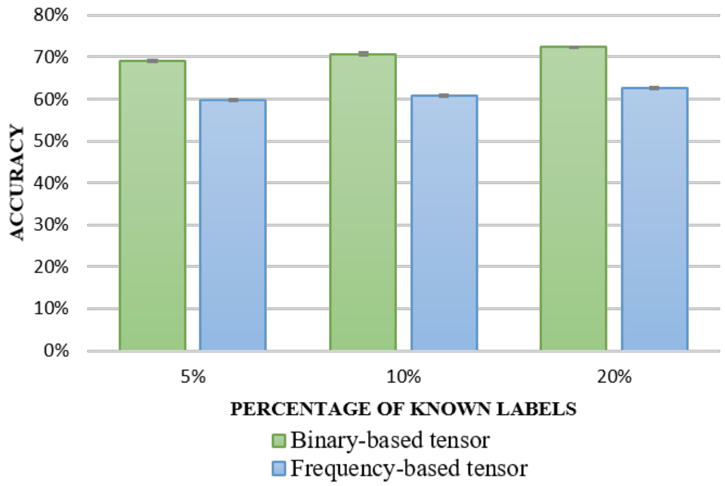


Figure 3.5: Detection performance using different tensor representations.

As demonstrated, using only 10% of labeled articles we achieve an accuracy of 70.76%. To compare the robustness of our tensor based embedding against widely used term frequency inverse-document-frequency (`tf-idf`) representation we constructed a k-NN graph built from the (`tf-idf`) representation of the articles as well. Fig. 3.6 illustrates that tensor embeddings con-

Table 3.3: Performance of the proposed method using our dataset with different percentages of labeled news.

%Labels	Accuracy	Precision	Recall	F1
5%	69.12 ± 0.0026	69.09 ± 0.0043	69.24 ± 0.0090	69.16 ± 0.0036
10%	70.76 ± 0.0027	70.59 ± 0.0029	71.13 ± 0.0101	70.85 ± 0.0043
20%	72.39 ± 0.0013	71.95 ± 0.0017	73.32 ± 0.0043	72.63 ± 0.0017
30%	73.44 ± 0.0008	73.13 ± 0.0028	74.14 ± 0.0034	73.63 ± 0.0007

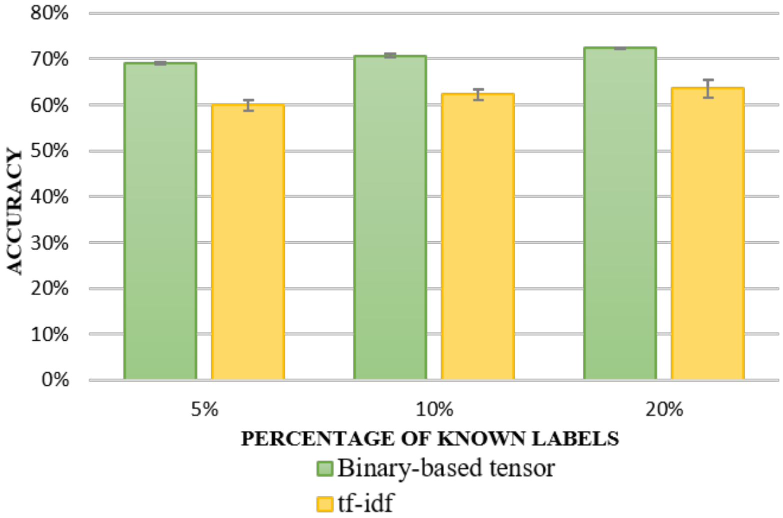


Figure 3.6: Comparing accuracy of tensor-based embedding approach against `tf-idf` approach for modeling of news content.

sistently results in better accuracy than (`tf-idf`) baseline for different known label percentages. This results empirically justify that binary-based tensor representations can capture spatial/contextual nuances of news articles better than widely used bag of words method for modeling the content of news articles.

We also experimented on an extremely sparse known labels setting i.e., known labels <5% for different values of nearest neighbors. Based on what is shown in Fig. 3.7, our proposed method achieves an accuracy of 70.92% using just 2% of known labels when the number of nearest neighbors is 200. Indeed, the performance of our approach decreases fairly with even smaller proportions of known labels.

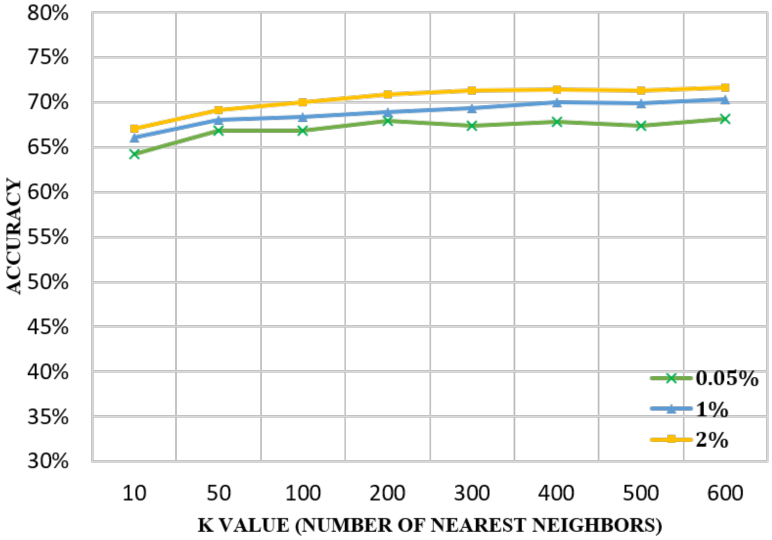


Figure 3.7: Performance of proposed approach using extremely sparse (<5%) labeled set of articles and varying number of nearest neighbors.

We also applied our proposed method on Dataset1 and Dataset2 and compared the accuracy against the accuracy of the following approaches:

- SVM on content-based features as proposed in [70]. We use suggested features extracted from news content and apply an SVM classifier on different percentages of training data.
- Logistic regression on content-based features proposed by [62]. We used publicly available implementation of this work to extract linguistic (n-gram) features of the articles content.

.

Fig. 3.8. demonstrates experimental results for different approaches on Dataset1. As illustrated, our approach achieves better accuracy even with fewer labels. For instance, using only 30% of news labels we achieved 75.43% accuracy, whereas SVM(30%/70% train/test), SVM(5-

fold cross-validation), and logistic regression (30%/70% train/test) attain 67.43%, 71% and 50.09% accuracy, respectively. The accuracy achieved by SVM (5-fold cross-validation) was reported by Horne et al. in [70].

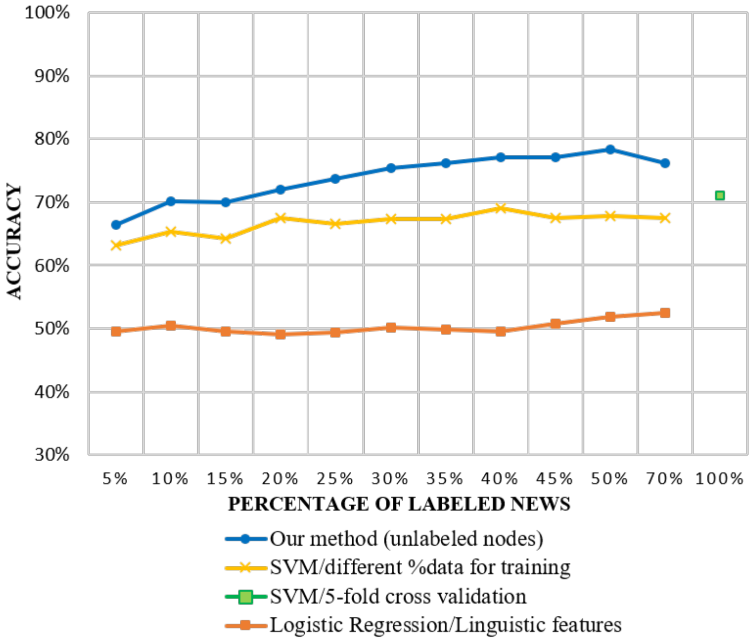


Figure 3.8: Performance using Dataset1 provided by Horne et al. [70]

Moreover, we applied logistic regression and SVM, using 10%/90% train/test split on Dataset2. The accuracy of these approaches is 59.84% and 64.79%, respectively whereas, the accuracy of our proposed method is 67.38%.

One justification for reported improvements in terms of classification accuracy is that the co-occurrence strategy of tensor based modeling better captures the nuanced patterns within news content than widely used bag of words and tf-idf approaches.

Furthermore, another justification is that we leverage the k-NN graph in addition to belief propagation approach which allows us to exploit similarity between even unlabeled news articles and make our proposed approach stronger than supervised classification techniques when we experiment on extremely sparse known label regimes.

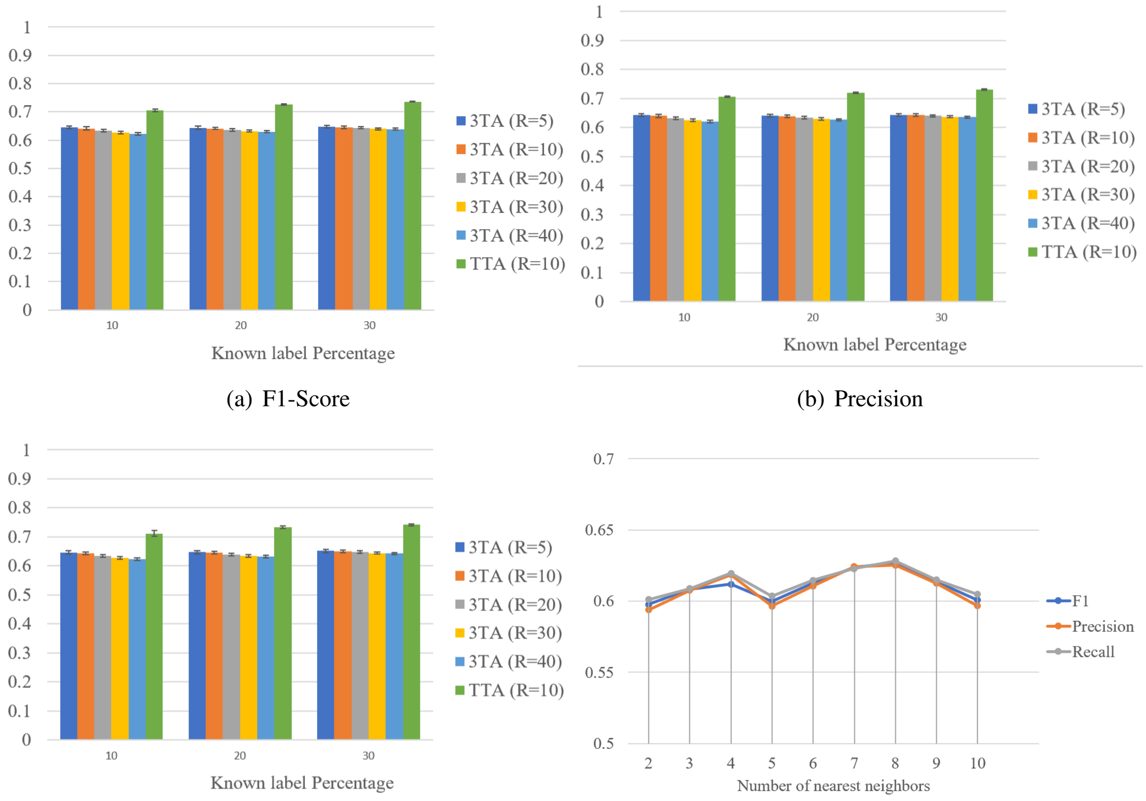
Evaluation and comparison of model 1 (TTA) and model 2 (3TA)

For the second experiment, we investigate the effect of increasing number of co-occurred terms, which corresponds to a higher mode tensor model, to evaluate whether or not considering a larger subset of co-occurred words creates a more robust model. Fig. 3.9 demonstrates the classification performance of model 2 (3TA) in comparison to model 1 (TTA) for different ranks of decomposition in terms of F1 score, precision, recall and accuracy.

As shown in Fig.3.9, increasing the number of co-occurred terms results in a considerable decline in the classification performance. One possible justification is that, by increasing the number of entries in co-occurrence tuple, we create a model that is more representative of an individual news article than a class. In other words, this model is essentially over-fitting to specific articles. In fact, it is harder to find the nodes (articles) that share the same patterns of triple-way co-occurrence than a co-occurring pair. Thus, our embedding of choice is model 1 (TTA).

Evaluation of model 3 (TTA out of titles)

In this section, we discuss the experimental results of model 3 or creating a TTA model out of articles' title. As we discussed in evaluation of model 1, the binary-based tensor results in a higher accuracy. With that in mind, and due to the fact that the number of words within titles are quite less than body, we choose binary-based tensor over frequency-based model for this experiment as well. We create 9 balanced binary-based tensors separately, each of which consists of 50% of under study



(a) F1-Score

(b) Precision

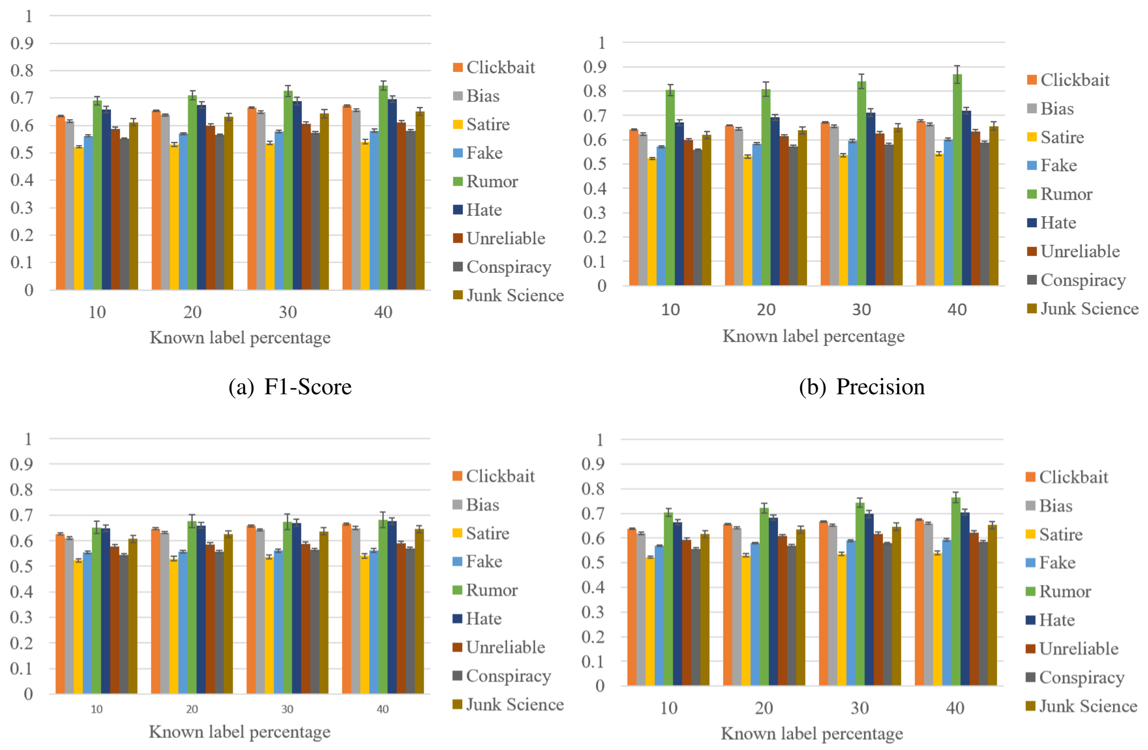
(c) Recall

(d) Effect of using different values of K

Figure 3.9: F1 score, precision and recall for applying our proposed method on TTA and 3TA models and the effect of using different K on these metrics. As illustrated, TTA with rank 10 outperforms 3TA with four different ranks of decomposition. Moreover, increasing the number of neighbors causes a significant decline in classification metrics.

category and 50% of real articles. The F1 score, precision, recall and accuracy are shown in Fig. 3.10.

As demonstrated in Fig. 3.10, when we only use the titles, the classification performance differs from category to category. We observe that the title of articles belong to news articles from clickbait, bias, rumor, hate and junk science categories are more informative and possibly convey more information about the content of the articles. Since we are more successful in classification task for these categories, we may conclude that there are more meaningful co-occurrence for the



(a) F1-Score

(b) Precision

(c) Recall

(d) Accuracy

Figure 3.10: F1 score, precision, recall and accuracy of applying our proposed method on a `TTA` tensor constructed out of title for different categories of fake articles. As shown, the results differ from category to category and best results belong to clickbait, bias, rumor, hate and junk science categories.

terms in the title of these categories than the rest of the classes which means the title of these classes is not as informative as the body.

3.5 Sensitivity Analysis

A question that may come to mind regarding the effectiveness of our proposed method is that how the length and categorical distribution of the articles impact the performance of our proposed method when we use frequency-based and binary-based tensor embeddings. To answer this question, we create sub-sampled datasets from our dataset which meet the following conditions:

- News articles have similar content length and are selected across news categories
- News articles vary in length and belong to the same news category
- News articles have similar content length and belong to the same news category

Table 3.4: Dataset statistics per fake news category

Dataset	**Article Length**		
	Minimum	**Mean**	**Maximum**
Bias	18	363	5,903
Clickbait	18	355	10,955
Conspiracy	19	422	10,716
Fake	20	378	8,803
Hate	20	315	5,390
Junk Science	22	364	5,390
Satire	18	307	8,913
Unreliable	20	360	5,268

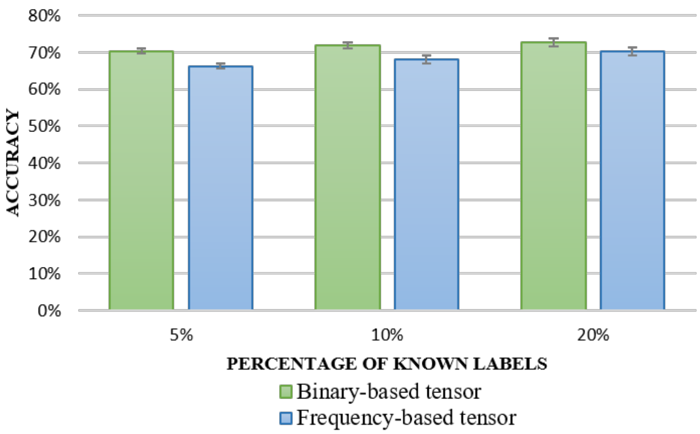
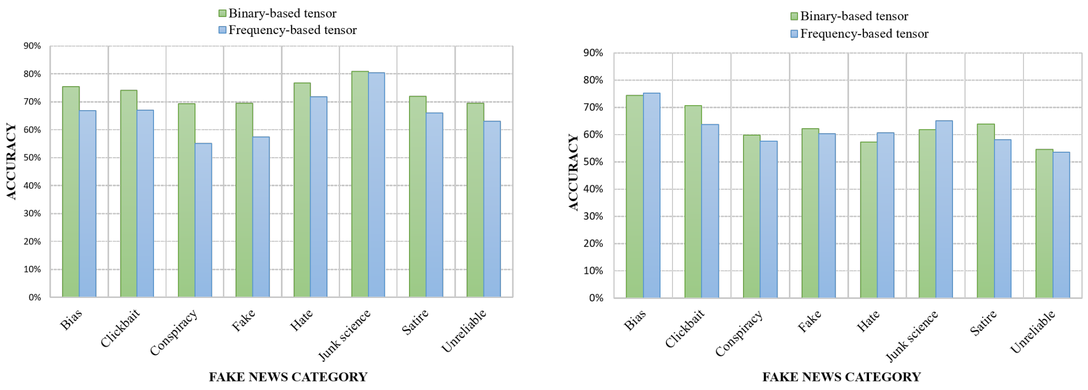


Figure 3.11: Comparing performance of binary vs. frequency-based tensor embeddings on news articles of all types with similar content length.

First, we evaluated our proposed method on a dataset of news articles with similar length



(a) Varying in length

(b) With similar length

Figure 3.12: Performance of binary vs. frequency-based tensor embeddings on category-partitioned news articles

where the fake articles are selected across news categories. (Table 3.4 illustrates summary statistics for article length across each fake news category).

Fig. 3.11 demonstrates the accuracy achieved by our method using both binary-based and frequency-based tensor embeddings. We observe that the performance of our method is not sensitive to news category especially when length is standardized.

In addition, we evaluated our method using 8 sample datasets, one for each misinforming news category: bias, clickbait, conspiracy, fake, hate, junk science, satire, and unreliable. Each dataset was balanced, containing the same number of fake and real articles. Note that in these sample datasets, news article length varies (see Table 3.4).

Fig. 3.12, shows the accuracy achieved by our proposed method on each category using both binary-based and frequency-based embeddings. These results show that the binary-based representation noticeably outperforms the frequency-based for all categories of fake news.Then, we perform another experiment where we only select news articles that have pretty much the same length per category. We observe that the news articles length greatly affects the performance of tensor embedding. More specifically, we conclude that binary-based tensors indicating boolean co-occurrences between words better captures spatial/contextual nuances of news articles that vary in length. On the same note, detection performance is relatively comparable for embedding types when considering articles of a fixed or almost-fixed length.

3.6 Related Work

3.6.1 Supervised content-based models

The majority of existing work for misinformation detection applies supervised learning models on extracted features from news content. For instance, in [62], Hartl et.al. extract linguistic (*n*-gram), credibility (punctuation, pronoun use, capitalization) and semantic features from the news content and then applied a logistic regression classifier to detect misinformation. In another work Horne et.al. apply a SVM classifier on stylistic, complexity and psychological features extracted from content of the articles and classified them into real, fake and satirical news [70]. In [123], Qazvinian et.al. leverage a naive-Bayes classifier on content, network and microblog-specific features for detecting rumors. For rumors detection task there is another work in which a Dynamic Series-Time Structure (DSTS) model is proposed to capture the social context of an event from content, user and propagation-based features [105]. Methods proposed in [104] and [132] model temporal structure using a recurrent neural network (RNN) to represent text and user characteristics. The majority of aforementioned works are based on complicated models and require human experts for feature based modeling and considerable volume of data for training and testing steps in a supervised manner, whereas we proposed a semi-supervised approach in which we construct a tensor based model out of news content which not only is simple and fast but also outperforms some state-of-the-art supervised approaches by using very few amount of labeled data.

3.6.2 Propagation models

There are previous works that leverage propagation-based models for evaluating news articles credibility. For instance, the authors in [58] leverage a PageRank-like credibility propagation method on multi-typed network of events, tweets and users. For news verification task, the authors in [75] propose a credibility network based on positive and negative view points about news articles. In [73], a hierarchical propagation model on a three-layer credibility network is proposed which comprises event, sub-event and message layers. All of aforementioned works require some kind of initial credibility values which obtained from the output of a supervised classifier. As mentioned before, in contrast to these works, we leverage a semi-supervised propagation based approach that requires very few labels and outperforms previous work while using less than 5% of labeled data.

3.7 Conclusions

In this chapter, we propose a tensor-based semi-supervised approach for distinguishing misinformation. We propose three different tensor-based models which decomposing them provide us with patterns that indicate different categories of news articles. We leverage a k-nearest neighbor graph to represent the proximity of news articles using the latent patterns extracted from decomposition of tensor models and apply belief propagation algorithm for propagating very few available labels, throughout the k-nearest neighbor graph. We evaluate our proposed method on two public datasets and a dataset we created from over 63K articles. Experimental results on these real-world datasets illustrates that our approach outperform many state of the art content based methods in terms of accuracy even when we use very few known labels. More specifically, our method achieves accuracy of 75% on first public dataset and accuracy of 67% on second public dataset using 30% and 10% of the labels respectively. Moreover, the classification accuracy of our proposed method on our dataset is 71% using only 2% of labels. Meanwhile, our tensor-based model created out of titles is able to classify articles into different categories of fake news specially clickbait, bias, rumor, hate and junk science categories with accuracy of roughly 70% using just 30% of the labeled data.

4

`Vec2Node` A Tensor-based Augmentation Technique for Few Shot Learning

Recent advances in state-of-the-art machine learning models like deep neural networks heavily rely on large amounts of labeled training data which is difficult to obtain for many applications. To

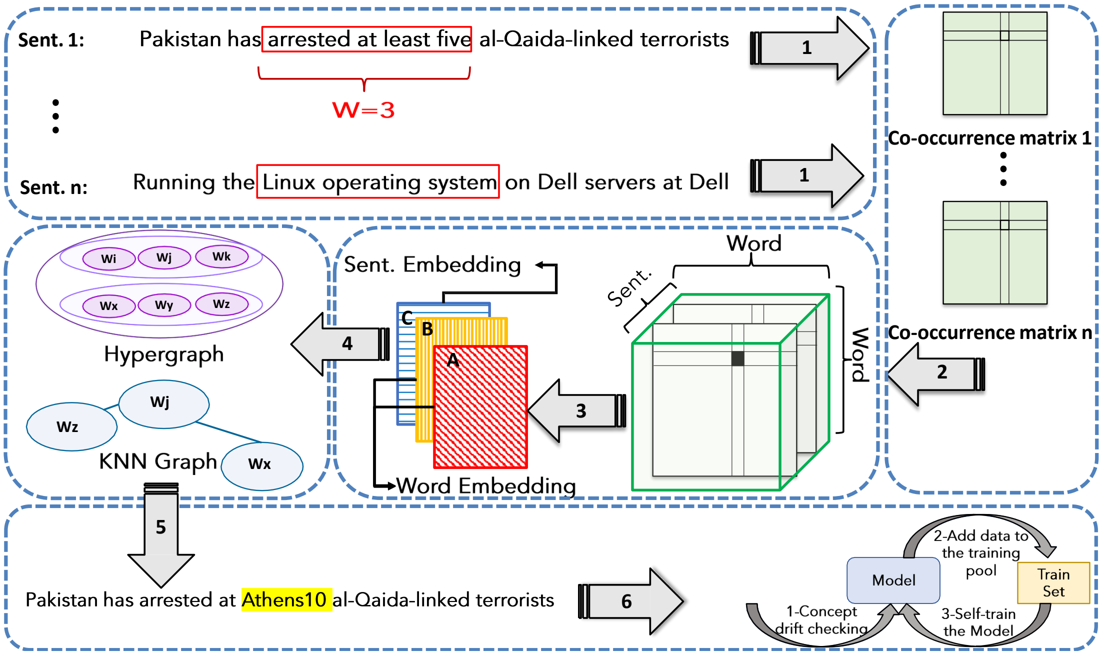


Figure 4.1: Overview of the proposed approach.

address label scarcity, recent work has focused on data augmentation techniques to create synthetic training data. In this work, we propose a novel approach of data augmentation leveraging tensor decomposition to generate synthetic samples by exploiting local and global information in text and reducing concept drift. We develop Vec2Node that leverages self-training from in-domain unlabeled data augmented with tensorized word embeddings that significantly improves over state-of-the-art models, particularly in low-resource settings. For instance, with only 1% of labeled training data, Vec2Node obtains a 21.5% improvement over the base model with augmentation. Furthermore, Vec2Node generates interpretable explanations for the augmented data leveraging tensor embeddings.

4.1 Introduction

In recent years, neural network models have obtained state-of-the-art performance in several language understanding tasks employing non-contextualized fastText [18] as well as contextualized BERT [36] word embeddings. Even though these models have been greatly successful, they rely on large amounts of labeled training data for their state-of-the-art performance. However, labeled data is not only difficult to obtain for many applications, especially for tasks dealing with sensitive information, but also requires time consuming and costly human annotation efforts. To mitigate label scarcity, recent techniques such as self-training [39, 65] and few shot learning [178, 185] methods have been developed to learn from large amounts of in-domain unlabeled or augmented data. The core idea of self-training is to augment the original labeled dataset with pseudo-labeled data [65] in an iterative teacher-student learning paradigm. Traditional self-training techniques are subject to gradual concept drift and error propagation [178, 192]. In general, data augmentation techniques aim to generate synthetic data with similar characteristics as the original ones. While data augmentation has been widely used for image classification tasks [116] leveraging techniques like image perturbation (e.g., cropping, flipping) and adding stochastic noise, there has been limited exploration of such techniques for text classification tasks. Recent work on data augmentation for text classification tasks like [185] rely on auxiliary resources like an externally trained Neural Machine Translation (NMT) system to generate back-translations[1] for consistency learning.

In contrast to the above works, we solely rely on the available in-domain unlabeled data for augmentation without relying on external resources like an NMT system. To this end, we develop Vec2Node that employs tensor embeddings to consider both the global context and local

[1] The process of translating the text to another language and translating it back to the original language.

word-level information. In order to do so, we leverage the association of words and their tensor embeddings with a graph-based representation to capture local and global interactions. Additionally, we learn this augmentation and the underlying classification task jointly to bridge the gap between self-training and augmentation techniques that are learned in separate stages in prior works.

Our contributions can be summarized as follows:

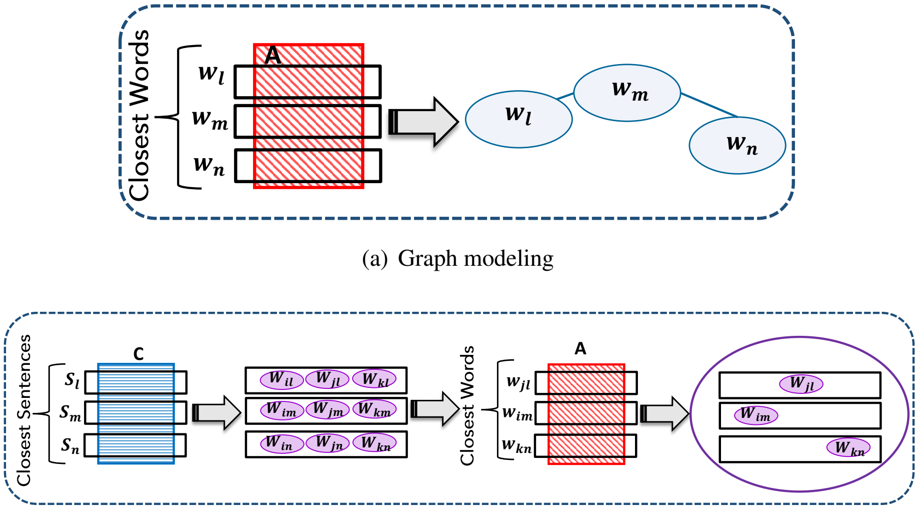
- A novel tensor embedding based data augmentation technique for text classification with few labels.
- A dynamic augmentation technique for detecting concept drift learned jointly with the downstream task in a self-training framework.
- Extensive evaluation on benchmark text classification datasets demonstrate the effectiveness of our approach, particular in low-resource settings with limited training labels along with interpretable explanations.

4.2 `Vec2Node` framework

4.2.1 Problem formulation

Given a corpus D of some labeled data, we want to **generate** augmented D' that improves the performance of a classification model M on the downstream task i.e. $f(M(D)) > f(M(D + D'))$, where f is an evaluation measure (e.g., accuracy of the classifier).

To address the problem above, we propose a novel tensor-based approach for generating synthetic texts from the corpus D. The details of the proposed approach, henceforth referred to as Vec2Node are described in the following section.



(a) Graph modeling

(b) Hypergraph modeling

Figure 4.2: Graph and hypergraph modeling for finding similar words i.e., candidates for substitution.

4.2.2 Data augmentation

Vec2Node leverages tensor decomposition to find word and text embeddings. These are further used for graph-based representations of the words to find similar ones for word replacement and generation of synthetic samples while minimizing the concept drift. Vec2Node consists of the following steps:

Tensor-based corpus representation

Textual content of documents can be represented by a co-occurrence tensor [52, 133] which embeds the patterns shared between different topics or classes. These patterns are formed by words that are more likely to co-occur in documents of the same class. We leverage similar principles to capture existing similarities within a given text. To this end, given a set of samples, we first slide a window of size w across the text of each sample and capture the co-occurring words to represent them in a co-occurrence matrix. Thereafter, we stack the co-occurrence matrices of all samples to form a 3-mode tensor of dimension $T \times T \times S$ where T is the number of terms or words in the entire corpus and S is the number of samples. This process is demonstrated in Fig.. 8.1. The rationale behind this approach is to capture the context (words) for a given target word. In the experimental section, we demonstrate how this approach captures contextually related words.

Decomposing tensors into word and text embeddings

The objective of this step is to embed the words and the texts of the corpus into rank-R representations which are later used for calculating word similarities. As explained in Section **??**, we can use CP/PARAFAC to decompose our 3-mode tensor as:

$$\boldsymbol{X} \simeq \Sigma_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{4.1}$$

Where $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \dots \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \dots \mathbf{b}_R]$, and $\mathbf{C} = [\mathbf{c}_1 \ \mathbf{c}_2 \dots \mathbf{c}_R]$ are factor matrices or embeddings that represent word, word and text respectively. The word co-occurrence **A** and **B** are symmetric. Thus, they capture the same information.

Leveraging tensor embeddings for KNN and hypergraph representation

In this step, we use the word and text embeddings provided by the tensor decomposition to model a given corpus using KNN tensor or hypergraph representation. We exploit these representations for estimating similar words and texts as the best candidates for replacement in a given context to generate new synthetic samples. We suggest two different graph based representation for the corpus as follows:

K-Nearest Neighbors Graph (KNN) modeling. Consider the factor matrix **A** (or **B**, as mentioned, they are symmetric and capture the same information) of dimension $N \times R$ where each row is tensor word embedding in R-dimensional space $\mathcal{R}^R$. We represent the ith row of this matrix corresponding to word i as a node in R dimensional space. This allows us to calculate the Euclidean distance between the nodes and represent the similarity between the nodes or words with a weighted undirected edge. In other words, the Euclidean distance between row i and row j measures the similarity of these two vectors in the R-dimensional space.

Hypergraph modeling. [147] propose a hypergraph modeling of the documents where hyperedges are defined by consecutive sentences and words within them, that are captured by sliding a window across the text. In contrast to this work where the similarity is considered by spatial closeness, we first leverage the factor matrix **C** corresponding to text embedding to find K closest samples and then we use factor matrix **A** to find $K^{'}$ closest words within these K samples. The details of this process are shown in Fig. 4.2. It is worth mentioning that *our proposed model uses KNN tensor graph* for modeling word similarities. However, for comparison purposes we also implement Vec2Node framework with hypergraph modeling.

4.2.3 Learning with data augmentation and limited labels

Contextualized word replacement

Modeling the corpus using graph or hypergraph representations allows us to find similar words by sorting the edge weights or the Euclidean distances between the nodes, and picking the ones with the smallest distance (i.e., closest words) as the best candidates for replacement and creation of synthetic samples. This process is fully unsupervised given that the tensor decomposition method does not require any labels. Also, it considers local and global contextual information given the graph and tensorial representation of words and texts.

Self-training with consistency learning

In order to eliminate noisy samples, we check for *concept drift* between the original samples and the synthetic ones leveraging consistency learning in a self-training framework.

Given a few labeled samples $\{x_l, y_l\} \in D_l$ for the downstream task, we first fine-tune a base model (e.g., FastText or BERT) with parameters θ.

Consider x_u to be the target augmented pair for a source instance x_l generated using the augmentation technique described before. We can use the current parameters θ of the model to predict the pseudo-label for the target x_u as:

$$y_u = argmax_y \; p(y|x_u;\theta) \tag{4.2}$$

Since the objective of data augmentation is to generate semantically similar instances for the model, we expect the output labels for the source-target augmented pair $\{x_l, x_u\}$ also to be similar; otherwise, we designate this as a concept drift and discard augmented pairs where $y_l \neq y_u$.

We add the remaining target pseudo-labeled data with consistent model predictions with the source data as our augmented training set $\{x_u, y_u\} \in D_u$ and re-train the base model to update θ. The above steps are repeated with iterative training of the base model with pseudo-labeled augmented data until convergence. The optimization objective for the above self-training process can be formulated as:

$$min_\theta\ \mathbb{E}_{x_l,y_l \in D_l}[-log\ p(y_l|x_l;\theta)] + \lambda\ \mathbb{E}_{x_u \in D_u}\mathbb{E}_{y_u \sim p(y|x_u;\theta^*)}[-log\ p(y_u|x_u;\theta)] \tag{4.3}$$

where $p(y|x;\theta)$ is the conditional distribution under model parameters θ. θ^* is given by the model parameters from the last iteration and fixed in the current iteration. Similar optimization functions have been used recently in variants of self-training for neural sequence generation [65], data augmentation [185] and knowledge distillation. The details of this process are shown in Fig. 4.3 with the pseudo-code in Algorithm 3.

4.2.4 Complexity analysis

In the proposed Vec2Node pipeline the main computation module is CP decomposition. In general, CP/PARAFAC is shown to be in the order of the number of non-zero elements [10] in the tensor. In other words, CPD is very fast and efficient for sparse tensors which is the case in this work due to sparsity of word co-occurrence. Meanwhile, some methods have been proposed for CPD which are amenable to hundreds of concurrent threads while maintaining load balance and low synchronization costs [145]. It is also worth mentioning that CPD is an offline step in the Vec2Node framework which i.e., we only execute it once to obtain the embeddings and we do not

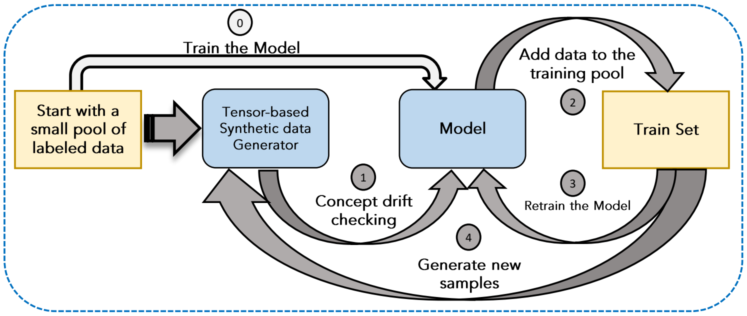

0
Train the Model
Add data to the training pool
2
Start with a small pool of labeled data
Tensor-based Synthetic data Generator
Model
Train Set
1
Concept drift checking
3
Retrain the Model
4
Generate new samples

Figure 4.3: Few-shot self-training with data augmentation and consistency learning to prevent concept drift.

Algorithm 3 Self-train Vec2Node

```
Input : Base model $M$, small labeled set $D_l$.
Return : Self-trained $M$.

1. Slide a window of size $w$ across the text of each sample in $D_l$, capture co-occurring words to create a co-occurrence matrix for each sample.
2. Stack all co-occurrence matrices to create a 3-mode tensor $\mathcal{X}$ of size $T \times T \times S$.
3. Decompose $\mathcal{X}$ into $\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$
4. Use $\mathbf{A}$,$\mathbf{C}$ to model the corpus using graph / hypergraph representations.
5. Calculate Euclidean distances between the nodes to find the closest words.
6. Train $M$ using $D_l = \{x_l, y_l\}$. Set $D = D_l$.
7. While not converged
    • For $\{x_l, y_l\} \in D$, generate augmented samples $D'_u$ by replacing closest words.
    • Assign pseudo-label $y_u$ to each sample $x_u \in D'_u$ using Equation 4.2.
    • If $y_l = y_u$ then $D = D \bigcup \{x_u, y_u\}$.
    • Retrain $M$ using augmented data $D$ using Equation 4.3.
8. Return model $M$
```

need to repeat it for generating new samples.

4.3 Experimental evaluation

4.3.1 Experimental setup

Hyper-parameter configurations. We perform grid search for the window size $w \in [3-10]$, $R \in [5-150]$, with the best value as $w = 3$. Similarly, we found the best value for the decomposition rank $R = 25$. We also perform a search for the word replacement ratio – where we replace words with their nearest neighbors in the embedding space to generate augmented samples (as outlined in Section 2). We observe that replacing $20-25$ % of the words results in fewer number of concept drift which consequently leads to the best accuracy. We also explored the possibility of consecutive vs. random word replacement. We found the classification accuracy of the former to be 3% higher as it preserves the context resulting in $10-12$ % less concept drift. Reported experimental results are averaged over 25 runs with different random seeds.

To investigate the performance of the Vec2Node, we assess its different components, namely, tensor embedding, KNN tensor graph, hypergraph, and self-training mechanism for few label classification against the following baselines.

Base models for classification

We experiment with the following base classifiers.

fastText is an efficient word embedding which is an extension of the word2vec model. fastText represents each word as an n-gram of characters, thereby, working well with rare words; whereas, other non-contextualized embeddings such as GloVe and Word2Vec fail to provide representations for unseen words [18, 77]. Considering this advantage of fastText over mentioned embeddings, we choose it as one of our base classifiers.

BERT advantages contextualized representations leveraging deep bidirectional transformers. We experiment with the pre-trained checkpoints from HuggingFace [182] transformers[2].

Word replacement techniques

We experiment with the following augmentation techniques to investigate the efficacy of the tensor embedding in our proposed Vec2Node framework. For all of the following techniques we retain all components of the proposed Vec2Node except the replacement technique.

Random replacement. In our Vec2Node framework, we replace the tensor embedding based augmentation with random word replacement along with self-training and consistency learning.

tf-idf We generate a KNN graph leveraging tf-idf measure. To do so, we first create the tf-idf matrix and decompose it into word embeddings using SVD. Similar to the previous setup, we retain other components in Vec2Node and only replace tensor embedding with tf-idf embedding. Both random replacement and tf-idf, with strong data augmentation and self-training techniques have been shown to obtain very competitive results for text classification [185].

fastText embedding. Not only do we use fastText for classification but also we replace the tensor embedding with fastText embedding to find the most similar words. we retain other components such as graph representation, concept drift checking and self-training in Vec2Node.

Word2Vec embedding. A shallow 2-layers neural network proposed by Google [93]. We use Word2Vec instead of tensor embedding to find the most similar words using cosine similarity. Similar to the previous setup, we retain other components in Vec2Node and only switch the augmentation method.

[2] https://github.com/huggingface/transformers

Dataset	**Class**	**Train**	**Test**	Avg. Words/Doc
SST2	2	67340	872	17
IMDB	2	25000	25000	235
AG News	4	12000	7600	40

Table 4.1: Dataset statistics.

Dataset	**%Train**	**#Train**	**fastText**	**Vec2Node**	**Average**
SST2	1	673	0.509±0.000	**0.638±0.0007**	**5.46↑**
	5	3367	0.710±0.100	**0.740±0.004**	
	100	67340	0.818±0.0018	**0.823±0.0006**	
IMDB	1	250	0.499±0.000	**0.605±0.004**	**10.26↑**
	5	1250	0.522±0.012	**0.718±0.001**	
	100	25000	0.857±0.0007	**0.863±0.002**	
AG News	1	1200	0.295±0.003	**0.687±0.023**	**18.56 ↑**
	5	6000	0.663±0.001	**0.825±0.002**	
	100	12000	0.900±0.0003	**0.903±0.0008**	

Table 4.2: Performance of fastText and Vec2Node with varying amounts of labeled training data. Vec2Node uses the fastText classifier (logistic regression) with tensor data augmentation.

Hypergraph Vs. graph representation

Hypergraph We also compare tensor embedding based KNN graph representation with hypergraphs [147]. Here, we use tensor embeddings for measuring word affinity as described in Section 4.2.2. Similar to the above setup, we only change the similarity representation to use KNN tensor or hyperpgraphs while keeping other components the same.

4.3.2 Evaluation

We experiment on SST2 [146], IMDB [106] and AG News [193] to investigate the performance of Vec2Node on short, long and multi-label textual datasets, respectively with statistics in Table 4.1. We report results on the corresponding test splits as available from the above works.

Dataset	%Train	#Train	BERT	Vec2Node
SST2	0.5	60	0.754	**0.826**
IMDB	0.5	125	0.776	**0.783**
AG News	0.5	600	0.869	**0.880**

Table 4.3: Performance of fastText and BERT with varying amounts of labeled training data. Vec2Node uses BERT encoder with tensor data augmentation.

Vec2Node with different base models

From Table 4.2 we observe that Vec2Node with tensor data augmentation obtains 21.5% and 13.5% improvement over fastText while using only 1% and 5% of labeled training data. In this experiment, Vec2Node is built on top of fastText to demonstrate the strength of augmentation. We also observe the relative improvement with augmentation to significantly increase with longer text. For example, the improvement in accuracy for IMDB is 15% more than that on SST2 dataset using 5% of labels. This could be attributed to the shorter context samples not being able to generate diverse variety of synthetic samples that are significantly different from the original ones. However, we still demonstrate significant accuracy improvement with augmentation on SST2. As illustrated, when we use 100% of the training data, we still observe improvement in classification accuracy which demonstrates the effectiveness of tensor augmentation in both low and high-resource settings.

In contrast to fastText, the BERT model is pre-trained over massive amounts of unlabeled data, and therefore, works well even in the low-data regime. Therefore, to demonstrate the strength of our tensor augmentation, we choose the few-shot setting with only 0.5% of labeled training data. From Table 4.3, we observe Vec2Node using BERT as an encoder along with tensor augmentation to obtain 3.5% improvement in average over the base BERT model using very few training labels. Meanwhile, in case of augmenting SST2, using BERT as a classifier improves the

			Vec2Node	
Dataset	**#Train**	**fastText**	**Hypergraph Aug.**	**KNN Aug.**
SST2	3367	0.710±0.100	0.722±0.003(1.2↑)	**0.740±0.004(3↑)**
IMDB	1250	0.522±0.012	0.664±0.004(14.2↑)	**0.718±0.001(19.6↑)**
AG News	6000	0.663±0.001	0.811±0.002(14.8↑)	**0.825±0.001(16.2↑)**

Table 4.4: Performance of Vec2Node built on top of fastText with tensor embedding based KNN and hypergraph augmentation on 5% of labeled training data.

overall performance of Vec2Node. Therefore, we observe 7% improvement of accuracy after augmentation. It is worth mentioning that the pre-trained BERT performs much better than fastText that is trained from scratch. However, in both the settings, Vec2Node outperforms each of the models by incorporating tensor data augmentation into the framework.

Tensor embedding based KNN and hypergraph representation

From Table 4.4, we observe that Vec2Node using tensor embedding based KNN and hypergraph representations to outperform the base classifier (without augmentation) on all the datasets. We also observe the performance of Vec2Node using KNN tensor graph to be higher than that with the hypergraph representation. We noticed that the KNN graph captures globally similar words, no matter whether they co-occur in sentences with same labels or not. On the other hand, the hypergraph representation confines the similarity search to words co-occurring in texts with similar labels. In fact, by limiting the replacement candidates to words occurring in similar sentences, the hypergraph approach makes repetitive substitutions for a target word due to a limited candidate pool.

			Vec2Node	
Dataset	#Train	fastText	w/o ST & CL	w/ ST & CL
SST2	3367	0.710±0.100	0.720±0.006(1↑)	**0.740±0.006(3↑)**
IMDB	1250	0.522±0.012	0.686±0.005(16.4↑)	**0.718±0.001(19.6↑)**
AG News	6000	0.663±0.001	0.791±0.001(12.8↑)	**0.825±0.001(16.2↑)**

Table 4.5: Performance of Vec2Node built on top of fastText with and without self-training & consistency learning (ST & CL) on 5% labeled training data.

Self-training with consistency learning

In this experiment, we ablate the self-training and consistency learning components in Vec2Node to analyze their contribution with results in Table 4.5. We observe the self-training component where the model leverages augmented data and pseudo-labels for consistency learning to further improve the performance of Vec2Node by roughly 10% on all datasets. Also, this component along with augmentation jointly contributes to roughly 13% improvement of Vec2Node over that of fastText.

Augmentation strategies

Table 4.6 shows the performance of Vec2Node with different word replacement strategies including random, tf-idf, and tensor embeddings. We observe that Vec2Node performs the best with the tensor embedding augmentation strategy, while in average outperforming random and tf-idf replacement by 3% and 5% respectively. Random and tf-idf word replacement strategies do not consider the local and global contextual information of the target word during replacement, and, consequently, generate noisy samples.

Table 4.7 illustrates performance of Vec2Node with different embedding methods including fastText, Word2Vec and tensor embeddings. As reported earlier, with longer sequence

		Vec2Node		
Dataset	**#Train**	**tf-idf**	**Random**	**Tensor**
SST2	3367	0.733±0.004 (2.3↑)	0.737±0.001(2.7↑)	**0.740±0.004(3↑)**
IMDB	1250	0.602±0.021(7.9↑)	0.659±0.013(13.7↑)	**0.718±0.001(19.6↑)**
AG News	6000	0.807±0.002(14.3↑)	0.799±0.002(13.6↑)	**0.825±0.001(16.2↑)**

Table 4.6: Performance of Vec2Node with different word replacement and augmentation strategies built on top of fastText using 5% of labeled data.

		Vec2Node		
Dataset	**#Train**	**Word2Vec**	**FastText**	**Tensor**
SST2	3367	**0.759±0.03(4.9↑)**	0.730±0.025(2↑)	0.740±0.004(3↑)
IMDB	1250	0.663±0.01(14.1↑)	0.680±0.045(15.8↑)	**0.718±0.001(16.2↑)**
AG News	6000	0.806±0.042(14.3↑)	0.810±0.054(14.7↑)	**0.825±0.001(16.2↑)**

Table 4.7: Performance of Vec2Node with different embedding methods using 5% of labeled data.

texts as in IMDB and AG News, Vec2Node with tensor embeddings outperforms other word embedding methods due to more tangible word co-occurrences in the sentences. SST2 samples consist of short phrases with fewer co-occurring non-stop words resulting in less diverse augmented examples in contrast to those in longer text sequences. Overall, we observe tensor embeddings to outperform fastText and Word2Vec embeddings.

4.3.3 Interpretability and examples

Table 4.8 shows augmentation examples from Vec2Node using different strategies like random, tf-idf and tensor embedding based word replacement from the AG news and SST2 datasets. We observe Vec2Node to generate better augmentations with the following features.

Preserving context for word replacement. In contrast to random selection which blindly substitutes words, the co-occurrence based structure of the tensor embedding preserves the context, and selects candidate words that are contextually similar to the original ones. For instance, in example

#	Original-Label	Augmented-Method	Closest Tensor Neighbors
1	Jermain Defoe may replace him for England on Saturday-**Sports**	Owen Michael may replace him for England on Saturday-**Tensor** Jermain Defoe may replace him for England vast desert Saturday-**Random** Jermain Defoe may replace 1.22 for England 0-11 Saturday 1,070-**tf-idf**	Jermain ↪ Owen Defoe ↪ Michael
2	The Samsung SCH-S250 5-megapixel camera phone will enable users to take photos-**Sci/Tech**	SCH-S250 Samsung the 5-megapixel camera phone will enable users to take photos-**Tensor** seem dominance in hurricane season SCH-S250 Samsung The will enable users to take photos **-Random** barbarians SCH-S250 Samsung The will enable users to take photos-**tf-idf**	Samsung ↪ SCH-S250 SCH-S250 ↪ Samsung
3	Pakistan has arrested at least five al-Qaida-linked-**World**	Pakistan has arrested at athens 10 al-Qaida-linked **-Tensor** Pakistan has arrested .26 pairings fatter al-Qaida-linked-**Random** Pakistan has 0.84 , 1,000-yard 1,000-yard al-Qaida-linked-**tf-idf**	least ↪ athens five ↪ 10
4	Working from a surprisingly sensitive script co-written by gianni romoli...**Positive**	add a surprisingly sensitive script co-written by gianni romoli... **Tensor** working from a surprisingly sensitive script co-written by gianni mixed second-rate **-Random** working from a surprisingly sensitive script co-written by becalmed chillingly ...**-tf-idf**	working ↪ add from ↪ ’ ’
5	Never seems hopelessly juvenile .-**Negative**	Never seems amateurishly accused-**Tesnor** Never seems dawn carmichael-**Random** Never seems born actioners **-tf-idf**	hopelessly ↪ amateurishly juvenile ↪ accused

Table 4.8: Snapshot of similar contexts captured with tensor embedding and augmented sentences created by replacing 20-30% words in a sentence.

#1 the entity “Jermain Defoe” is replaced by “Owen Michael” as they are more likely to co-occur in a Sport text related to “Real Madrid”. As illustrated, the other approaches replace words quite randomly. This feature helps to minimize the concept drift that might happen due to the replacement process.

Paraphrasing context. Vec2Node leverages a sliding window to capture co-occurring concepts in a sentence, such that non-adjacent words that occur within the same context can be substituted with each other. This contributes to paraphrased sentences generated during augmentation as illustrated in example #2 with re-ordered proper nouns “Samsung” and “SCH-S250”.

Tensor embedding preserves word-level similarities. Tensor embedding not only preserves the context-level similarity but also retains the semantics of the replaced concept. More precisely, it is more likely that a number gets replaced by another number (# 3) or an adverb by another adverb (# 5), and so on. We observe that not only numbers and verbs, but also prepositions like “a”, “an”, and “the” are replaced with similar concepts in the synthetic samples while preserving the context.

4.4 Related work

4.4.1 Self-training and few-shot learning

Self-training is one of the well-known semi-supervised learning approaches which has been widely used to minimize the need for annotation leveraging large-scale unlabeled data [65, 97]. Recent work like MetaST [178] leverage self-training and meta-learning for few-shot training of neural sequence taggers. Another recent work, Unsupervised Data Augmentation (UDA) [185] leverages consistency learning with paraphrasing and back-translation (BT) from Neural Machine Translation systems for few-shot learning.

4.4.2 Tensor embedding for text Classification

In recent years, tensors have been used for semi-supervised or unsupervised text classification tasks. For instance, in [2, 52] tensor embedding along with belief propagation have been used for semi-supervised classification of news articles. Prior to the aforementioned works, a co-occurrence tensor was used for unsupervised soft co-clustering of news articles in [71].

4.5 Conclusions

In this work, we propose a novel tensor embedding based technique i.e., Vec2Node, for augmenting textual datasets leveraging local and global information in corpus. Vec2Node leverages tensor data augmentation with self-training and consistency learning for text classification with few labels. Our experiments demonstrate that synthetic data generated by Vec2Node are interpretable and improve the classification accuracy over different datasets significantly in low-resource settings. For instance, Vec2Node improves the accuracy of fastText by 21.5% while using only 1% of labeled data. Overall, we demonstrate Vec2Node to work well both in low and high-data regime with improved performance when built on top of different encoders (e.g., fastText, BERT).

5

A Hybrid Summarization Approach for Extracting Misinformative Key Phrases

As it was discussed in the chapter 1, fake news intervention research aims to reduce the effects of fake news by leveraging proactive intervention methods that try to minimize the scope of spread or reactive intervention methods which are applied after fake news goes viral. In this chapter, we propose an intervention method which aims to extract key phrases of tweets including a link to a fake article to help both users and fact checkers to recognize and discriminate similar topics, hate or bias languages and use the list of keywords to 1) highlight suspicious key phrases and 2) use the list of key phrases as a baseline for recognizing similar content in unlabeled tweets.

5.1 Introduction

Fact checkers and fake news detectors, especially crowd-source services aim to ”understand the content and main idea” of a text and annotate articles based on key information and salient details. In other words, they summarize the content of an article abstractly or concretely and extract key points in order to reduce the amount of unnecessary information and then make a decision about the category of fake news that the article belongs to.

With that being said, generating meaningful summarization has been of great importance for variety of Natural Language Processing (NLP) tasks such as social media, medical or educational analysis [8]. Generally speaking, text summarization methods can be categorized into two main classes: extractive summarization and abstractive summarization. Extractive summarization creates summaries by extracting salient phrases from the original text, whereas, abstractive summarization creates new phrases by internal semantic representation of the original text. Due to the fact that abstractive methods paraphrase the text, the resulting summary is much closer to the human summarization while being more challenging and complicated task [110].

In this chapter, we aim to introduce a hybrid framework which incorporates both techniques for extracting key phrases that are commonly used in different categories of fake news i.e, Fake, Satire, Bias, Conspiracy, Rumor, JunkScience, Hate, Clickbait, unreliable. By doing so, not only we make the crowd-source annotation process easier for the fact checkers and experts but also we provide interpretable key phrases that could be leveraged for finding common language patterns in different topics, highlighting bias or possible anomalies in the annotating process.

5.2 Background

As mentioned earlier, automatic text summarization, is the process of creating a short and coherent version of a longer document while preserving the most important parts [8, 48, 181]. We as human beings are quite capable at this task as it involves the following:

- Understanding the meaning of the text
- Capturing salient details and rephrasing them into the summary.

As we discussed earlier, there are two main techniques for text summarization: extractive and abstractive methods:

- **Extractive text summarization:** refers to the process of selecting phrases and sentences from the original text to create the summary. Extractive summarization involves ranking the phrases based on the importance and relevance in order to choose candidates for extraction.
- **Abstractive techniques:** are the category of approaches that involve in generating entirely new phrases and sentences to capture the meaning of the source document. This category is more similar to the human summarization process and definitely more challenging [8, 181].

5.2.1 Comparison of extractive vs. abstractive summarization

As mentioned above, extractive methods usually work based on phrase ranking which prerequisites ranking the words within the phrases as well. Thus, this category of techniques are not often able to distinguish words and in turn phrases that convey the same meaning and as a result of this incapable of abstracting contextually similar sentences into a shorter one. On the other hand, due to the fact that abstractive techniques operate based on rephrasing and contextual meanings, they are very successful in summarizing documents that comprise multiple sentences that are similar and convey pretty much the same information. However, if sentences of a document are irrelevant, e.g., the document is created by concatenating sentences from different contexts they usually fail and in such situations, sometimes extractive methods are more successful. In the next sections, we discuss two robust and commonly used techniques of each category.

5.2.2 Rapid Automatic Keyword Extraction (RAKE) for Extractive Summariation

Rapid Automatic Keyword Extraction or RAKE is an extremely efficient keyword extraction algorithm which could be applied on new domains easily especially when the text follows specific grammar conventions. Rake uses grammatical conventions such as keywords frequently contain multiple words with standard punctuation, stop words or propositional words that have minimum lexical meaning and could be discarded. Therefore, stop words are typically removed in various text analyses as they are considered to be pretty much meaningless. Words that carry a meaning related to the text are considered as content bearing and are called content words [125].

RAKE parses the text and partition it into candidate key phrases with the help of stop words and delimiters. More preciously, first the text is split into an array of words using delimiters, and secondly, the array is again split into a sequence of contiguous words at phrase delimiters and stop word positions. Lastly, the words that lie in the same sequence are assigned the same position in the text and together are considered as a candidate key phrases. After identifying all the candidate key phrases, a graph of word co-occurrence is generated which represents a score for each candidate phrase i.e., keyword score. The keyword score is calculated based on the degree and frequency of the vertices in the graph as follows:

$$\text{Keyword Score} = \Sigma_{w_i \in W} deg(w_i)/freq(w_i)$$

Where $deg(w_i)$ and $freq(w_i)$ denote the degree and frequency of word w_i in set member W respectively. After the candidate keyword score is calculated, the top T candidate are selected from the document. T is usually one-third the number of words in the graph. Algorithm 4 demonstrates the ranking process of RAKE.

5.2.3 BERT transformer for abstractive summarization

BERT is a powerful and effective model for a variety of Natural Language Processing (NLP related tasks). BERT's key feature is the bidirectional training of Transformer i.e., a popular attention model for language modelling. It has been shown that a language model which is bidirectionally trained better captures the language context than a single-direction language model [37].

As far as summarization task is concerned, the BERT model could generate sentence embeddings for multiple sentences. This is done by inserting [CLS] token before the start of the first sentence. The output is then a sentence vector for each sentence. Then, to extract category level features, the sentence vectors are passed through multiple layers. The generated summary evaluated against a ground truth and the loss is used to train both the summarization layers and the BERT model [102]. A simplified BERT architecture used for summarization task is demonstrated in Fig. 5.1.

5.2.4 Crowd source article annotation

As mentioned earlier, collecting human annotation for misinformation detection is a challenging and time-consuming task. However, there exist some crowd-sourced schemes such as the "B.S. Detector" which provide human annotated labels. B.S. Detector is a browser extension for both Chrome and Mozilla-based browsers. It searches all links on a given webpage for references to unreliable sources by checking against a manually created list of domains.Then,it provides visual warnings about the presence of questionable links or the browsing of questionable websites. B.S. detector categorizes article domains as[1] one of the following categories:

- **Fake News:** Sources that fabricate stories out of whole cloth with the intent of pranking the public.
- **Satire:** Sources that provide humorous commentary on current events in the form of fake news.
- **Extreme Bias:** Sources that traffic in political propaganda and gross distortions of fact.
- **Conspiracy Theory:** Sources that are well-known promoters of kooky conspiracy theories.
- **Rumor Mill:** Sources that traffic in rumors, innuendo, and unverified claims.
- **State News:** Sources in repressive states operating under government sanction.

[1] https://github.com/selfagency/bs-detector

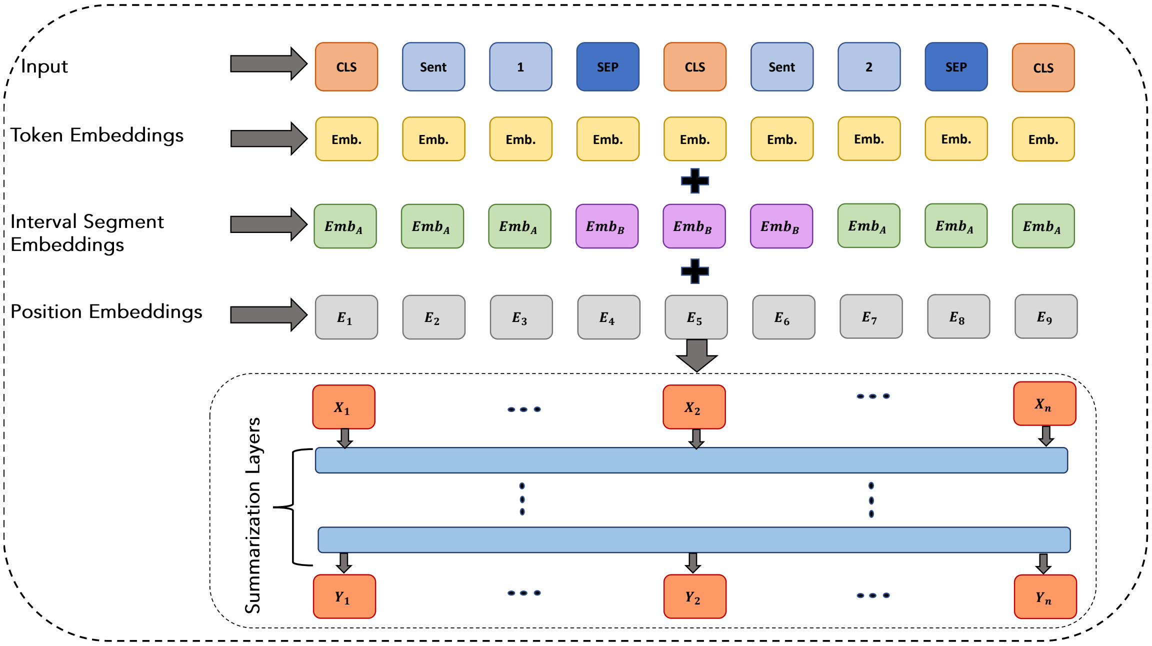


Figure 5.1: The overview architecture of BERT for summarization [102].

- **Junk Science:** Sources that promote pseudoscience, metaphysics, naturalistic fallacies, and other scientifically dubious claims.
- **Hate Group:** Sources that actively promote racism, misogyny, homophobia, and other forms of discrimination.
- **Clickbait:** Sources that are aimed at generating online advertising revenue and rely on sensationalist headlines or eye-catching pictures.
- **Proceed With Caution:** Sources that may be reliable but whose contents require further verification

Algorithm 4 RAKE for Scoring Keywords

```
Input: phrase list $P = [p_1 . . . . , p_n]$
Output: word scores $S = [s_1, . . . , s_n]$
Initialize word-freq,word-deg,S
for $p_i \in P$ do
    W =tokenize($p_i$)
    deg(W) = len(W)-1
    for $w_j \in W$ do
        freq($w_j$) += 1
        word-freq.add(freq($w_j$))
        deg($w_j$) += deg(W)
        word-deg.add(deg($w_j$))
    end
end
for $w_i \in word - freq$ do
    deg($w_j$) =deg($w_j$) +freq($w_j$)
end
for $w_i \in word - freq$ do
    $s_j$ =deg($w_j$) / freq($w_j$)
end
```

5.3 Methodology

In this section, we discuss the problem formulation and proposed approach.

5.3.1 Problem formulation

Given N tweets including a link to articles which are labeled as class **L** where

L ∈ [*Fake, Satire, Bias, Conspiracy, Rumor, JunkScience, Hate, Clickbait, unreliable*],

Extract key phrases whithin the tweets that correspond to each class in **L**.

5.3.2 Proposed method

Motivation:

A large set of tweets definitely consists of a variety of irrelevant sentences shared by different users for multiple topics. In this situation, extractive summarization is a proper option for extracting key phrases. However, sometimes especially when a topic is going viral, it is very likely that multiple users share the same news article repeatedly while paraphrasing the title or main points of the news. In this case, abstractive summarization is immensely useful. Due to the fact that social media content analysis usually encounter both scenarios above, we propose a hybrid approach to take advantage of both techniques. In fact, we aim to cover as many topics as we can and at the same time abstract redundant and similar information into a salient yet shorter text. Our proposed hybrid approach to address the problem stated above, includes the following steps:

Data gathering and crowd-source article annotation

First, we crawl Twitter to create a dataset out of tweets published in a time span of 3 months including links to a news article. Then, we leverage B.S. Detector scheme to annotate the articles that are extracted from the crawled tweets.

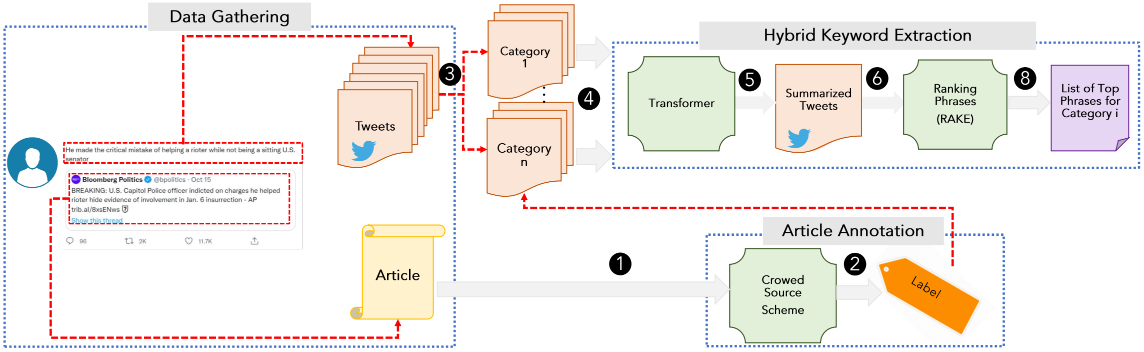
Automatic summarization of N tweets for each category

There is a sheer amount of textual content that users share on social media, and it is growing every single day. So, the set of textual content shared by users is pretty much unstructured and in some cases even redundant. Having this in mind, there is a great need to reduce much of this text data to a shorter and more concise text that captures the key information and salient details.

In this work, we take a hybrid approach. More precisely, first, we summarize users tweets and then we extract top **K** phrases out of the summarized tweets using a ranking technique. By doing so, we make sure that we are extracting the most important parts of the text. Our hybrid approach consists of:

- **Abstractive summarization of tweets using BERT** After annotating tweets using B.S. Detector and categorizing them into categories we discussed in the background section, we leverage BERT transformer to summarize each categories into a shorter and more concise text.

- **Ranking key phrases and extracting top ones** We apply RAKE Algorithm on summarized text and extract top **K** phrases for each summarized categories. An overview of the proposed method is depicted in Fig. 5.2.


He made the critical mistake of helping a rioter while not being a sitting U.S. senator

Bloomberg Politics @bpolitics · Oct 15
BREAKING: U.S. Capitol Police officer indicted on charges he helped rioter hide evidence of involvement in Jan. 6 insurrection - AP trib.al/8xsENws
Show this thread

Figure 5.2: An overview of the proposed hybrid approach to extract key phrases of each fake news category.

5.4 Evaluation

In this section, first we discuss the setting we experiment on and then we present the key phrases corresponding to each category using the proposed hybrid technique.

5.4.1 Experimental Setting

As mentioned in the data gathering step, we crawled Twitter and extracted tweets including article links that were published in a time interval of three months and leveraged B.S. Detector to annotate the article and the tweets that shared them. The distribution of different categories in our dataset is demonstrated in Table 5.1.

To implement the hybrid framework, we leveraged transformer version 4.9.2 and bert-extractive-summarizer version 0.8.1. To implement the RAKE method we used python-rake library 1.4.4 and rake-nltk 1.0.4.

Category	Number of Samples
Clickbait	54698
Conspiracy	16328
Bias	14915
Hate	11973
Satire	5734
Fake	3378
Unreliable	2401
Junk Science	1743
Rumor	226

Table 5.1: Distribution of different categories in our dataset.

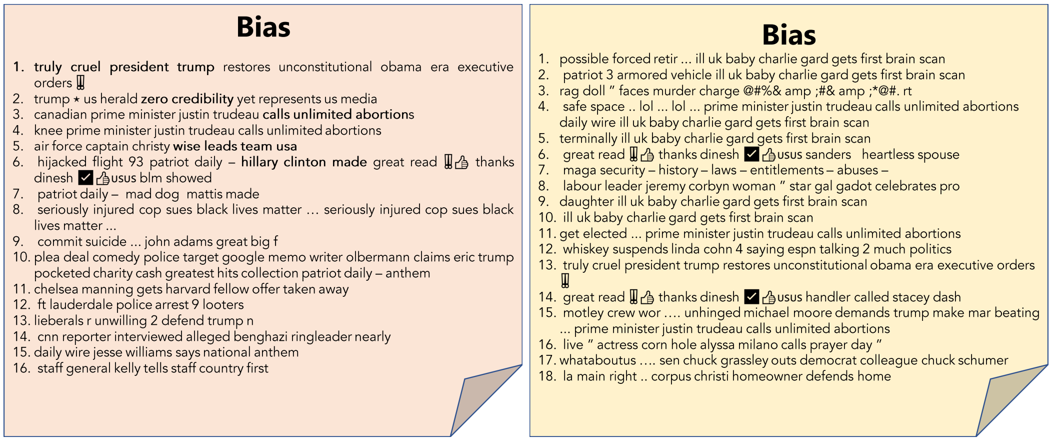
5.4.2 Experimental Result

To investigate the efficacy of hybrid approach i.e., ranking after summarization against ordinary ranking, we extract top key phrases of both summarized and original text and compare the results. Extracted key phrases of both approach on 9 different categories of fake news are illustrated in Fig. 5.3 to 5.11.

If we look closely at these figures, we observe that for multiple categories e.g., junk science and hate, using summarization removes the redundant phrases even if they are not quite the same. This is an extremely important feature because users usually use different languages and often paraphrase the main message of an article when sharing it. Using summarization enables us to abstract all similar phrases that convey the same message into a shorter text. Moreover, we observe that sometimes summarization helps us to capture more contextually important phrases that we would not be able to extract while using RAKE solely. For example, in junk science category, there are multiple of redundant phrases that are not salient for the readers. On the contrary, the majority of phrases that are captured by the proposed hybrid approach, are more salient and understandable.

5.5 Related Work

Text summarization for fake news detection The overwhelming number of news articles that users share across different social media platforms everyday, has brought scientists' attention into the usage of summarization techniques for fake news detection. For instance, Esmaeilzadeh et. al. [43] leverage abstractive text summarization by different deep learning models such as LSTM-encoder-decoder with attention, pointer-generator networks, coverage mechanisms, and transformers as a feature extractor for fake news detection task where the news articles prior to classification will be summarized. In another work [64], Hartl et. al. compress the original article into some form of automatically generated summary before classifying it. Moreover, Kim et. al. use summarization technique and propose a novel graph-based method [85] to detect misinformation. Their proposed method represent the relationship between all sentences using a graph, and exploits an attention mechanism to compute the reflection rate of contextual information among sentences.



Bias

1. **truly cruel president trump** restores unconstitutional obama era executive orders ‼
2. trump ⋆ us herald **zero credibility** yet represents us media
3. canadian prime minister justin trudeau **calls unlimited abortions**
4. knee prime minister justin trudeau calls unlimited abortions
5. air force captain christy **wise leads team usa**
6. hijacked flight 93 patriot daily – **hillary clinton made** great read ‼👍 thanks dinesh ✅👍usus blm showed
7. patriot daily – mad dog mattis made
8. seriously injured cop sues black lives matter … seriously injured cop sues black lives matter …
9. commit suicide … john adams great big f
10. plea deal comedy police target google memo writer olbermann claims eric trump pocketed charity cash greatest hits collection patriot daily – anthem
11. chelsea manning gets harvard fellow offer taken away
12. ft lauderdale police arrest 9 looters
13. lieberals r unwilling 2 defend trump n
14. cnn reporter interviewed alleged benghazi ringleader nearly
15. daily wire jesse williams says national anthem
16. staff general kelly tells staff country first

(a) With Summarization

Bias

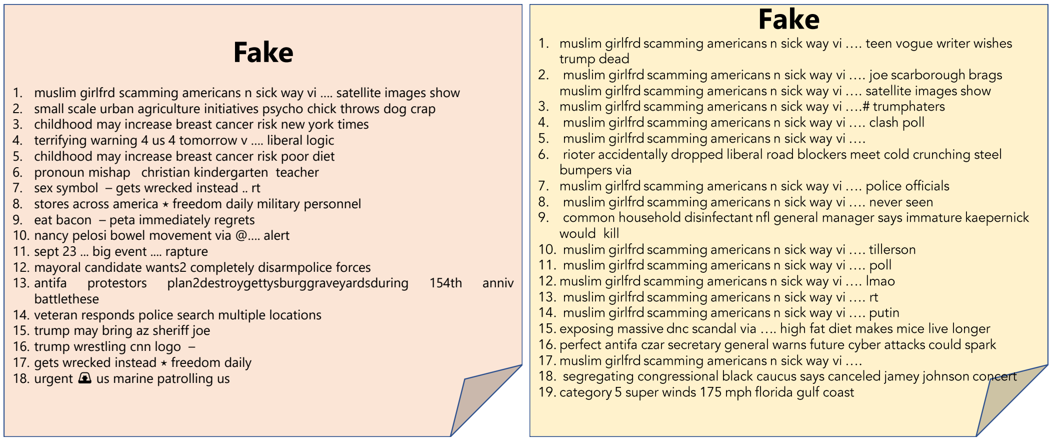
1. possible forced retir … ill uk baby charlie gard gets first brain scan
2. patriot 3 armored vehicle ill uk baby charlie gard gets first brain scan
3. rag doll ” faces murder charge @#%& amp ;#& amp ;*@#. rt
4. safe space .. lol … lol … prime minister justin trudeau calls unlimited abortions daily wire ill uk baby charlie gard gets first brain scan
5. terminally ill uk baby charlie gard gets first brain scan
6. great read ‼👍 thanks dinesh ✅👍usus sanders heartless spouse
7. maga security – history – laws – entitlements – abuses –
8. labour leader jeremy corbyn woman ” star gal gadot celebrates pro
9. daughter ill uk baby charlie gard gets first brain scan
10. ill uk baby charlie gard gets first brain scan
11. get elected … prime minister justin trudeau calls unlimited abortions
12. whiskey suspends linda cohn 4 saying espn talking 2 much politics
13. truly cruel president trump restores unconstitutional obama era executive orders ‼
14. great read ‼👍 thanks dinesh ✅👍usus handler called stacey dash
15. motley crew wor …. unhinged michael moore demands trump make mar beating … prime minister justin trudeau calls unlimited abortions
16. live ” actress corn hole alyssa milano calls prayer day ”
17. whataboutus …. sen chuck grassley outs democrat colleague chuck schumer
18. la main right .. corpus christi homeowner defends home

(b) Without Summarization

Figure 5.3: Top key phrases extracted from summarized ”Bias” category.

5.6 Conclusions

In this chapter, we propose a hybrid approach for extracting key phrases of different categories of misinformation in order to recognize misinformative parts and diffuse further spread of misinformation. We propose a hybrid approach that leverages both extractive and abstractive summarization techniques. More precisely, we propose using BERT transformer to summarize tweets that share a certain type of misinformation into a shorter and more abstract summary and then extract key phrases with Rapid Automatic Keyword Extraction (RAKE) algorithm. Our results illustrates that using the hybrid approach not only removes the redundant phrases but also achieves more salient and understandable phrases in comparison to when we extract key phrases of original tweets.



Fake

1. muslim girlfrd scamming americans n sick way vi .... satellite images show
2. small scale urban agriculture initiatives psycho chick throws dog crap
3. childhood may increase breast cancer risk new york times
4. terrifying warning 4 us 4 tomorrow v .... liberal logic
5. childhood may increase breast cancer risk poor diet
6. pronoun mishap christian kindergarten teacher
7. sex symbol – gets wrecked instead .. rt
8. stores across america ⋆ freedom daily military personnel
9. eat bacon – peta immediately regrets
10. nancy pelosi bowel movement via @.... alert
11. sept 23 ... big event .... rapture
12. mayoral candidate wants2 completely disarmpolice forces
13. antifa protestors plan2destroygettysburggraveyardsduring 154th anniv battlethese
14. veteran responds police search multiple locations
15. trump may bring az sheriff joe
16. trump wrestling cnn logo –
17. gets wrecked instead ⋆ freedom daily
18. urgent 🚨 us marine patrolling us

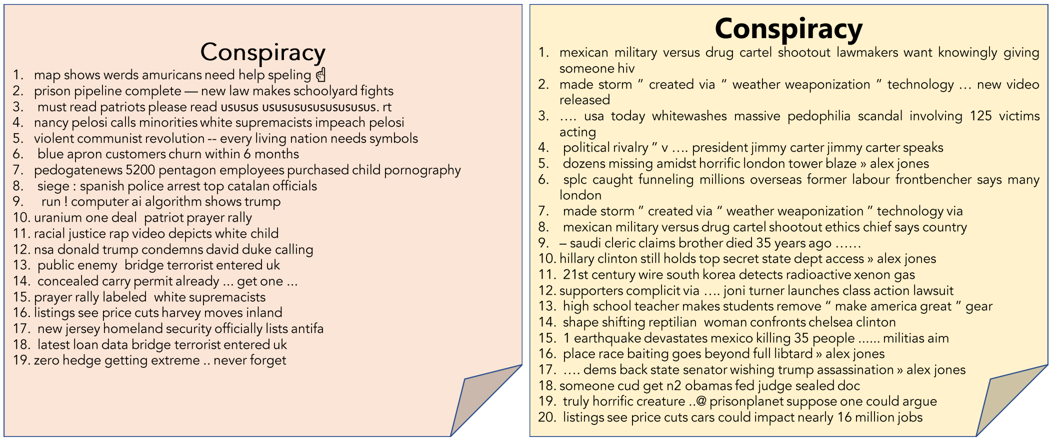
(a) With Summarization

Fake

1. muslim girlfrd scamming americans n sick way vi .... teen vogue writer wishes trump dead
2. muslim girlfrd scamming americans n sick way vi .... joe scarborough brags muslim girlfrd scamming americans n sick way vi .... satellite images show
3. muslim girlfrd scamming americans n sick way vi ....# trumphaters
4. muslim girlfrd scamming americans n sick way vi .... clash poll
5. muslim girlfrd scamming americans n sick way vi ....
6. rioter accidentally dropped liberal road blockers meet cold crunching steel bumpers via
7. muslim girlfrd scamming americans n sick way vi .... police officials
8. muslim girlfrd scamming americans n sick way vi .... never seen
9. common household disinfectant nfl general manager says immature kaepernick would kill
10. muslim girlfrd scamming americans n sick way vi .... tillerson
11. muslim girlfrd scamming americans n sick way vi .... poll
12. muslim girlfrd scamming americans n sick way vi .... lmao
13. muslim girlfrd scamming americans n sick way vi .... rt
14. muslim girlfrd scamming americans n sick way vi .... putin
15. exposing massive dnc scandal via .... high fat diet makes mice live longer
16. perfect antifa czar secretary general warns future cyber attacks could spark
17. muslim girlfrd scamming americans n sick way vi ....
18. segregating congressional black caucus says canceled jamey johnson concert
19. category 5 super winds 175 mph florida gulf coast

(b) Without Summarization

Figure 5.4: Top key phrases extracted from summarized ”Fake” category.



Conspiracy

1. map shows werds amuricans need help speling ☝
2. prison pipeline complete — new law makes schoolyard fights
3. must read patriots please read ususus usususususususus. rt
4. nancy pelosi calls minorities white supremacists impeach pelosi
5. violent communist revolution -- every living nation needs symbols
6. blue apron customers churn within 6 months
7. pedogatenews 5200 pentagon employees purchased child pornography
8. siege : spanish police arrest top catalan officials
9. run ! computer ai algorithm shows trump
10. uranium one deal patriot prayer rally
11. racial justice rap video depicts white child
12. nsa donald trump condemns david duke calling
13. public enemy bridge terrorist entered uk
14. concealed carry permit already ... get one ...
15. prayer rally labeled white supremacists
16. listings see price cuts harvey moves inland
17. new jersey homeland security officially lists antifa
18. latest loan data bridge terrorist entered uk
19. zero hedge getting extreme .. never forget

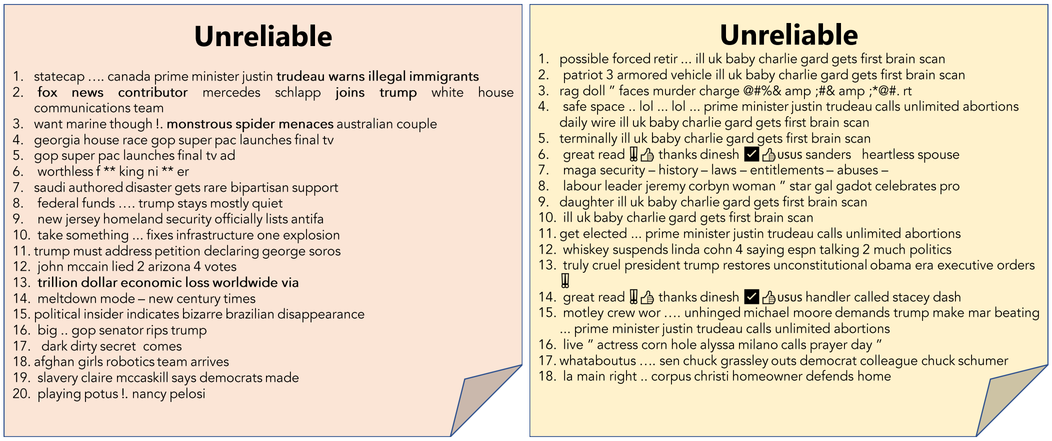
(a) With Summarization

Conspiracy

1. mexican military versus drug cartel shootout lawmakers want knowingly giving someone hiv
2. made storm ” created via ” weather weaponization ” technology ... new video released
3. .... usa today whitewashes massive pedophilia scandal involving 125 victims acting
4. political rivalry ” v .... president jimmy carter jimmy carter speaks
5. dozens missing amidst horrific london tower blaze » alex jones
6. splc caught funneling millions overseas former labour frontbencher says many london
7. made storm ” created via ” weather weaponization ” technology via
8. mexican military versus drug cartel shootout ethics chief says country
9. – saudi cleric claims brother died 35 years ago ......
10. hillary clinton still holds top secret state dept access » alex jones
11. 21st century wire south korea detects radioactive xenon gas
12. supporters complicit via .... joni turner launches class action lawsuit
13. high school teacher makes students remove ” make america great ” gear
14. shape shifting reptilian woman confronts chelsea clinton
15. 1 earthquake devastates mexico killing 35 people ...... militias aim
16. place race baiting goes beyond full libtard » alex jones
17. .... dems back state senator wishing trump assassination » alex jones
18. someone cud get n2 obamas fed judge sealed doc
19. truly horrific creature ..@ prisonplanet suppose one could argue
20. listings see price cuts cars could impact nearly 16 million jobs

(b) Without Summarization

Figure 5.5: Top key phrases extracted from summarized ”Conspiracy” category.



Unreliable

1. statecap …. canada prime minister justin **trudeau warns illegal immigrants**
2. **fox news contributor** mercedes schlapp **joins trump** white house communications team
3. want marine though !. **monstrous spider menaces** australian couple
4. georgia house race gop super pac launches final tv
5. gop super pac launches final tv ad
6. worthless f ** king ni ** er
7. saudi authored disaster gets rare bipartisan support
8. federal funds …. trump stays mostly quiet
9. new jersey homeland security officially lists antifa
10. take something ... fixes infrastructure one explosion
11. trump must address petition declaring george soros
12. john mccain lied 2 arizona 4 votes
13. **trillion dollar economic loss worldwide via**
14. meltdown mode – new century times
15. political insider indicates bizarre brazilian disappearance
16. big .. gop senator rips trump
17. dark dirty secret comes
18. afghan girls robotics team arrives
19. slavery claire mccaskill says democrats made
20. playing potus !. nancy pelosi

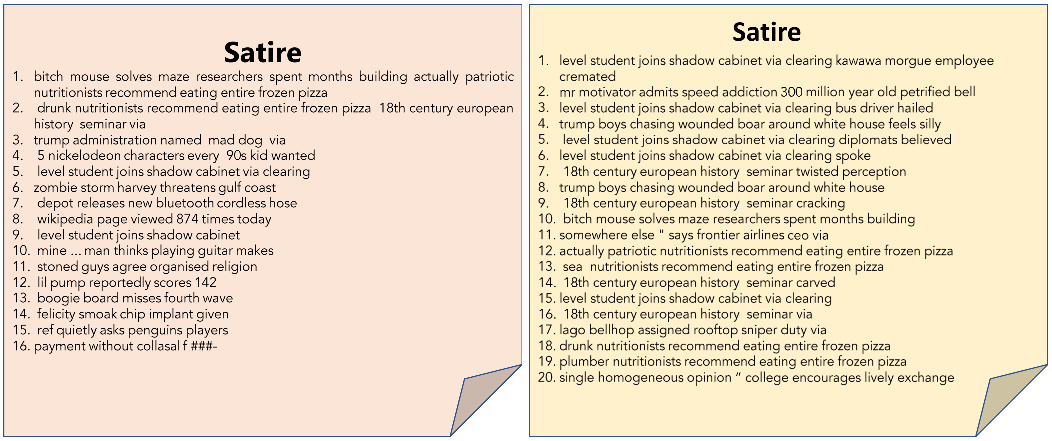
(a) With Summarization

Unreliable

1. possible forced retir ... ill uk baby charlie gard gets first brain scan
2. patriot 3 armored vehicle ill uk baby charlie gard gets first brain scan
3. rag doll ” faces murder charge @#%& amp ;#& amp ;*@#. rt
4. safe space .. lol ... lol ... prime minister justin trudeau calls unlimited abortions daily wire ill uk baby charlie gard gets first brain scan
5. terminally ill uk baby charlie gard gets first brain scan
6. great read ‼👍 thanks dinesh ✅👍usus sanders heartless spouse
7. maga security – history – laws – entitlements – abuses –
8. labour leader jeremy corbyn woman ” star gal gadot celebrates pro
9. daughter ill uk baby charlie gard gets first brain scan
10. ill uk baby charlie gard gets first brain scan
11. get elected ... prime minister justin trudeau calls unlimited abortions
12. whiskey suspends linda cohn 4 saying espn talking 2 much politics
13. truly cruel president trump restores unconstitutional obama era executive orders ‼
14. great read ‼👍 thanks dinesh ✅👍usus handler called stacey dash
15. motley crew wor …. unhinged michael moore demands trump make mar beating ... prime minister justin trudeau calls unlimited abortions
16. live ” actress corn hole alyssa milano calls prayer day ”
17. whataboutus …. sen chuck grassley outs democrat colleague chuck schumer
18. la main right .. corpus christi homeowner defends home

(b) Without Summarization

Figure 5.6: Top key phrases extracted from summarized ”Unreliable” category.



Satire

1. bitch mouse solves maze researchers spent months building actually patriotic nutritionists recommend eating entire frozen pizza
2. drunk nutritionists recommend eating entire frozen pizza 18th century european history seminar via
3. trump administration named mad dog via
4. 5 nickelodeon characters every 90s kid wanted
5. level student joins shadow cabinet via clearing
6. zombie storm harvey threatens gulf coast
7. depot releases new bluetooth cordless hose
8. wikipedia page viewed 874 times today
9. level student joins shadow cabinet
10. mine ... man thinks playing guitar makes
11. stoned guys agree organised religion
12. lil pump reportedly scores 142
13. boogie board misses fourth wave
14. felicity smoak chip implant given
15. ref quietly asks penguins players
16. payment without collasal f ###-

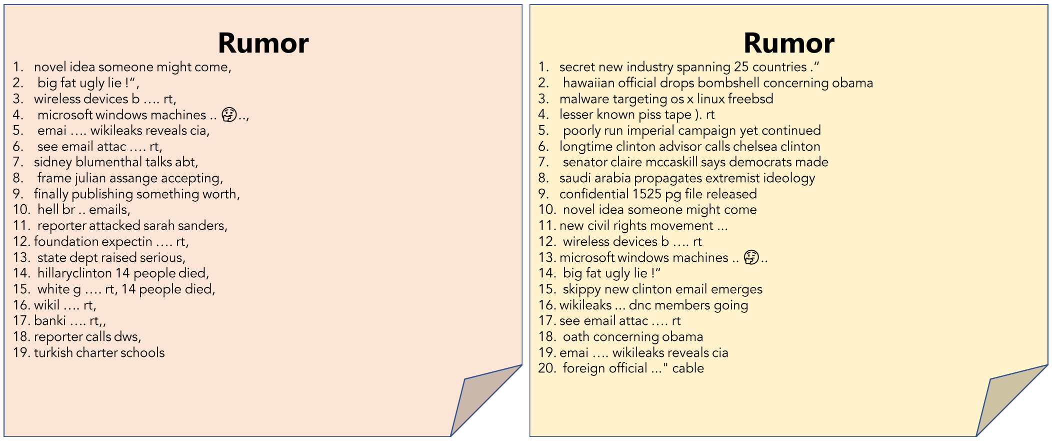
(a) With Summarization

Satire

1. level student joins shadow cabinet via clearing kawawa morgue employee cremated
2. mr motivator admits speed addiction 300 million year old petrified bell
3. level student joins shadow cabinet via clearing bus driver hailed
4. trump boys chasing wounded boar around white house feels silly
5. level student joins shadow cabinet via clearing diplomats believed
6. level student joins shadow cabinet via clearing spoke
7. 18th century european history seminar twisted perception
8. trump boys chasing wounded boar around white house
9. 18th century european history seminar cracking
10. bitch mouse solves maze researchers spent months building
11. somewhere else " says frontier airlines ceo via
12. actually patriotic nutritionists recommend eating entire frozen pizza
13. sea nutritionists recommend eating entire frozen pizza
14. 18th century european history seminar carved
15. level student joins shadow cabinet via clearing
16. 18th century european history seminar via
17. lago bellhop assigned rooftop sniper duty via
18. drunk nutritionists recommend eating entire frozen pizza
19. plumber nutritionists recommend eating entire frozen pizza
20. single homogeneous opinion ” college encourages lively exchange

(b) Without Summarization

Figure 5.7: Top key phrases extracted from summarized ”Satire” category.



Rumor

1. novel idea someone might come,
2. big fat ugly lie !",
3. wireless devices b …. rt,
4. microsoft windows machines .. 🤔..,
5. emai …. wikileaks reveals cia,
6. see email attac …. rt,
7. sidney blumenthal talks abt,
8. frame julian assange accepting,
9. finally publishing something worth,
10. hell br .. emails,
11. reporter attacked sarah sanders,
12. foundation expectin …. rt,
13. state dept raised serious,
14. hillaryclinton 14 people died,
15. white g …. rt, 14 people died,
16. wikil …. rt,
17. banki …. rt,,
18. reporter calls dws,
19. turkish charter schools

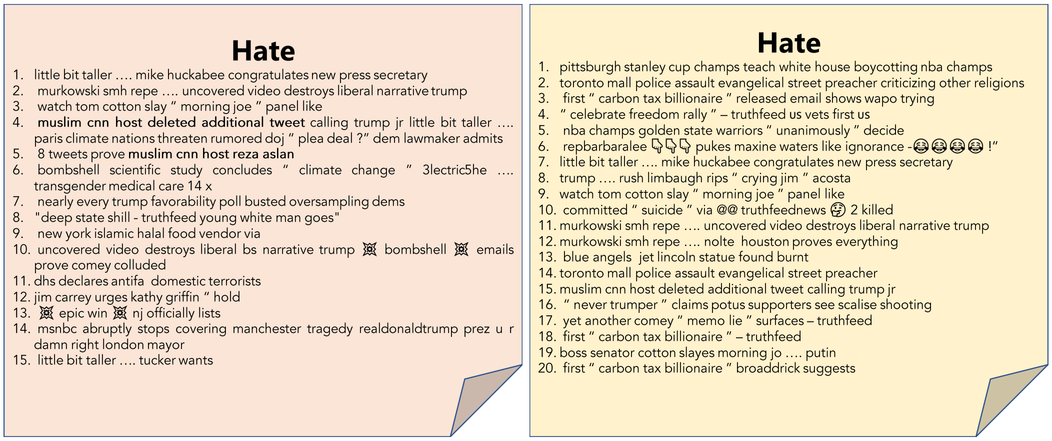
(a) With Summarization

Rumor

1. secret new industry spanning 25 countries ."
2. hawaiian official drops bombshell concerning obama
3. malware targeting os x linux freebsd
4. lesser known piss tape). rt
5. poorly run imperial campaign yet continued
6. longtime clinton advisor calls chelsea clinton
7. senator claire mccaskill says democrats made
8. saudi arabia propagates extremist ideology
9. confidential 1525 pg file released
10. novel idea someone might come
11. new civil rights movement ...
12. wireless devices b …. rt
13. microsoft windows machines .. 🤔..
14. big fat ugly lie !"
15. skippy new clinton email emerges
16. wikileaks ... dnc members going
17. see email attac …. rt
18. oath concerning obama
19. emai …. wikileaks reveals cia
20. foreign official ..." cable

(b) Without Summarization

Figure 5.8: Top key phrases extracted from summarized "Rumor" category.



Hate

1. little bit taller …. mike huckabee congratulates new press secretary
2. murkowski smh repe …. uncovered video destroys liberal narrative trump
3. watch tom cotton slay " morning joe " panel like
4. **muslim cnn host deleted additional tweet** calling trump jr little bit taller …. paris climate nations threaten rumored doj " plea deal ?" dem lawmaker admits
5. 8 tweets prove **muslim cnn host reza aslan**
6. bombshell scientific study concludes " climate change " 3lectric5he …. transgender medical care 14 x
7. nearly every trump favorability poll busted oversampling dems
8. "deep state shill - truthfeed young white man goes"
9. new york islamic halal food vendor via
10. uncovered video destroys liberal bs narrative trump 💥 bombshell 💥 emails prove comey colluded
11. dhs declares antifa domestic terrorists
12. jim carrey urges kathy griffin " hold
13. 💥 epic win 💥 nj officially lists
14. msnbc abruptly stops covering manchester tragedy realdonaldtrump prez u r damn right london mayor
15. little bit taller …. tucker wants

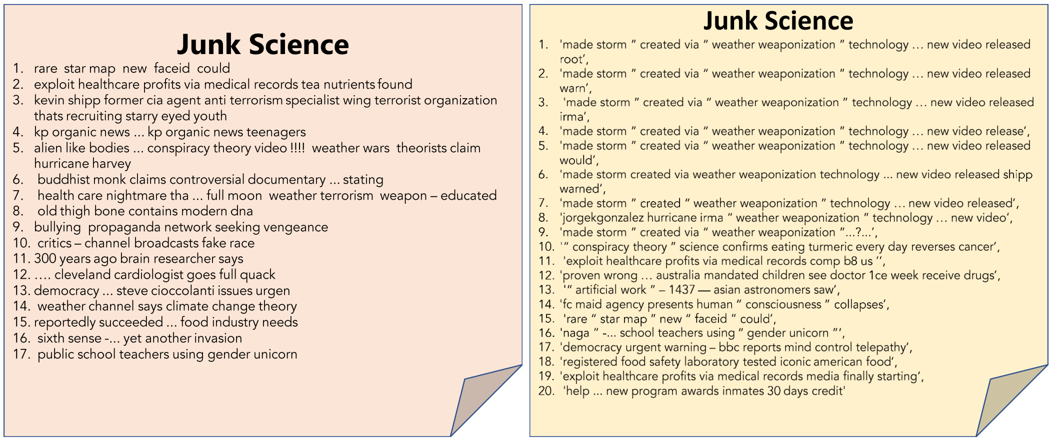
(a) With Summarization

Hate

1. pittsburgh stanley cup champs teach white house boycotting nba champs
2. toronto mall police assault evangelical street preacher criticizing other religions
3. first " carbon tax billionaire " released email shows wapo trying
4. " celebrate freedom rally " – truthfeed us vets first us
5. nba champs golden state warriors " unanimously " decide
6. repbarbaralee 👇👇👇 pukes maxine waters like ignorance -😂😂😂😂 !"
7. little bit taller …. mike huckabee congratulates new press secretary
8. trump …. rush limbaugh rips " crying jim " acosta
9. watch tom cotton slay " morning joe " panel like
10. committed " suicide " via @@ truthfeednews 🤔 2 killed
11. murkowski smh repe …. uncovered video destroys liberal narrative trump
12. murkowski smh repe …. nolte houston proves everything
13. blue angels jet lincoln statue found burnt
14. toronto mall police assault evangelical street preacher
15. muslim cnn host deleted additional tweet calling trump jr
16. " never trumper " claims potus supporters see scalise shooting
17. yet another comey " memo lie " surfaces – truthfeed
18. first " carbon tax billionaire " – truthfeed
19. boss senator cotton slayes morning jo …. putin
20. first " carbon tax billionaire " broaddrick suggests

(b) Without Summarization

Figure 5.9: Top key phrases extracted from summarized "Hate" category.



Junk Science

1. rare star map new faceid could
2. exploit healthcare profits via medical records tea nutrients found
3. kevin shipp former cia agent anti terrorism specialist wing terrorist organization thats recruiting starry eyed youth
4. kp organic news ... kp organic news teenagers
5. alien like bodies ... conspiracy theory video !!!! weather wars theorists claim hurricane harvey
6. buddhist monk claims controversial documentary ... stating
7. health care nightmare tha ... full moon weather terrorism weapon – educated
8. old thigh bone contains modern dna
9. bullying propaganda network seeking vengeance
10. critics – channel broadcasts fake race
11. 300 years ago brain researcher says
12. .... cleveland cardiologist goes full quack
13. democracy ... steve cioccolanti issues urgen
14. weather channel says climate change theory
15. reportedly succeeded ... food industry needs
16. sixth sense -... yet another invasion
17. public school teachers using gender unicorn

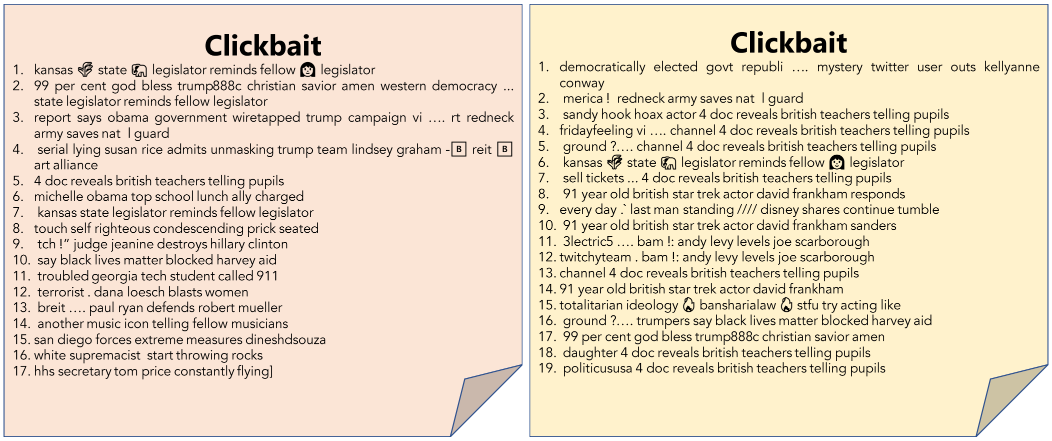
(a) With Summarization

Junk Science

1. 'made storm ″ created via ″ weather weaponization ″ technology … new video released root',
2. 'made storm ″ created via ″ weather weaponization ″ technology … new video released warn',
3. 'made storm ″ created via ″ weather weaponization ″ technology … new video released irma',
4. 'made storm ″ created via ″ weather weaponization ″ technology … new video release',
5. 'made storm ″ created via ″ weather weaponization ″ technology … new video released would',
6. 'made storm created via weather weaponization technology ... new video released shipp warned',
7. 'made storm ″ created ″ weather weaponization ″ technology … new video released',
8. 'jorgekgonzalez hurricane irma ″ weather weaponization ″ technology … new video',
9. 'made storm ″ created via ″ weather weaponization ″...?...',
10. '″ conspiracy theory ″ science confirms eating turmeric every day reverses cancer',
11. 'exploit healthcare profits via medical records comp b8 us '',
12. 'proven wrong … australia mandated children see doctor 1ce week receive drugs',
13. '″ artificial work ″ – 1437 — asian astronomers saw',
14. 'fc maid agency presents human ″ consciousness ″ collapses',
15. 'rare ″ star map ″ new ″ faceid ″ could',
16. 'naga ″ -... school teachers using ″ gender unicorn ″',
17. 'democracy urgent warning – bbc reports mind control telepathy',
18. 'registered food safety laboratory tested iconic american food',
19. 'exploit healthcare profits via medical records media finally starting',
20. 'help ... new program awards inmates 30 days credit'

(b) Without Summarization

Figure 5.10: Top key phrases extracted from summarized ”Junk Science” category.



Clickbait

1. kansas 💐 state 🐘 legislator reminds fellow 👩 legislator
2. 99 per cent god bless trump888c christian savior amen western democracy ... state legislator reminds fellow legislator
3. report says obama government wiretapped trump campaign vi …. rt redneck army saves nat l guard
4. serial lying susan rice admits unmasking trump team lindsey graham -🅱 reit 🅱 art alliance
5. 4 doc reveals british teachers telling pupils
6. michelle obama top school lunch ally charged
7. kansas state legislator reminds fellow legislator
8. touch self righteous condescending prick seated
9. tch !” judge jeanine destroys hillary clinton
10. say black lives matter blocked harvey aid
11. troubled georgia tech student called 911
12. terrorist . dana loesch blasts women
13. breit …. paul ryan defends robert mueller
14. another music icon telling fellow musicians
15. san diego forces extreme measures dineshdsouza
16. white supremacist start throwing rocks
17. hhs secretary tom price constantly flying]

(a) With Summarization

Clickbait

1. democratically elected govt republi …. mystery twitter user outs kellyanne conway
2. merica ! redneck army saves nat l guard
3. sandy hook hoax actor 4 doc reveals british teachers telling pupils
4. fridayfeeling vi …. channel 4 doc reveals british teachers telling pupils
5. ground ?…. channel 4 doc reveals british teachers telling pupils
6. kansas 💐 state 🐘 legislator reminds fellow 👩 legislator
7. sell tickets ... 4 doc reveals british teachers telling pupils
8. 91 year old british star trek actor david frankham responds
9. every day .` last man standing //// disney shares continue tumble
10. 91 year old british star trek actor david frankham sanders
11. 3lectric5 …. bam !: andy levy levels joe scarborough
12. twitchyteam . bam !: andy levy levels joe scarborough
13. channel 4 doc reveals british teachers telling pupils
14. 91 year old british star trek actor david frankham
15. totalitarian ideology 🔥 bansharialaw 🔥 stfu try acting like
16. ground ?…. trumpers say black lives matter blocked harvey aid
17. 99 per cent god bless trump888c christian savior amen
18. daughter 4 doc reveals british teachers telling pupils
19. politicususa 4 doc reveals british teachers telling pupils

(b) Without Summarization

Figure 5.11: Top key phrases extracted from summarized ”Clickbait” category.

Part III

Ensemble Techniques for Multi-Aspect Detection of Misinformation

6

HiJoD A Tensor-based Ensemble Technique for Misinformation Detection

Distinguishing between misinformation and real information is one of the most challenging problems in today's interconnected world. The vast majority of the state-of-the-art in detecting misinformation is fully supervised, requiring a large number of high-quality human annotations. However, the availability of such annotations cannot be taken for granted, since it is very costly, time-consuming, and challenging to do so in a way that keeps up with the proliferation of misinformation. In this work, we are interested in exploring scenarios where the number of annotations is limited. In such scenarios, we investigate how to tap into a diverse number of resources that characterize a news article, henceforth referred to as "aspects" can compensate for the lack of labels. In particular, our contributions in this work are twofold: 1) We propose the use of three different aspects: article content, context of social sharing behaviors, and host website/domain features, and 2) We introduce a principled tensor based embedding framework that combines all those aspects effectively. We propose HiJoD a 2-level decomposition pipeline which not only outperforms state-of-the-art methods with F1-scores of 74% and 81% on Twitter and Politifact datasets respectively, but also is an order of magnitude faster than similar ensemble approaches.

6.1 Introduction

In recent years, we have experienced the proliferation of websites and outlets that publish and perpetuate misinformation. With the aid of social media platforms, such misinformation propagates wildly and reaches a large number of the population, and can, in fact, have real-world consequences. Thus, understanding and flagging misinformation on the web is an extremely important and timely problem, which is here to stay.

There have been significant advances in detecting misinformation from the article content, which can be largely divided into knowledge-based fact-checking and style-based approaches; [91, 137] survey the landscape. Regardless of the particular approach followed, the vast majority of

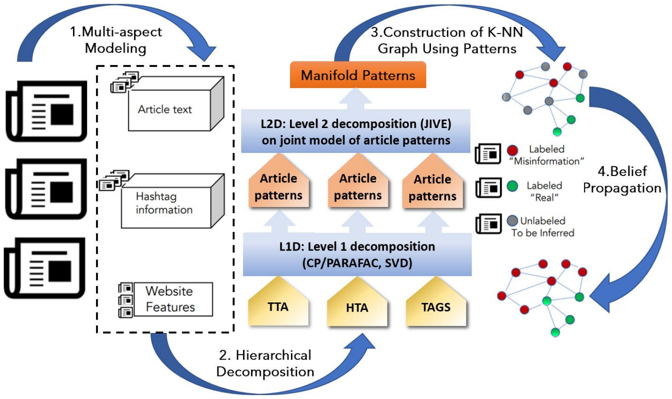


Figure 6.1: Overview: We propose using three aspects1: a) content, b) social context (in the form of hashtags), and c) features of the website serving the content. We, further, propose HiJoD hierarchical approach for finding latent patterns derived from those aspects, generate a graphical representation of all articles in the embedding space, and conduct semi-supervised label inference of unknown articles.

the state-of-the-art is fully supervised, requiring a large number of high-quality human annotations in order to learn the association between content and whether an article is misinformative. For instance, in [25], a decision-tree based algorithm is used to assess the credibility of a tweet, based on Twitter features. In another work, Rubin et al. [131] leverage linguistic features and a SVM-based classifier to find misleading information. Similarly, Horne and Adali [70] apply SVM classification on content-based features. There are several other works [58, 73, 75] for assessing credibility of news articles, all of which employ propagation models in a supervised manner. Collecting human annotation for misinformation detection is a complicated and time consuming task, since it is challenging and costly to identify human experts who can label news articles devoid of their own subjective views and biases, and possess all required pieces of information to be a suitable "oracle." However, there exist crowd-sourced schemes such as the browser extension "BS Detector" [1] which

[1] http://bsdetector.tech/

provide coarse labels by allowing users to flag certain articles as different types of misinformation, and subsequently flagging the entire source/domain as the majority label. Thus, we are interested in investigating methods that can compensate for the lack of large amounts of labels with leveraging different signals or aspects that pertain to an article. A motivating consideration is that we as humans empirically consider different aspects of a particular article in order to distinguish between misinformation and real information. For example, when we review a news article on a web site, and we have no prior knowledge about the legitimacy of its information, we not only take a close look at the content of the article but also we may consider how the web page looks (e.g. does it look "professional"? Does it have many ads and pop-up windows that make it look untrustworthy?). We might conclude that untrustworthy resources tend to have more "messy" web sites in that they are full of ads, irrelevant images, pop-up windows, and scripts. Most prior work in the misinformation detection space considers only one or two aspects of information, namely content, headline or linguistic features. In this chapter, we aim to fill that gap by proposing a comprehensive method that emulates this multi-aspect human approach for finding latent patterns corresponding to different classes, while at the same time leverages scarce supervision in order to turn those insights into actionable classifiers. In particular, our proposed method combines multiple aspects of article, including (a) article content (b) social sharing context and (c) source webpage context each of which is modeled as a tensor/matrix. The rationale behind using tensor based model rather than state-of-the-art approaches like deep learning methods is that, such approaches are mostly supervised methods and require considerable amount of labeled data for training. On the contrary, we can leverage tensor based approaches to find meaningful patterns using less labels. Later on, we will compare the performance of tensor-based modeling against deep learning methods when there is scarcity of labels. We summarize the contributions as follows:

- **Leveraging different aspects of misinformation**: In this work, we not only consider content-based information but also we propose to leverage multiple aspects for discriminating misinformative articles. In fact, we propose to create multiple models, each of which describes a distinct aspect of articles.
- **A novel hierarchical tensor based ensemble model**: We leverage a hierarchical tensor based ensemble method i.e., HiJoD to find manifold patterns that comprise multi-aspect information of the data.
- **Evaluation on real data**: We extensively evaluate HiJoD on two real world datasets i.e., Twitter and Politifact. HiJoD not only outperforms state-of-the-art alternatives with F1 score of 74% and 81% on above datasets respectively, but also is significantly faster than similar ensemble methods. We make our implementation publicly available[2] to promote the reproducibility.

6.2 Problem formulation

Considering the following formulation of misinformation detection problem:

Given *N* articles with associated 1) article text, social sharing context (hashtags that are used when sharing the article) 2) HTML source of the article's publisher webpage, and, 3) binary (misinformative/real) labels for p% of articles, **Classify** the remaining articles into the two classes.

[2] https://github.com/Saraabdali/HiJoD-ECMLPKDD

At a high level, we aim to demonstrate the predictive power of incorporating multiple aspects of article content and context on the misinformation detection task, and consider doing so in a low-label setting due to practical challenges in data labeling for this task. Our proposed method especially focuses on (a) multi-aspect data modeling and representation choices for downstream tasks, (b) appropriate triage across multiple aspects, and (c) utility of a semi-supervised approach which is ideal in sparse label settings. In the next section, we detail our intuition and choices for each of these components.

We aim to develop a manifold approach which can discriminate misinformative and real news articles by leveraging content-based information in addition to other article aspects i.e. social context and publisher webpage information. To this end, we propose using tensors and matrices to analyze these aspects jointly. We develop a three-stage approach: (a) multi-aspect article representation: we first introduce three feature context representations (models) which describe articles from different points of view (aspects), (b) Manifold patterns finding using a hierarchical approach: In proposed HiJoD, we first decompose each model separately to find the latent patterns of the articles with respect to the corresponding aspect, then we use a strategy to find shared components of the individual patterns (c) semi-supervised article inference: finally, we focus on the inference task by construing a K-NN graph over the resulted manifold patterns and propagating a limited set of labels.

6.3 Proposed Methodology

6.3.1 Multi-aspect article representation

We first model articles with respect to different aspects. We suggest following tensors/matrix to model content, social context and source aspects respectively:

- **(Term×Term×Article) Tensor ($\mathcal{X}_{\mathbf{TTA}}$)**: The most straightforward way that comes into mind for differentiating between a fake news and a real one is to analyze the content of the articles. Different classes of news articles, i.e., fake and real classes tend to have some common words that co-occur within the text. Thus, we use a tensor proposed by [71] to model co-occurrence of these common words. This model not only represents content-based information but also considers the relations between the words and is stronger than widely used bag of words and tf/idf models. To this end, we first create a dictionary of all articles words and then slide a window across the text of each article and capture the co-occurred words. As a result, we will have a co-occurrence matrix for each article. By stacking all these matrices, we create a three mode tensor where the first two dimensions correspond to the indices of the words in the dictionary and the third mode indicates the article's ID. We may assign binary values or frequency of co-occurrence to entries of the tensors. However, as shown in [52] binary tensor is able to capture more nuance patterns. So, in this work we also use binary values.

- **(Hashtag×Term×Article) Tensor ($\mathcal{X}_{\mathbf{HTA}}$)**: Hashtags often show some trending across social media. Since, hashtags assigned to an article usually, convey social context information which is related to content of news article, we propose to construct a hashtag-content tensor to model such patterns. In this tensor, we want to capture co-occurrence of words within the articles and the hashtags assigned to them. For example, an article tagged with a hashtag #USElection2016 probably consists of terms like “Donald Trump” or “Hillary Clinton”. These kind of co-occurrences are meaningful and convey some patterns which may be shared between different categories of articles. The first two modes of this tensor correspond to hashtag and word indices respectively and the third mode is article mode.

- **(Article×HTML features) Matrix ($\mathbf{X_{TAGS}}$)**: Another source of information is the trustworthiness of serving webpage. In contrast to reliable web resources which usually have a standard form, misleading web pages may often be messy and full of different advertisement, pop-ups and multimedia features such as images and videos. Therefore, we suggest to create another model to capture the look and the feel of the web page serving the content of news article. We can approximate look of a web page by counting HTML features and tags and then represent it by a (article, HTML feature) matrix. The rows and columns of this matrix indicate the article and hashtag IDs respectively. We fill out the entries by frequency of HTML tags in the web source of each article domain. Fig. 6.2 demonstrates aforementioned aspects.

6.3.2 Hierarchical decomposition

Now, the goal is to find manifold patterns with respect to all introduced aspects. To this end, we look for shared components of the latent patterns.

Level-1 decomposition: finding article patterns with respect to each aspect

As mentioned above, first, we find the latent patterns of the articles with respect to each aspect. To do so, we decompose first two tensor models using Canonical Polyadic or CP/PARAFAC decomposition we discussed in Algorithm 2. We define factor matrices as $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2 \dots \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2 \dots \mathbf{b}_R]$, and $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2 \dots \mathbf{c}_R]$ where for $\mathcal{X}_{\texttt{TTA}}$ and $\mathcal{X}_{\texttt{HTA}}$, $\mathbf{C}$ corresponds to the article mode and comprises the latent patterns of the articles with respect to content and social context respectively. For the tag matrix, to keep the consistency of the model we suggest to use SVD decomposition because as we discussed in the background section, CP/PARAFAC is one extension of SVD for higher mode arrays. In this case, $\mathbf{U}$ comprises the latent patterns of articles with respect to overall look of the serving webpage.

Level-2 decomposition: finding manifold patterns

Now, we want to put together article mode factor matrices resulted from individual decompositions to find manifold patterns with respect to all aspects. Let's $\mathbf{C_{TTA}} \in \mathbb{R}^{N \times r_1}$ and $\mathbf{C_{HTA}} \in \mathbb{R}^{N \times r_2}$ be the third factor matrices resulted from CP decomposition of $\mathcal{X}_{\texttt{TTA}}$ and $\mathcal{X}_{\texttt{HTA}}$ respectively and let's $\mathbf{C_{TAGS}} \in \mathbb{R}^{N \times r_3}$ be $\mathbf{U}$ matrix resulted from rank r_3 SVD of $\mathbf{X}_{\texttt{TAGS}}$ where N is the number of articles. Since all these three matrices comprise latent patterns of the articles, we aim at finding patterns which are shared between all. To do so, we concatenate the three article embedding matrices ($\mathbf{C}$) and decompose the joint matrix of size $(r_1 + r_2 + r_3) \times N$ to find shared components. Like level-1 decomposition, we can simply take the SVD of joint matrix. SVD seeks an accurate representation in least-square setting but to find a meaningful representation we need to consider additional information i.e., the relations between the components. Independent components analysis (ICA) tries to find projections that are statistically independent as follows:

$$[\mathbf{C}_{\mathbf{TTA}}; \mathbf{C}_{\mathbf{HTA}}; \mathbf{C}_{\mathbf{TAGS}}] = \mathbf{AS} \tag{6.1}$$

Where **A** is the corresponding shared factor of size $(r_1 + r_2 + r_3) \times N$ and **S** is the mixed signal of size $(r_1 + r_2 + r_3) \times R$. To achieve the statistical independence, it looks for projections that leads to projected data to be as far from Gaussian distribution as possible. The problem is that, SVD (PCA) and ICA may result in identical subspaces. In fact, this happens when the direction of greatest variation and the independent components span the same subspace [162]

In order to consider relations between components and find a meaningful representation, we can consider shared and unshared components for **C** matrices such that unshared components are orthogonal to shared ones. To this end, we propose to use the Joint and Individual Variation Explained (JIVE) for level-2 decomposition [103]. More precisely, let's consider $\mathbf{A}_i$ and $\mathbf{J}_i$ to be the matrices representing the individual structure and submatrix of the joint pattern for $i \in \{\texttt{HTA},\texttt{TTA},\texttt{TAGS}\}$ respectively such that they satisfy the orthogonality constraint. Using JIVE method, we decompose each article mode factor matrix **C** as follows:

$$\mathbf{C}_i = \mathbf{J}_i + \mathbf{A}_i + \epsilon_i \tag{6.2}$$

To find **A** and **J** matrices, we use the approach presented in [103]. In other words, we fix **A** and find **J** that minimizes the following residual matrix:

$$||\mathbf{R}||^2 = \left\|\epsilon_{TTA}; \epsilon_{HTA}; \epsilon_{Tags}\right\|^2 \tag{6.3}$$

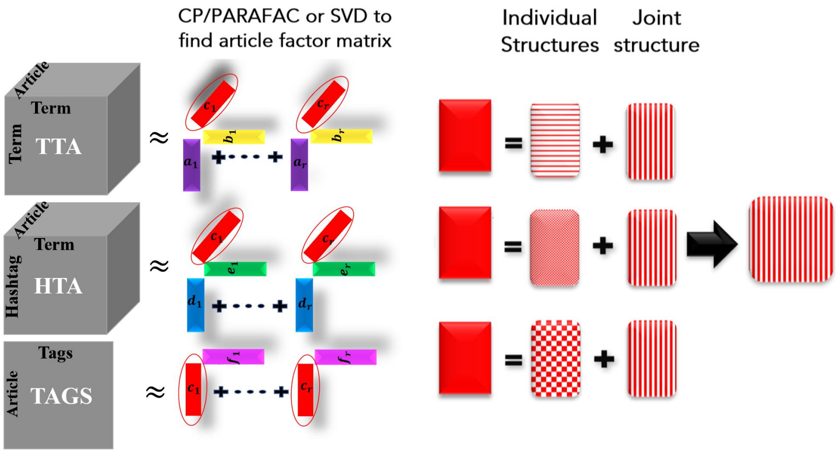


Figure 6.2: HiJoD finds manifold patterns of the articles that can be used for classification.

The joint structure **J** which minimizes $||\mathbf{R}||^2$ is equal to the rank r SVD of joint matrix when we remove the individual structure and in the same way individual structure for each **C** matrix is the rank r_i SVD of **C** matrix when we remove the joint structure [103]. Fig. 6.2 and Algorithm. 5 demonstrate the details.

6.3.3 Semi-supervised article inference

Previous step, provides us with a $N \times r$ matrix which comprises manifold patterns of N articles. In this step, we leverage this matrix to address the semi-supervised problem of classifying misinformation. Row i of this matrix represents article i in r dimensional space, we suggest to construct a K-NN graph using this matrix such that each node represents an article and edges are Euclidean distances between articles (rows of manifold patterns matrix) to model the similarity of articles.

Algorithm 5 HiJoD Hierarchical decomposition

```
Input :$\mathcal{X}_{\texttt{TTA}}$,$\mathcal{X}_{\texttt{HTA}}$,$\mathbf{X}_{TAGS}$
Output : $\mathbf{J_{joint}}$
//Level-1 Decomposition
$\mathbf{C}_{\texttt{TTA}}$ = CP-ALS($\mathcal{X}_{\texttt{TTA}}, \mathbf{r_1}$)
$\mathbf{C}_{\texttt{HTA}}$ = CP-ALS($\mathcal{X}_{\texttt{HTA}}, \mathbf{r_2}$)
$\mathbf{C}_{\texttt{TAGS}}$ = SVD($\mathbf{X}_{\texttt{TAGS}}, r_3$)
$\mathbf{C^{Joint}} = [\mathbf{C}_{\texttt{TTA}}; \mathbf{C}_{HTA}; \mathbf{C}_{\texttt{TAGS}}]$
//Level-2 Decomposition
while $\|\mathbf{R}\|^2 < \epsilon$ do
    $\mathbf{J} = \mathbf{U\Sigma V}^T$ $[\mathbf{J}_{\texttt{TTA}}; \mathbf{J}_{\texttt{HTA}}; \mathbf{J}_{\texttt{TAGS}}]$ =SVD ($\mathbf{C}_{Joint}, r_{joint}$) //Calculate $r_{joint}$ using Algorithm 2
    for $i \in$ { TTA,HTA,TAGS } do
        $\mathbf{A_i} = \mathbf{C_i} - \mathbf{J_i}$
        $\mathbf{A_i}$ =SVD ($\mathbf{A_i} * (\mathbf{I} - \mathbf{VV}^T), r_i$) //To satisfy the orthogonality constraint
    end
    $\mathbf{C_i}^{new} = \mathbf{C_i}^{joint} - \mathbf{A_i}$
    $\epsilon_i = |\mathbf{C_i}^{joint} - \mathbf{C_i}^{new}|$
    $\mathbf{C}^{joint} = \mathbf{C}^{new}$
end
```

Algorithm 6 Calculating the rank for joint and individual matrices

```
Input: $\alpha \in (0, 1), n\_perm$ Output: r
Let's $\lambda_j$ be the $j$'th singular value of $\mathbf{X}, i = 1, \dots, rank(\mathbf{X})$.
while $n \leq n\_perm$ do
    Permute the columns within each $X_i$, and calculate the singular values of the resulting $\mathbf{C_{joint}}$
end
$\lambda_i^{perm} = 100(1 - \alpha)$ percentile of $j$'th singular values
Choose largest $r$ such that $\forall j \leq r, \lambda_j \geq \lambda_j^{perm}$
```

We utilize the Fast Belief Propagation (FaBP) algorithm as is described in 2 to propagate labels of known articles (fake or real) throughout K-NN graph.

6.4 Experimental evaluation

In this section, we discuss the datasets, baselines and experimental results.

6.4.1 Dataset description

Twitter dataset To evaluate HiJoD, we created a new dataset by crawling Twitter posts contained links to articles and shared between June and August 2017. This dataset comprises 174k articles from more than 652 domains. For labeling the articles, we used BS-Detector, which is a crowd-sourced toolbox in form of a browser extension, as ground truth. BS-Detector categorizes domains into different categories such as bias, clickbait, conspiracy, fake, hate, junk science, rumor, satire, and unreliable. We consider above categories as "misinformative" class. A key caveat behind BS-Detector, albeit being the most scalable and publicly accessible means of labeling articles at-large, is that labels actually pertains to the domain rather than the article itself. At the face of it, this sounds like the labels obtained are for an entirely different task, however, Helmstetter et al. [66] show that training for this "weakly labeled" task (using labels for the domains), and subsequently testing on labels pertaining to the articles, yields minimal loss in accuracy and labels still hold valuable information. Thus, we choose BS-Detector for ground truth. However, in order to make our experiments as fair as possible, in light of the above fact regarding the ground truth, we do as follows:

- We restrict the number of articles per domain we sample into our pool of articles. So, we experimented with randomly selecting a single article per domain and iterating over 100 such sets. Since we have 652 different domains in Twitter dataset, in each iteration we chose 652 non-overlapping articles so, totally we examine different approaches for 65.2K different articles.

- In order to observe the effect of using more articles per domain, we repeated the experiments for different number of articles per domain. We observed that the embedding that uses HTML tags receives a disproportionate boost in its performance. We attribute this phenomenon to the fact that all instances coming from the same domain have exactly the same HTML features, thus classifying correctly one of them implies correct classification for the rest, which is proportional to the number of articles per domain.

- We balance the dataset so that we have 50% fake and 50% real articles at any given run per method. We do so to have a fair evaluation setting and prevent the situation in which there is a class bias. To show the insensitivity of HiJoD to class imbalance, we also experiment on an imbalanced dataset.

Politifact dataset: For second dataset, we leverage FakeNewsNet dataset that the authors of [79] used for their experiments[3] [80, 137]. This dataset consists of 1056 news articles from the Politifact fact checking website, 60% being real and 40% being fake. Using this imbalanced dataset, we can experiment how working on an imbalanced dataset may affect the proposed approach. Since not all of the signals we used from Twitter dataset exist in Politifact dataset, For this dataset we created the following embeddings:

- $\mathcal{X}_{\texttt{TTA}}$ tensor: To keep the consistency, we created $\mathcal{X}_{\texttt{TTA}}$ from articles text.
- User-News Interaction Embedding: We create a matrix which represent the users who tweets a specific news article, as proposed in [79].
- Publisher-News Interaction Embedding: We create another matrix to show which publisher published a specific news article, as proposed in [79].

Using the aforementioned signals, we can also test the efficacy of HiJoD when we leverage aspects other than those we proposed in this work.

[3] https://github.com/KaiDMML/FakeNewsNet

6.4.2 Baselines for comparison

As mentioned earlier, the two major contributions of HiJoD are: 1) the introduction of different aspects of an article and how they influence our ability to identify misinformation more accurately, and 2) how we leverage different aspects to find manifold patterns which belongs to different classes of articles. Thus, we conduct experiments with two categories of baseline to test each contribution separately:

Content-based approaches to test the effect of additional aspects. We compare with state-of-the-art content-based approaches to measure the effect of introducing additional aspects (hashtags and HTML features) into the mix, and whether the classification performance improves. We compare against:

- **TTA/BP** In [52], Bastidas et al. effectively use the co-occurrence tensor in a semi-supervised setting. They demonstrate how this tensor embedding outperforms other purely content-based state-of-the-art methods such as SVM on content-based features and Logistic regression on linguistic features [62, 70]. Therefore, we select this method as the first baseline, henceforth referenced as "$\mathcal{X}_{\texttt{TTA}}$". The differences between our results and the results reported by Bastidas et al [52] is due to using different datasets. However, since we used the publicly available code by Bastidas et al. [4] [52], if we were to use the same data in [52] the results would be exactly the same.
- tf_idf/SVM is the well-known term frequency–inverse document frequency method widely

[4] https://github.com/Saraabdali/Fake-News-Detection-ASONAM-2018

used in text mining and information retrieval and illustrates how important a word is to a document. We create a tf-idf model out of articles text and apply SVM classifier on the resulted model.

- **Doc2Vec/SVM** is an NLP toolbox proposed by Le et al. [93] from Google. This model is a shallow, two-layer neural network that is trained to reconstruct linguistic contexts of document. This algorithm is an extension to word2vec which can generate vectors for words. Since the SVM classifier is commonly used on this model, we also leverage SVM for document classification.[5]
- **fastText** is an NLP library by Facebook Research that can be used to learn word representations to efficiently classify document. It has been shown that fastText results are on par with deep learning models in terms of accuracy but an order of magnitude faster in terms of performance.[6]
- **GloVe/LSTM** GloVe is an algorithm for obtaining vector representations of the words. Using an aggregated global word-word co-occurrence, this method results in a linear substructures of the word vector space. We use the method proposed in [92, 101] and we create a dictionary of unique words and leverage Glove to map indices of words into a pre-trained word embedding. Finally, as suggested, we use LSTM to classify articles.[7]

Ensemble approaches to test the efficacy of our fused method. Another way to jointly derive the article patterns, specially when the embeddings are tensors and matrices, is to couple matrices and tensors on shared mode(s) i.e., article mode in this work. In this technique which called coupled

[5] https://github.com/seyedsaeidmasoumzadeh/Binary-Text-Classification-Doc2vec-SVM
[6] https://github.com/facebookresearch/fastText
[7] https://github.com/prakashpandey9/Text-Classification-Pytorch

Matrix and Tensor Factorization (CMTF), the goal is to find optimized factor matrices by considering all different optimization problems we have for individual embeddings. Using our proposed embeddings, we the optimization problem is:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{E},\mathbf{F}} \|\mathcal{X}_{\texttt{TTA}} - [\mathbf{A},\mathbf{B},\mathbf{C}]\|^2 + \|\mathcal{X}_{\texttt{HTA}} - [\mathbf{D},\mathbf{E},\mathbf{C}]\|^{\mathbf{2}} + \|\mathbf{X}_{\texttt{TAGS}} - \mathbf{C}\mathbf{F}^{\mathbf{T}}\|^{\mathbf{2}} \tag{6.4}$$

where $[\mathbf{A},\mathbf{B},\mathbf{C}]$ denotes $\Sigma_{r=1}^{R}\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$, and $\mathbf{C}$ is the shared article mode as shown in Fig.6.3. In order to solve the optimization problem above, we use the approach introduced in [41], which proposes all-at-once-optimization by computing the gradient of every variable of the problem, stacking the gradients into a long vector, and applying gradient descent. There is an advanced version of coupling called ACMTF that uses weights for rank-one components to consider both shared and unshared ones [4]. We applied ACMTF as well and it led to similar results, however, slower than standard CMTF. Thus, we just report the result of CMTF.

Moreover, as mentioned earlier, to derive the manifold pattern which is shared between the aspects, we can leverage different mathematical approaches such as:

- SVD (Singular Value Decomposition)
- JICA (Joint Independent Component Analysis)

on $\mathbf{C}_{\mathbf{joint}}$. To measure the performance of JIVE method for finding manifold patterns, we will also examine the above approaches for level-2 decomposition.

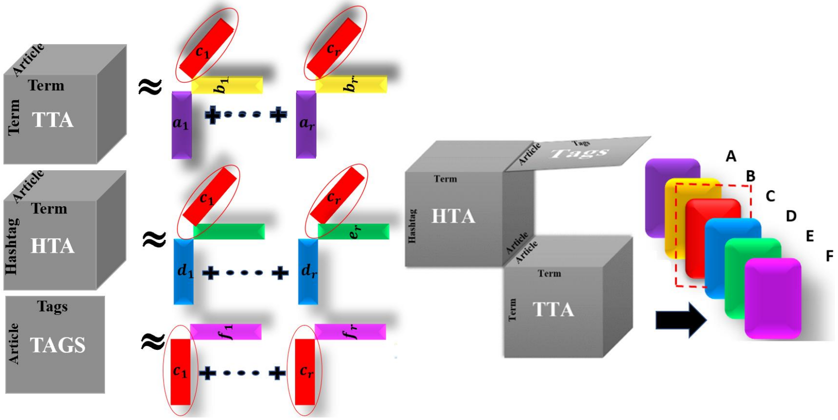


Figure 6.3: Couple Tensor Matrix Factorization for finding shared patterns of the articles.

6.4.3 Comparing with baselines

Implementation.

We implemented HiJoD and all other approaches in MATLAB using Tensor Toolbox version 2.6. Moreover, to implement JICA approach, we used FastICA[8] , a fast implementation of ICA. For the baseline approach CMTF we used the Toolbox[9] in [40, 41]. For the belief propagation step, we used implementation introduced in [89]. Based on the experiments reported in [52], we employed a sliding window of size 5 for capturing the co-occurring words. Using AutoTen [119], which finds the best rank for tensors, we found out the best rank for $\mathcal{X}_{\mathtt{TTA}}$ and $\mathcal{X}_{\mathtt{HTA}}$ is 10 and 40 respectively. For the $\mathbf{X}_{\mathtt{TAGS}}$ embedding, we took the full SVD to capture the significant singular values and set the $\mathbf{X}_{\mathtt{TAGS}}$ SVD rank to 20. Since for CMTF approach we have to choose the same rank, as a heuristic,

[8] https://github.com/aludnam/MATLAB/tree/master/FastICA_25

[9] http://www.models.life.ku.dk/joda/CMTF_Toolbox

we used the range of ranks for individual embeddings (10 for $\mathcal{X}_{\texttt{HTA}}$ and 40 for $\mathcal{X}_{\texttt{TTA}}$) and again searched for the ensemble rank in this range. Based on our experiments, rank 30 leads to the best results in terms of F1-score. Moreover, grid search over 1-30 nearest neighbors yielded choice of 5 neighbors for CMTF and 15 for HiJoD. For the rank of joint model as well as individual and joint structures i.e., $\mathbf{A}_i$ and $\mathbf{J}_i$, we used the strategy proposed in [103] and for reproducibility purposes is demonstrated in Algorithm 6. The intuition is to find the rank of joint structure i by comparing the singular values of the original matrix with the singular values of n_{perm} randomly permuted matrices. If the j^{th} singular value in the original matrix is $\geq 100(1-\alpha)$ percentile of the j^{th} singular value of n_{perm} matrices, we keep it as a significant one. Number of these significant singular values shows the rank. n_{perm} and α are usually set to 100 and 0.05 respectively. For timing experiment, we used the following configuration:

- CPU: Intel(R) Core(TM) i5-8600K CPU @3.60GHz
- OS: CentOS Linux 7 (Core)
- RAM: 40GB

Testing the effect of different aspects.

This experiment refers to the first category of baselines i.e., state-of the-art content-based approaches introduced earlier. Table 6.1 demonstrates the comparison; label% shows the percentage of known data used for propagation/training of the models. As reported, HiJoD outperforms all content-based approaches significantly. For example, using only 10% of known labels the F1 score of HiJoD is around 12% and 13% percent more than Doc2Vec/SVM and fastText respectively. In case of GloVe/LSTM, due to small size of training set i.e., 10 and 20 percent, the model overfits

%Labels	tf_idf/SVM	Doc2Vec/SVM	fastText	GloVe/LSTM	TTA/BP	HiJoD
10	0.500±0.032	0.571±0.092	0.562±0.031	-	0.582±0.018	**0.693±0.009**
20	0.461±0.013	0.558±0.067	0.573±0.027	-	0.598±0.018	**0.717±0.010**
30	0.464±0.015	0.548± 0.048	0.586±0.024	0.502±0.087	0.609±0.019	**0.732±0.011**
40	0.475±0.023	0.547±0.034	0.592± 0.028	0.503±0.060	0.614±0.022	**0.740±0.010**

Table 6.1: F1 score of HiJoD outperforms all content-based methods on Twitter dataset.

	Twitter			Politifact	
%Labels	CMTF	CMTF++	HiJoD	CMTF	HiJoD
10	0.657±0.009	0.657±0.010	**0.693±0.009**	0.733±0.007	**0.766±0.007**
20	0.681±0.010	0.681±0.009	**0.717± 0.010**	0.752±0.012	**0.791±0.007**
30	0.691±0.010	0.692±0.009	**0.732±0.011**	0.774±0.006	**0.802±0.007**
40	0.699±0.010	0.699±0.009	**0.740±0.010**	0.776±0.004	**0.810±0.008**

Table 6.2: HiJoD outperforms coupling approaches in terms of F1 score in both datasets.

easily which shows the strength of HiJoD against deep models when there is scarcity of labeled data for training. Moreover, our ensemble model that leverages $\mathcal{X}_{\text{TTA}}$ as one of its embeddings beats the individual decomposition of $\mathcal{X}_{\text{TTA}}$ which illustrates the effectiveness of adding other aspects of the data for modeling the news articles.

Testing the efficacy of fused method in HiJoD.

For the second category of baselines, we compare HiJoD against the recent work for joint decomposition of tensors/matrices i.e., CMTF. As discussed earlier, we couple $\mathcal{X}_{\text{TTA}}$, $\mathcal{X}_{\text{HTA}}$ and $\mathbf{X}_{\text{TAGS}}$ on shared article mode. Since the "term" mode is also shared between $\mathcal{X}_{\text{TTA}}$ and $\mathcal{X}_{\text{HTA}}$, we also examine the CMTF by coupling on both article and term modes, henceforth referenced as CMTF++ in the experimental results. For the Politifact dataset, there is only one shared mode i.e., article mode. therefore, we only compare against CMTF approach. The experimental results for theses approaches are shown in Table 6.2. As illustrated, HiJoD leads to higher F1 score which confirms the effectiveness of HiJoD for jointly classification of articles. One major drawback of CMTF approach is that,

	L1D		**L2D+Rank finding**		**Total time (Secs.)**	
L1D Rank	**CMTF**	**CP/SVD**	**CMTF**	**JIVE**	**CMTF**	**HiJoD**
5	352.96	7.63	-	13.69	352.96	21.32
10	1086.70	16.35	-	49.69	1086.70	66.04
20	11283.51	44.25	-	133.70	11283.51	177.95
30	13326.80	84.76	-	278.43	13326.80	363.19

Table 6.3: Comparing execution time (Secs.) of CMTF against HiJoD on Twitter dataset shows that HiJoD is an order of magnitude faster than CMTF approach.

we have to use a unique decomposition rank for the joint model which may not fit all embeddings and may lead to losing some informative components or adding useless noisy components due to inappropriate rank of decomposition. Another drawback of this technique is that, as we add more embeddings, the optimization problem becomes more and more complicated which may cause the problem become unsolvable and infeasible in terms of time and resources. The time efficiency of HiJoD against CMTF approach is reported in Table. 6.3; we refer to level-1 and level-2 decompositions as L1D and L2D respectively. As shown, for all ranks, HiJoD is an order of magnitude faster than CMTF due to the simplicity of optimization problem in comparison to equation 6.4 which means HiJoD is more applicable for real world problems.

Testing the efficacy of using JIVE for level-2 decomposition.

In this experiment we want to test the efficacy of JIVE against other approaches for deriving the joint structure of **C**. The evaluation results for this experiment are reported in Table. 6.4. As shown, SVD and JICA resulted in same classification performance which as explained earlier indicates that the directions of greatest variation and the independent components span the same subspace [162]. However, the F1 score of HiJoD is higher than two other methods on both datasets which practically justifies that considering orthogonal shared and unshared parts and minimizing the residual can

%Labels	Twitter			Politifact		
	JICA	SVD	JIVE	JICA	SVD	JIVE
10	0.684±0.009	0.685±0.017	**0.693±0.009**	0.749±0.006	0.756±0.009	**0.766±0.007**
20	0.710±0.009	0.712±0.014	**0.717±0.010**	0.775±0.006	0.783±0.009	**0.791±0.007**
30	0.724±0.009	0.724±0.018	**0.732±0.011**	0.786±0.006	0.796±0.012	**0.802±0.007**
40	0.734±0.009	0.735±0.012	**0.740±0.010**	0.797±0.006	0.807±0.008	**0.810±0.008**

Table 6.4: Comparing the efficacy of different approaches for level-2 decomposition illustrates that using JIVE approach for HiJoD ouperforms other methods.

improve the performance of naive SVD.

HiJoD vs. single aspect modeling.

Finally, we want to investigate how adding other aspects of the articles affect the classification performance. To this end, we decompose our proposed embeddings i.e., $\mathcal{X}_{\texttt{TTA}}$, $\mathcal{X}_{\texttt{HTA}}$ and $\mathbf{X}_{\texttt{TAGS}}$ extracted from Twitter dataset individually and leverage the **C** matrices for classification. The result of this experiment is demonstrated in Fig. 6.4. As mentioned before, in contrast to CMTF, in HiJoD we can use different ranks for different aspects as we did so previously. However, in order to conduct a fair comparison between individual embeddings and HiJoD, we merge the embeddings of the same rank. It is worth mentioning that, performance of HiJoD is higher than what is shown in Fig. 6.4 due to concatenation of **C** matrices of the best rank. So, this result is just for showing the effect of merging different aspects by fixing other parameters i.e., rank and k. As shown, the HiJoD even when we join embeddings of the same rank and do not use the best of which outperforms individual decompositions.

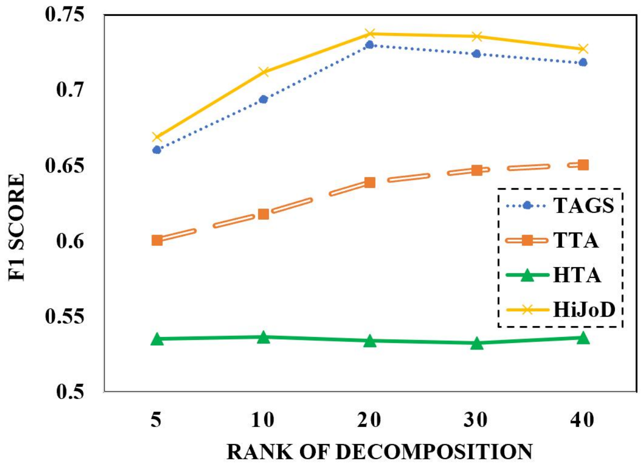


Figure 6.4: F1-score of using individual embeddings vs. HiJoD. Even when the best rank of each embedding is not used, HiJoD outperforms individual decompositions

6.5 Related work

Ensemble modeling for misinformation detection. A large number of misinformation detection approaches focus on a single aspect of the data, such as article content [52, 183], user features [**?**], and temporal properties [91]. There exist, however, recent approaches that integrate various aspects of an article in the same model. For example, in [79] the authors propose an ensemble model for finding fake news. In this approach, a bag of words embedding is used to model content-based information, while in this work, we leverage a tensor model i.e., $\mathcal{X}_{\texttt{TTA}}$ which not only enables us to model textual information, but also is able to capture nuanced relations between the words. The different sources of information used in [79] (user-user, user-article and publisher-article interactions) do not overlap with the aspects introduced here (hashtags and HTML features), however, in our experiments we show that HiJoD effectively combines both introduced aspects as well as the ones in [79]. In another work [81], news contents and user comments are exploited jointly to detect fake news. Although user comment is a promising aspect, still the main focus is on the words of comments. However, we use HTML tags and hashtags in addition to the textual content.

Semi-supervised learning / Label propagation models. The majority of mono-aspect modeling proposed so far leverage a supervised classifier. For instance, in [62] a logistic regression classifier is used which employs linguistic and semantic features for classification. In [70], authors apply a SVM classifier for content based features. Moreover, some works have been done using recurrent neural network (RNN) and Dynamic Series-Time Structure (DSTS) models [105, 132]. In contrary to the aforementioned works, we use a model which achieves very precise classification when leverage very small amount of ground truth. There are some proposed methods that mainly rely on propagation models. For example, in [73] the authors proposed a hierarchical propagation model on a suggested three-layer credibility network. In this work, a hierarchical structure is constructed using event, sub-event and message layers, even though a supervised classifier is required to obtain initial credibility values. In [75], a PageRank-like credibility propagation method is proposed to apply on a network of events, tweets and users. In this work, we leverage belief propagation to address the semi-supervised problem of misinformation detection. We show that proposed approach outperforms state-of-the-art approaches in label scarcity settings.

6.6 Conclusions

In this chapter, we propose HiJoD, a 2-level decomposition pipeline that integrates different aspects of an article towards more precise discovery of misinformation on the web. Our contribution is two-fold: we introduce novel aspects of articles which we demonstrate to be very effective in classifying misinformative vs. real articles, and we propose a principled way of fusing those aspects leveraging tensor methods. We show that HiJoD not only is able to detect misinformation in a semi-supervised setting even when we use only 10% of the labels but also an order of magnitude faster than similar ensemble approaches in terms of execution time. Experimental results illustrates that HiJoD achieves F1 score of roughly 74% and 81% on Twitter and Politifact datasets respectively which outperforms state-of-the-art content-based and neural network based approaches.

7

K-Nearest Hyperplanes Graph (KNH) for Misinformation Detection

Graphs are efficient structures for representing datapoints and their relationships, and they have been largely exploited for different applications such as community detection, nearest neighbors modeling etc. To account the pairwise modeling limitation of simple graphs, hypergraphs are used to model higher-order relationships between nodes. In hypergraphs, the edges are defined by a set of nodes i.e., hyperedges to demonstrate the higher-order relationships between the data. Our work is inspired by the following: despite hyperedges' capacity to model higher-order relationships between nodes, there is no explicit higher-order generalization for nodes themselves. In this work, we introduce a novel generalization of graphs i.e., K-Nearest Hyperplanes graph (KNH) where the nodes (or what we call, hypernodes) are Euclidean subspaces, facilitating multi-aspect modeling of the entities. To demonstrate the potential of KNH graphs, we evaluate their use on two multi-aspect datasets for misinformation detection. Our experimental results demonstrate that multi-aspect modeling of articles using KNH outperforms the classic KNN graph in terms of classification performance and robustness against noisy aspect representation.

7.1 Introduction

Nowadays, the abundance of information or aspects that describe a particular entity such as a document or a news article, brings about a more holistic view of the data. For instance, the publisher, textual content or users who publish the articles are all important aspects that can be encapsulated and leveraged to determine the trustworthiness of a particular article [81, 133]. This suggests the importance and necessity of effective and representative multi-aspect modeling techniques.

Over the last decades, multiple approaches have been introduced for complex data representation. Of these, graphs are employed extensively by mathematicians and computer scientists for countless applications including anomaly detection [7], biological network analysis [120], and misinformation finding and article classification [52]. For instance, in [52, 133], the graph data structure in form of a K-Nearest Neighbours graph (KNN) is used to model similarity of articles (nodes) and pairwise relationships using edges that connect them. The problem of representing

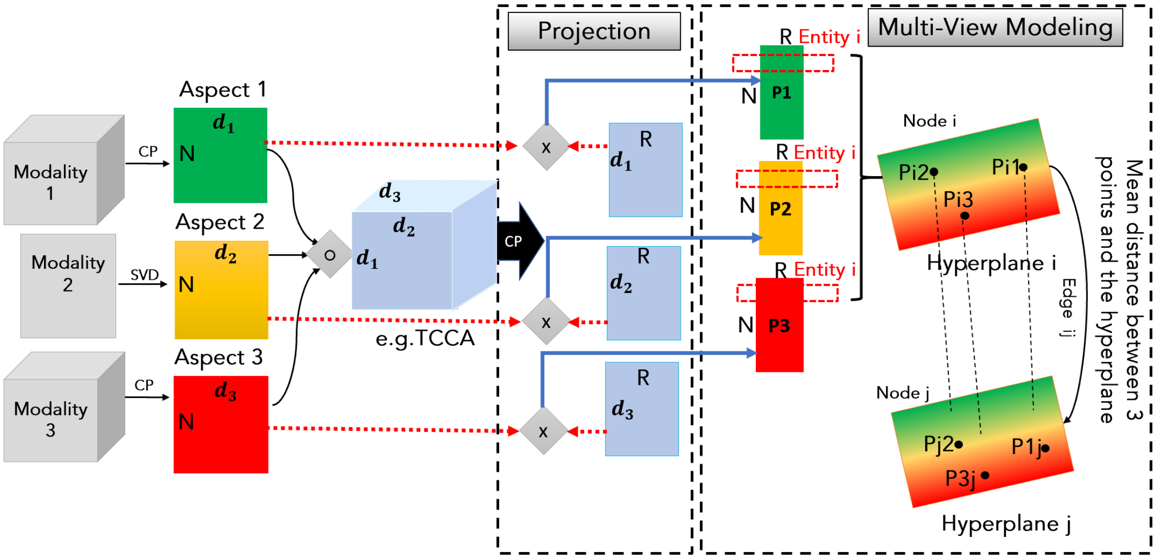


Figure 7.1: Overview of K-Nearest Hyperplane graph (KNH) for multi-aspect modeling and ensemble learning.

nearest neighbors across multiple data aspects via a single KNN graph is not well-studied; the KNN graph naturally is subject to the definition of the representation, which is nontrivial to combine across multiple aspects. Several works propose merging all aspects into a single joint structure by concatenating embeddings, and using the concatenated form for construction of the graph. For instance, to take advantage of multi-aspect information and KNN graph at the same time, in [133] joint patterns are derived from different aspects of the articles and then the patterns are leveraged to construct a KNN graph. However, this alternative has to tackle the ad hoc challenge of weighting aspect contributions and avoiding any one aspect from dominating the distance measure. Another conceivable alternative in using KNNs for higher order representation is to create a separate graph for each aspect, and merge them somehow, but considering the high dimensionality of many real world datasets, this solution could be expensive and complex.

Our work considers a third option for extending the KNN concept to multi-aspect data, motivated by the idea of hypergraphs: hypergraphs are generalizations of graphs, where edges correspond to subsets of “nodes” that are similar in terms of features or distance. Contrary to hypergraph learning techniques which aim to define hyperedges to model high order relationships between nodes, our work instead focuses on modeling higher order representation of the nodes (hypernodes) themselves. To this end, we propose to first capture the entity representation with respect to different aspects, and then leverage these for defining higher order geometric subspaces which are in fact, manifold representations of the nodes. In other words, we introduce a novel generalization of graph for multi-aspect modeling and ensemble learning. The intuition behind this approach is to define a common “feature space” comprising multiple-aspects of nodes i.e., articles in this work, using geometric objects which is not only capable of multi-aspect modeling of the articles but also might be exploited to predict missing or arriving features. Moreover, defining a common feature space to model entities enables us to use geometric techniques to calculate different higher order relationships such as intersection, orthogonality, distance etc. between different articles (nodes). The contributions of this chapter are as follows:

- **A novel graph based modeling for multi-aspect representation of articles using Euclidean subspaces.** We introduce a generalization of KNN graph i.e., KNH where the nodes (hypernodes) are defined by hyperplanes.
- **A novel embedding to represent hypernodes in terms of relative location in the space.** We propose a new embedding which is a generalization of Euclidean distance between subspaces and encodes the relative location of hypernodes.
- **Experimental results on real-world datasets for fake news detection.** We examine the

KNH modeling and ensemble learning on two real-world datasets including textual, user and social context aspects of news articles.

7.2 Problem formulation

The problem formulation of multi-aspect modeling and classification using K-Nearest Hyperplanes graph is:

Given N entities (articles), M embeddings of size $N \times d_m$, $m = 1, \dots, M$ for M aspects of the entities s.t. row i of each embedding corresponds to a d_m-dimensional representation of entity i with respect to that aspect.

Find a joint representation for the entities

Such that the manifold structures are preserved when used for modeling and classification.

One simple solution that comes into mind is to stack embeddings and form a long vector and use KNN graph for modeling and classification. However, by doing so, we may destroy potentially useful structures. We address this problem by defining a M-dimensional flat in R-dimensional space for each entity where R is the dimensionality of aspect representation. For example, if we have 2 aspects, we model each entity by a line and if we have 3 aspects, we model entities using a plane. These flats are generalized form of points (nodes) in KNN graph. Later on, we leverage geometrical properties of hyperplanes to calculate a manifold distance (edge) between entities. We will show that, retaining the proposed representation results in better quality in downstream classification tasks.

7.3 Hyperplane modeling and KNH graph

In what follows, the hyperplane modeling and classification will be described step by step.

7.3.1 Modeling aspects using tensor/matrix and decomposing them into aspect embeddings

For modeling individual aspects of the entities, we leverage tensors (matrices) such that one mode of each tensor (matrix) correspond to the entities' representations that formed by other aspects (modes) for instance article representation using co-occurrences of words. To capture hidden patterns with respect to the considered aspect that are shared between different entities, we decompose the matrix (tensor) into factor matrices. With that said, the first step of the proposed approach is to decompose M aspects of the entities (matrices or tensors) into M factor matrices each of which of size $N \times d_m$ where N is the number of entities and d_m is the size of embedding space defined by rank of decomposition. In fact, each embedding comprises latent patterns of the entities with respect to the considered aspect.

7.3.2 Projecting embeddings into a common space

Previous step provides us with M embeddings of size $N \times d_m$, $m = 1, \ldots, M$. Now, we want to leverage all these embeddings to create a manifold or multi-aspect description of entities. In fact, the goal is to define a new space that consolidates all M representation of the entities. Since these matrices represent the entities in different spaces, we may need to project all theses representations into a common space such that the correlation between all of them is maximized. One solution that comes into mind for this requirement is applying CCA/TCCA on factor matrices (depends on the number of aspects as discussed in chapter 2. In case of TCCA, first we create a tensor $\mathcal{X}$ of size $d_1 \times d_2 \ldots \times d_m$ out of all M embeddings which is equivalent to the covariance tensor. Then we apply CP decomposition to find canonical matrices and project embedding matrices into a common space as illustrated in Fig. 7.1. By doing so, we are effectively deriving a low dimensional representation, which also helps to get rid of coarse of dimensionality and noisy representation.

7.3.3 Creating M-dimensional flat in R-dimensional space for each entity

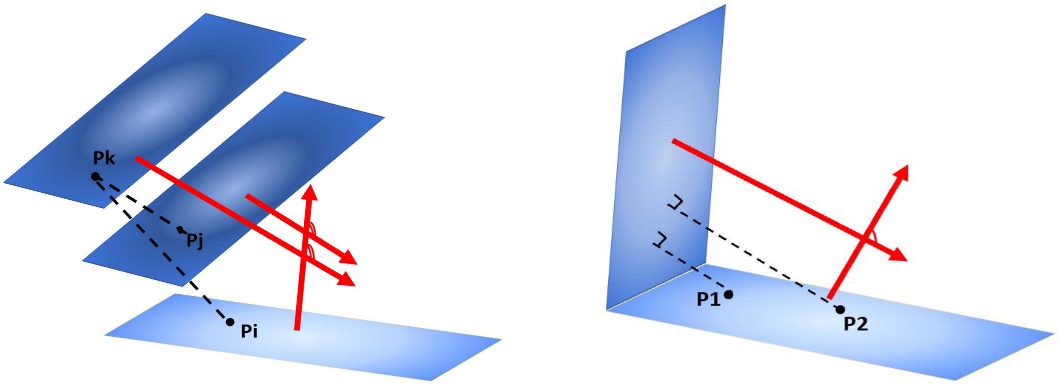
The output of the projection step is M matrices of size $N \times R$ where the rows of each matrix corresponds to one datapoint (vector) in R-dimensional space such that the correlation between rows i of all matrices is maximized. Now, there are M datapoints for each entity. We can leverage these datapoints to define a flat for each entity. For example, if we have 2 embeddings, we can define 2-flats (lines) in R-dimensional space as follows:

$$\mathbf{x}_1 = a_1 t + b_1, \mathbf{x}_2 = a_2 t + b_2, \ldots, \mathbf{x}_n = a_R t + b_R \tag{7.1}$$

where the number of parametric equation is equal to $R - 2$. In general, a flat of dimension $R - k$. is described by k parametric equations.

7.3.4 Creating KNH graph for classification

. Previous step results in N, M-flats in R-dimensional space, each of which a manifold representation of entity i, $i = 1, \ldots, N$. Now we create a graph such that each node of the graph is a M-flat in R-dimensional space and the edges between the nodes show the multilateral similarity between



(a) Similarity based on angles between normal vectors.

(b) Similarity based on point-plane Euclidean distance

Figure 7.2: Comparing different approaches for measuring similarity of hyperplanes. a)Similarity based on angles between normal vectors. As depicted, angle between the hyperplanes i.e., angle between the normal vectors is not a proper metric for measuring the similarity e.g. although both planes **j**, **k** make the same angel with plane *i*, plane **j** is closer to **i** than **k**. b)Similarity based on point-plane Euclidean distance of points of one plane to another plane. The closer the points are to the intersection of the hyperplanes, the smaller the distance is.

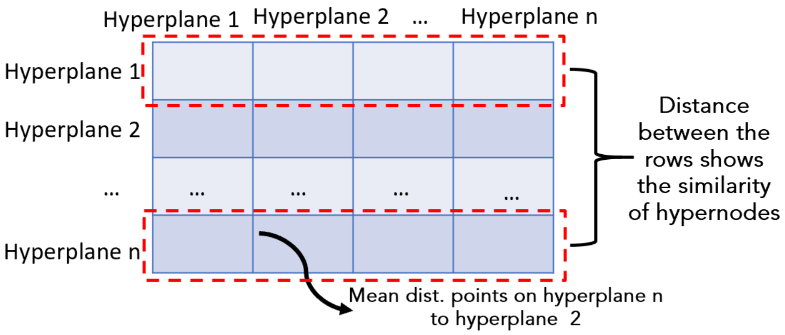

Hyperplane 1
Hyperplane 2
...
Hyperplane n
Hyperplane 1
Hyperplane 2
...
Hyperplane n
...
...
...
...
Distance between the rows shows the similarity of hypernodes
Mean dist. points on hyperplane n to hyperplane 2

Figure 7.3: Defining an embedding that shows pairwise distances between the nodes along with the relative position of each node in the space with respect to other nodes.

the flats. The question that raises here is: ”how to calculate the distances between M-flats? if we are to use the Euclidean distance between the hyperplanes, they ought to be parallel, otherwise the distance between them is zero.

One option to address the distance calculation issue is to measure the angles between the hyperplanes (the angles between the normal vectors). However, considering the situation illustrated

Algorithm 7 KNH modeling

```
Input: $\mathbf{M}_1$,$\mathbf{M}_2$ of size $N \times d_m$
Output: $K$-nearest hyperplane (line) graph

$\mathbf{W} = CCA(\mathbf{M}_1, \mathbf{M}_2, R)$
$\mathbf{p_1} = \mathbf{W}(:, 1 : R)$
$\mathbf{p_2} = \mathbf{W}(:, R + 1 : 2R)$
// Defining the 2-flats (lines)
$x_1 = a_1 t + b_1, x_2 = a_2 t + b_2, \ldots, x_n = a_R t + b_R$
for all $i, i = 1, \ldots, N$ do
    for all $j, j = i, \ldots, N$ do
        $\mathbf{p_{1j}} = \mathbf{p_1}(\mathbf{j}, :) - \mathbf{p_1}(\mathbf{i}, :)$
        $\mathbf{p_{2j}} = \mathbf{p_2}(\mathbf{j}, :) - \mathbf{p_1}(\mathbf{i}, :)$
        $\mathbf{p_{12i}} = \mathbf{p_2}(\mathbf{i}, :) - \mathbf{p_1}(\mathbf{i}, :)$
        $t_1 = dot(\mathbf{p_{1j}}, \mathbf{p_{12i}})/dot(\mathbf{p_{12i}}, \mathbf{p_{12i}})$
        $t_2 = dot(\mathbf{p_{2j}}, \mathbf{p_{12i}})/dot(\mathbf{p_{12i}}, \mathbf{p_{12i}})$
        $\mathbf{d_1} = (\mathbf{p_{1j}} - t_1 * \mathbf{p_{12i}})$
        $\mathbf{d_2} = (\mathbf{p_{2j}} - t_2 * \mathbf{p_{12i}})$
        $\mathbf{D}(i, j) = (sqrt(sum(\mathbf{d_1}.^2)) + sqrt(sum(\mathbf{d_2}.^2)))/2$

    end
end
Create˙Graph($\mathbf{D}$) //Generate graph from embedding $\mathbf{D}$
```

in Fig. 7.2(a), where plane j and k are parallel to plane i and form the same angle with plane i, this is not an ideal option. In this situation, there might be a point $\mathbf{p_i}$ lying on plane i which is closer to a point $\mathbf{p_j}$ on plane j than a point $\mathbf{p_k}$ on plane k. The angular distance is not capable to capture this difference. Another option is the point-hyperplane distance in 2.29 which may capture the insightful difference depicted in 7.2(b). The closer the points are to theintersection of the hyperplanes, the smaller the $d_{point-hyperplane}$ gets.

With that being said, we define the following distance as the distances between the hypernodes. we take the mean of Euclidean distances between each one of the M datapoints defining a hyperplane i to the hyperplane j using equation 2.29. As mentioned earlier, each data point has an embedding representation in the space. For classifying the hypernodes based on their similarity we define a vector representation for each hyperplane in the space using the second scenario. We propose to create an $N \times N$ matrix such that entry ij of this matrix is equal to the distance between hyperplanes i and j as described above. As the base case, if the nodes are points (1 aspect), the relative distances between each node and the rest of the nodes in the space is equivalent to the position of the nodes in the space. So, the pairwise distances are equivalent to pairwise Euclidean distance. Likewise, if we have more than two aspects, this matrix embeds relative locations of the hypernodes in the space and so on and so forth. Therefore, this could be considered as generalization of Euclidean distance between the hypernodes. Fig. 7.1 and Algorithm 7 demonstrate the overview and etails of the KNH method.

7.3.5 Complexity analysis

Time complexity of KNH is dominated by hyperplanes calculations or more precisely calculating the normal vectors i.e. calculating N cross product each of which of size M(number of aspects) or $O(NM)$. The complexity of edge calculation is the same as the time complexity of KNN and is $O(N^2)$. Meanwhile, there are approaches to speed up the normal quadratic nearest neighbor calculation by approximation [19].

As shown in [186], the space and time complexity of TCCA prepossessing, which is independent of the proposed approach, depend on the size of the tensor and ALS calculations and are $O(d_1 d_2 \cdots d_m)$ and $O(trd_1 d_2 \cdots d_m)$ respectively.

7.4 Experiments

In this section, we evaluate the efficacy of KNH method against KNN for multi-aspect modeling and classification. We experiment on a 2-aspects document-publisher dataset extracted from Twitter's tweets [1] and another 2-aspects news article dataset extracted from FakeNewsNet dataset[2] but this time we experiment on different sets of features, i.e., user-news interaction and the publisher-news interaction aspect. Henceforth, we refer to the first dataset as Twitter dataset and to the second dataset as Politifact dataset.

7.4.1 Implementation

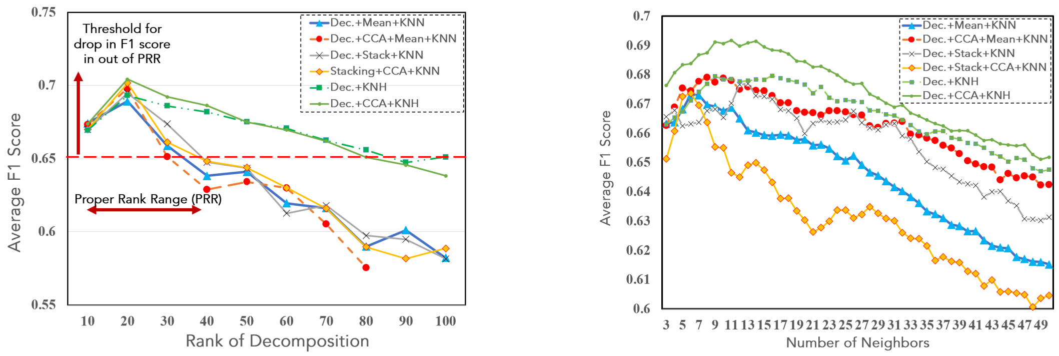
We implemented all experiments in Matlab using Tensor Toolbox version 2.6 [9]. For rank of decomposition (dimensionality) r_m and the number of nearest neighbors K we grid searched the values between range $1-100$ for r_m and $1-50$ for K. We report the average F1 score of all methods for 10 runs.

7.4.2 Baselines

The goal of this chapter is to generalize the KNN graph. Thus, we compare the KNH graph with different variations of KNN based approaches. It is worth mentioning that, since we only have two aspects, we use CCA instead of TCCA which is applicable for when we have more than two aspects. The baselines are:

[1] https://github.com/Saraabdali/Fake-News-Detection-˙ASONAM-2018

[2] https://github.com/KaiDMML/FakeNewsNet



(a) Average F1-score for 10 runs of decomposition using k=15 when modeling the articles by KNN and KNH graphs. The results suggest that KNH leads to higher performance especially by increasing the rank.

(b) Average F1-score for 10 runs of decomposition using R=30 when modeling the articles by KNN and KNH graphs. As depicted, for all number of neighbors KNH results in higher classification performance.

Figure 7.4: Average F1 score of KNH and KNN modeling for different ranks and number of neighbors.

- **Dec.+Mean+KNN**: After decomposing the aspects, we calculate the pairwise (articles i and i) distances between vectors of each aspect embedding and then take the average of distances and consider it as the edge between the corresponding articles in the KNN graph.

- **Dec.+CCA+Mean+KNN**: Same as the previous method but the embeddings are projected before calculating the average pairwise distances.

- **Dec.+Stack+KNN**: After decomposing the aspects we stack all aspect embeddings and create a long vector for each article. Then we calculate the distances between these long vectors and create a KNN graph where the nodes are articles and the edges are the distances between the corresponding long vectors.

- **Dec.+Stack+CCA+KNN**: Same as the previous approach but the embeddings are projected before stacking.

We compare the above baselines against **Dec.+KNH** and **Dec.+CCA+KNH** i.e. two versions of the

Twitter dataset	
Features	**Total Number**
Total number of words	18853
Total number of domains	652
Total news articles	335 (Real)/317 (Fake)

Table 7.1: Twitter dataset description

proposed approach where we add or remove the projection from the pipeline to investigate the effect of projection in KNH modeling as well.

7.4.3 Experiment 1: article modeling with 2-flats out of text and domain aspects

In this section, we discuss the implementation details and experimental results of the first experiment.

Description of dataset and aspects

As mentioned earlier, for the first experiment, we use the dataset we introduced in chapter 3. This dataset comprises multi-aspect information about news articles and the Twitter tweets that shared these articles as URL links. As mentioned before, the labels are extracted using the B.S. Detector browser. Table 7.1 describes the details of this dataset.

As the base case, we consider 2-aspects modeling or 2-flat (line) modeling of articles. we leverage the most promising aspects i.e., TTA and Tags introduced in chapter 6. The following describes these aspects:

- **(Term, Term, Article) Tensor (TTA)**: We use the TTA model that we described in chapter 3. In this tensor, we find the co-occurred words by sliding a window across the article text. This yields to a word by word matrix for each article. By stacking all these matrices, a three mode tensor is created where the first two modes are word modes and the third one is the article mode.

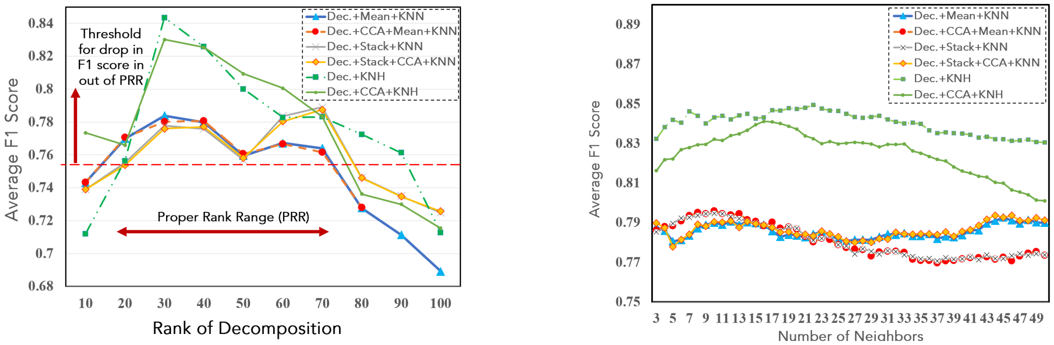
- **(Article, Domain feature) Matrix (TAGS)**: We use the matrix introduced in chapter 6. As mentioned in this chapter, the rationale behind using is that different domains have different web styles. For instance, trustworthy publishers like BBC and CNN tend to have standard webpages while unreliable resources often have messy webpages full of Ads, pop-ups etc.

Modeling articles with 2-flats

To capture the article representation with respect to the introduced aspects above, we use CP/-PARAFAC and SVD to decompose $\mathcal{X}_{TTA}$ and $\mathbf{X}_{TAGS}$ into factor matrices corresponding to the article mode. After decomposing $\mathcal{X}_{TTA}$ and $\mathbf{X}_{TAGS}$ into aspect matrices, we apply the CCA on factor matrices. The result provides us with the canonical matrices where the row i of these matrices correspond to datapoints (vectors $\mathbf{p_{1i}}$ and $\mathbf{p_{2i}}$ which can be leveraged to define a line or a 2-aspects representation of news article i. Then we construct a graph such that the lines are the hypernodes and the edges are defined as the distances between embedding representation of the lines as discussed earlier. Finally, for classification, we apply belief propagation method in [89] to propagate 40% of the labels throughout KNH in a *semi-supervised* manner.

Evaluation

The average F1 score achieved by 10 runs of KNH and all baseline methods for different ranks of decomposition and number of neighbors k is demonstrated in Fig. 7.4. As shown in part (a), KNH and all baseline methods achieve the best classification performance for the ranks in range 10-30. We call this range the Proper Rank Range or PRR. In this experiment, the KNH negligibly outperforms KNN based methods in PRR. However, by increasing the rank which means adding noisy components to the representations, the KNH modeling shows admissible robustness against noise. On the contrary, the F1 score of all KNN based methods decreases dramatically. Therefore, we may conclude that the KNH modeling is considerably more robust against noisy representation. Meanwhile, using CCA for the proposed approach slightly increases the F1 score specially in low rank settings. However, the effect of projection is negligible for all under study methods. To demonstrate the trend of F1 score when using different number of neighbors, we executed all methods for arbitrary rank R=30 and k in range 3-50 and as shown, KNH approaches keep the same trend as the previous experiment for different values of k.



(a) Average F1-score for 10 runs of decomposition and k=20 when modeling the articles by KNN and KNH graphs. As shown, KNH modeling achieves higher F1 scores in comparison to KNN modeling.

(b) Average F1-score for 10 runs of decomposition and R=30 when modeling the articles by KNN and KNH graphs. As shown, KNH modeling achieves higher F1 scores in comparison to KNN modeling.

Figure 7.5: Average F1 score for different rank of decomposition R and different number of neighbors K.

Politifact dataset		
Features	**Real**	**Fake**
Total news articles	432	624
Total number of tweets	116005	261262
Total news with social engagement	342	314
Total number of Users	214049	700120

Table 7.2: FakeNewsNet dataset description

7.4.4 Experiment 2: article modeling with 2-Flats (lines) out of user-news and publisher-news interactions aspects

In this section, we discuss the implementation details and experimental results of the second experiment.

Description of dataset and aspects

For the second experiment, we again model the news articles using two promising aspects but this time we choose another dataset to investigate the efficacy of KNH on data with different nature. To this end, we use the FakeNewsNet dataset [90] [3] which consists of users and publisher information for news articles crawled from PolitiFact web site. The details of the FakeNewsNet dataset is reported in table 7.2.

As suggested in [79] we use the following aspects:

- **User-News Interaction $\mathbf{X_{UN}}$**: We create a matrix to model the users who tweet a specific news article. The rows of this matrix are users and the columns are the news IDs.
- **Publisher-News Interaction $\mathbf{X_{PN}}$**: We create a matrix to model the publishers that publish a specific news article. The rows of this model are the publishers and the columns are the news IDs.

[3] https://github.com/KaiDMML/FakeNewsNet

Modeling articles with 2-flats

To capture the latent representation of articles in aspect spaces, we first decompose the $\mathbf{X_{UN}}$ and $\mathbf{X_{PN}}$ with SVD rank r_m. After applying CCA, we create a KNH graph in which the hypernodes are lines (2-flats) in r_m dimentional space. Then we construct KNH graph as discussed earlier. Finally, like the previous experiment, we leverage the belief propagation algorithm for the semi-supervised classification of articles.

Evaluation

As shown in Fig. 7.5 part (a), all methods achieve the best classification performance for the ranks in range 20-70. So, we consider this range as the PRR of this experiment. As illustrated, the proposed approach achieves considerably better performance in comparison to KNN based methods in this range. Moreover, by increasing the noisy components KNH graph keeps a higher threshold of drop in F1 score than KNN based methods which again suggests that the KNH modeling is considerably more robust against noisy representations. Applying CCA increases the performance in the low rank settings specially for KNH based modeling. However, the effect of projection is negligible for KNN based methods.

The trend of F1 score when using different K is demonstrated in part (b). For arbitrary rank R=30 and k in range 3-50 the KNH outperforms all KNN based methods and keep the trend for all values of k.

After experimenting on two datasets with different aspects, we can summarize the comparison between KNH and KNN modeling as follows:

- **Performance**: KNH modeling outperforms KNN based methods for all ranks of decomposition. This performance is considerable for datasets where the aspects used in KNH have the same nature e.g., aspects in FakeNewsNet dataset.

- **Robustness of KNH against improper representations**: By increasing the rank, the performance of KNN modeling decreases dramatically but KNH modeling is more robust in such improper rank settings e.g., noisier/redundant representation and achieves higher threshold of performance.

- **CCA increases the performance in proper rank range (PRR)**: Even though the effect of CCA is negligible for KNN based modeling, using it in KNH pipeline, improves the performance in PRR settings. In fact, it derives a proper representation by decreasing the coarse of dimensionality and noisy/redundant information. However, as shown, in improper rank settings (high rank for experiments of this chapter), using CCA may sometimes lead to useful information loss.

7.5 Related work

7.5.1 Ensemble learning for fake news detection

The majority of misinformation detection approaches focus on a single aspect of the data and mostly the article content [137, 183]. There are also works that leverage other aspects like user features [184], and temporal properties [91]. However, there exist few ensemble approaches that consider all different aspects simultaneously. For instance, in [79] the authors propose an ensemble model by merging a bag of words embedding, user-user, user-article and publisher-article interactions. In another work [81], news contents and user comments are consolidated to detect the misinformation jointly. In this chapter, we leverage both promising aspects introduced in both [79] and chapter 6 but this time with a novel multi-aspect graph modeling and formulation.

7.6 Conclusions and future work

In this work, we introduce a novel multi-aspect modeling of the articles i.e. KNH graph by generalizing the classic KNN graph. We propose hypernodes in form of hyperplanes (m-flats) which are defined by vectors each of which a representation of the articles with respect to a different aspect. Moreover, for classifying the articles, we propose a novel embedding that encodes the relative position of the hypernodes in the space. We experiment on two real world datasets. We observe that the KNH graph not only outperforms the KNN graph in terms of F1 score but also is significantly more robust against improper rank representations.

Our work shows great promise of KNH for multi-aspect modeling. In the future, we plan to extend our investigation on 1) higher dimensional mathematical formulation of hypernodes, which requires an additional discussion and we skipped it here due to the space limitation, 2) capability of hypernodes i.e., common spaces for extrapolating missing/unknown features, and 3) defining hyperedges that consider higher order relationships between the subspaces. We reserve all the aforementioned directions for future work.

Part IV

Vision-based Techniques for Misinformation Detection

8

Tensor Emdedding for Misinformation Detection from Website Screenshots

Can the look and the feel of a website give information about the trustworthiness of an article? In this chapter, we propose to use a promising, yet neglected aspect in detecting the misinformativeness: the overall look of the domain webpage. To capture this overall look, we take screenshots of news articles served by either misinformative or trustworthy web domains and leverage a tensor decomposition based semi-supervised classification technique. The proposed approach i.e., `VizFake` is insensitive to a number of image transformations such as converting the image to grayscale, vectorizing the image and losing some parts of the screenshots. VizFake leverages a very small amount of known labels, mirroring realistic and practical scenarios, where labels (especially for known misinformative articles), are scarce and quickly become dated. The F1 score of VizFake on a dataset of 50k screenshots of news articles spanning more than 500 domains is roughly 85% using only 5% of ground truth labels. Furthermore, tensor representations of VizFake, obtained in an unsupervised manner, allow for exploratory analysis of the data that provides valuable insights into the problem. Finally, we compare VizFake with deep transfer learning, since it is a very popular black-box approach for image classification and also well-known text text-based methods. VizFake achieves competitive accuracy with deep transfer learning models while being two orders of magnitude faster and not requiring laborious hyper-parameter tuning.

8.1 Introduction

Despite the benefits that the emergence of web-based technologies has created for news and information spread, the increasing spread of fake news and misinformation due to access and public dissemination functionalities of these technologies has become increasingly apparent in recent years. Given the growing importance of the fake news detection task on web-based outlets, researchers have placed considerable effort into design and implementation of efficient methods for finding misinformation on the web, most notably via natural language processing methods [**?**,30,131,137]. intended to discover misinformation via nuances in article text. article's text Although utilizing textual information is a natural approach, there are few drawbacks: most notably, such approaches

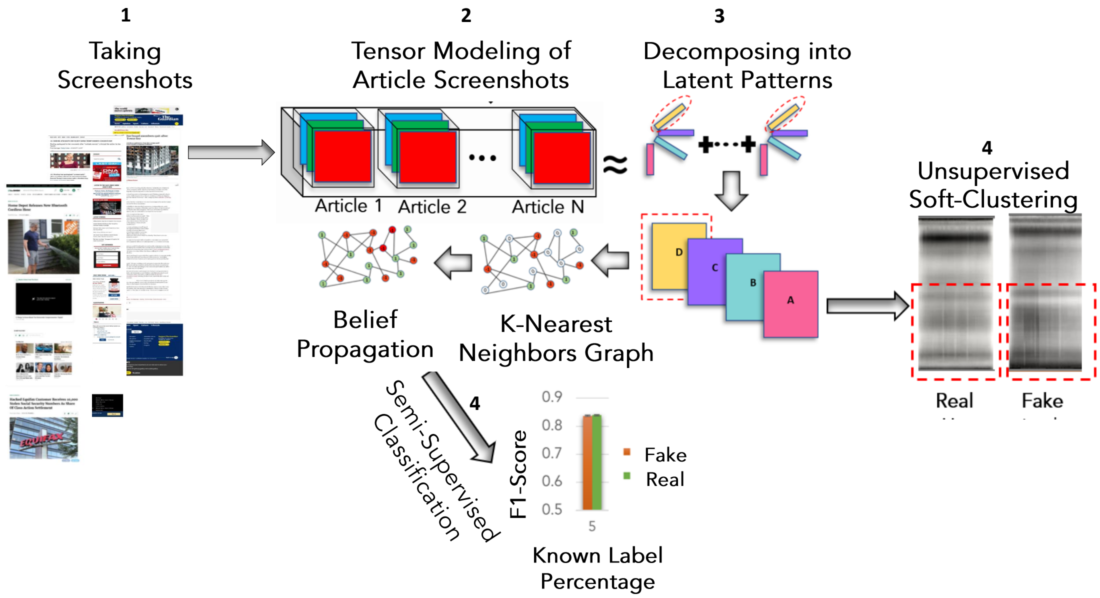

1
Taking Screenshots
2
Tensor Modeling of Article Screenshots
3
Decomposing into Latent Patterns
Article 1
Article 2
Article N
≈
D
C
B
A
4
Unsupervised Soft-Clustering
Real
Fake
Belief Propagation
K-Nearest Neighbors Graph
4
Semi-Supervised Classification
F1-Score
0.9
0.8
0.7
0.6
0.5
5
🟧 Fake
🟩 Real
Known Label Percentage

Figure 8.1: Creating a tensor-based model out of news articles' screenshots and decomposing the tensor using CP/PARAFAC into latent factors and then creating a nearest neighbor graph based on the similarity of latent patterns and leveraging belief propagation to propagate very few known labels throughout the graph. As illustrated, the F1 score of both real and fake classes is roughly 85% using just 5% of known labels. Moreover, VizFake has exploratory capabilities for unsupervised clustering of screenshots.

require complicated and time-consuming analysis to extract linguistic, lexical, or psychological features such as sentiment, entity usage, phrasing, stance, knowledge-base grounding, etc. Moreover, the problem of identifying misinformativeness using textual cues is challenging to define well, given that each article is composed of many dependent statements (not all of which are fact-based) and editorialization. Finally, most such approaches require extremely large labeled sets of misinformative articles, which are often unavailable in practice due to lack of reliable human annotators, as well as quickly become "dated" due to shift in topics, sentiment, and reality and time itself. These article-based labels inherently result in event-specificity and bias in resulting models, which can lead to poor generalization in the future for different article types.

In this work, we take a step back to tackle the problem with a human, rather than an algorithmic perspective. We make two choices that are not made jointly in prior work. Firstly, we tackle misinformation detection by leveraging a domain-level feature. Secondly, we focus on the discovery of misinformation using visual cues rather than textual ones. We expand upon these two points below. Firstly, leveraging domain features for misinformation detection is not only an easier but also a likely more fruitful/applicable problem setting in practice. In reality, most highly reputed news sources do not report misinformative articles due to high editorial standards, scrutiny, and expectations. For example, the public fallout from misinformation being spread through famous organizations like CNN or BBC would be disastrous. However, there are many misinformation farms and third-parties which create new domains with the intent of deceiving the public [17]. Moreover, these actors have little incentive to spread real articles in addition to fake ones. Thus, in most cases, domain feature could prove to be a better target to stymie the spread of misinformation. Conveniently, several crowd-sourced tools and fact-checkers like BS Detector [1] or Newsguard [2] provide domain-level labels rather than article-level, which we utilize here.

Secondly, visual cues are a promising, yet underserved research area, especially in the context of misinformation detection. While past literature in text-based methods in this space is rich (see [115] for an overview), prior work on visual cues is sparse. Past works [57, 74, 150] primarily focus on doctored/fake-news associated images and visual coherence of images with article text. However, since these works are limited to fake news which spreads with images, they are inapplicable for articles which do not incorporate multimedia. Moreover, these works all have inherent article specificity, and none consider the overall visual look and representation of the host-

[1] http://bsdetector.tech/

[2] https://www.newsguardtech.com

ing domain or website for a given article. Intuitively and anecdotally, in contrast to unreliable sources that tend to be visually messy and full of advertisements and popups, trustworthy domains often look professional and ordered. For example, real domains often request users to agree to privacy policies, have login/signup/subscription functionalities, have multiple featured news articles clearly visible, etc. Conversely, strong tells for fake domains tend to include errors, negative space, unprofessional/hard-to-read fonts, and blog-post style [32, 180, 187]. Fig. 8.1 demonstrates this dichotomy with a few examples. While we as humans use these signals to quickly discern the quality and reliability of news sources without delving into the depth of the text, prior works have not directly considered them. Thus, we focus on bridging this gap with the assumption that many misinformative articles do not need to be read to be suspected.

Given these two facets, we ask: *"can we identify misinformation by leveraging the visual characteristics of their domains?"* In this work, we propose an approach for classification of article screenshots using image processing approaches. In contrast to deep learning approaches such as convolutional neural networks (CNNs) which take a relatively long time to train, are data-hungry, and require careful hyperparameter tuning, we propose a novel tensor-based semi-supervised classification approach which is fast, efficient, robust to image resolution, and missing image segments, and data-limited. We demonstrate that our approach henceforth refereed to as `VizFake`, can successfully classify articles into fake or real classes with an F1 score of 85% using very few (i.e., $<$ 5% of available labels). Summarily, our major contributions are as follows:

- **Using visual signal for modeling domain structure**: We propose to model article screenshots from different domains using a tensor-based formulation.

- **Fast and robust tensor decomposition approach for classification of visual information**:

We propose a tensor-based model to find latent article patterns. We compare it against typical deep learning models. `VizFake` performs on par, while being significantly faster and needless to laborious hyperparameter tuning.

- **Unsupervised exploratory analysis**: Tensor-based representations of `VizFake` derived in an unsupervised manner, allow for interpretable exploratory analysis of the data which correlate with existing ground truth.
- **Performance in label-scarce settings**: In contrast to deep learning approaches, `VizFake` is able to classify news articles with high performance using very few labels, due to a semi-supervised belief propagation formulation.
- **Experimenting on real-world data**: We evaluate `VizFake` on a real-world dataset we constructed with over 50K news article screenshots from more than 500 domains, by extracting tweets with news article links. Our experiments suggest strong classification results (85% F1 score) with very few labels (< 5%) and over two orders of speedup compared to CNN-based methods.

The remainder of this chapter is organized as follows: First, the proposed `VizFake` is described. Next, we discuss the implementation details and the dataset. Afterwards, the experimental evaluation of the `VizFake` as well as variants and baselines is presented. Then, we discuss the related work, and finally we draw the conclusions.

8.2 Proposed method

Here, we discuss our formulation and proposed semi-supervised tensor-based approach i.e., VizFake method.

8.2.1 Problem formulation

We solve the following problem:

> **Given** (i) a collection of news domains and a number of full-page screenshots of news articles published by each domain and (ii) a small number of labels.
>
> **Classify** the unlabeled screenshots as misinformation or not.

8.2.2 Semi-supervised tensor-based method i.e VizFake

VizFake aims to explore the predictive power of visual information about articles published by domains. As we argued above, there is empirical evidence that suggests this proposition is plausible.Thus, we introduce a novel model to leverage this visual information. We propose a tensor-based semi-supervised approach that is able to effectively extract and use the visual cue which yields highly predictive representations of screenshots, even with limited supervision, also, due to its elegant and simple nature, allows for interpretable exploration. VizFake has the following steps:

Tensor-based modeling

The first step of `VizFake` refers to constructing a tensor-based model out of articles' screenshots. RGB digital images are made of pixels each of which is represented by three channels, i.e., red, green, and blue. So each image channel shows the intensity of the corresponding color for each pixel of the image.

A tensor is a multi-way array. We use a 4-mode tensor embedding for modeling news articles' screenshots. since each channel of an RGB digital image is a matrix, by stacking all three channels we create a 3-mode tensor for each screenshot and if we put all 3-mode tensor together, we create a 4-mode tensor out of all screenshots.

Tensor decomposition

We use CP/PARAFAC to decompose our 4-mode tensor $\boldsymbol{X}$ of dimensions $I \times J \times K \times L$ into a sum of outer products of four vectors as follows:

$$\boldsymbol{X} \simeq \Sigma_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r$$

where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, $\mathbf{c}_r \in \mathbb{R}^K$ $\mathbf{d}_r \in \mathbb{R}^l$.

We define the factor matrices as $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2 \ldots \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2 \ldots \mathbf{b}_R]$, $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2 \ldots \mathbf{c}_R]$ and $\mathbf{D} = [\mathbf{d}_1\ \mathbf{d}_2 \ldots \mathbf{d}_R]$ where $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$ and $\mathbf{D} \in \mathbf{R}^{L \times R}$ denote the factor matrices and R is the rank of decomposition or the number of columns in the factor matrices.

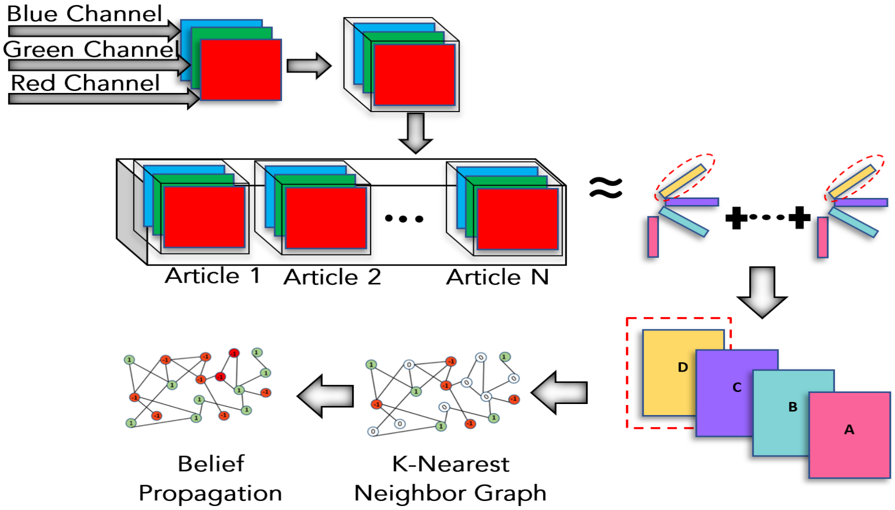


Figure 8.2: Proposed tensor-based modeling and semi-supervised classification of the screenshots i.e. VizFake.

Moreover, the optimization problem for estimating the factor matrices is defined as follows:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}} = \|X - \Sigma_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r\|^2 \tag{8.1}$$

For solving the optimization problem above we use algorithm 2.

Having the mathematical explanation above in mind, the second step of `VizFake` is decomposition of the proposed tensor-based model for finding the factor matrix corresponding to article mode, i.e., factor matrix **D** which comprises latent patterns of screenshots. We will leverage these latent patterns for screenshot classification.

.

Semi-supervised classification

The third and last step of VizFake is the classification of news articles using the factor matrix **D** corresponding to article mode resulted from the decomposition of the tensor-based model.

As we mentioned before, each factor matrix comprises the latent patterns of the corresponding mode in **R** dimensional space. Therefore, each row of factor matrix **D** is an **R** dimensional representation of the corresponding screenshot. So, we can consider each screenshot as a data point in **R** dimensional space. We create a K-nearest neighbor graph (K-NN) Graph by considering data points as nodes, and the Euclidean distance between the nodes as edges of the graph.

Since we model homophily (similarity) of screenshots patterns using a K-NN graph as explained above, we can leverage Belief Propagation in a semi-supervised manner to propagate very few available labels throughout the graph. A fast and linearized implementation of Belief propagation is proposed in [89] which solves the following linear system in equation 2.34. An overview of VizFake is depicted in Fig. 8.2.

8.3 Experimental evaluation

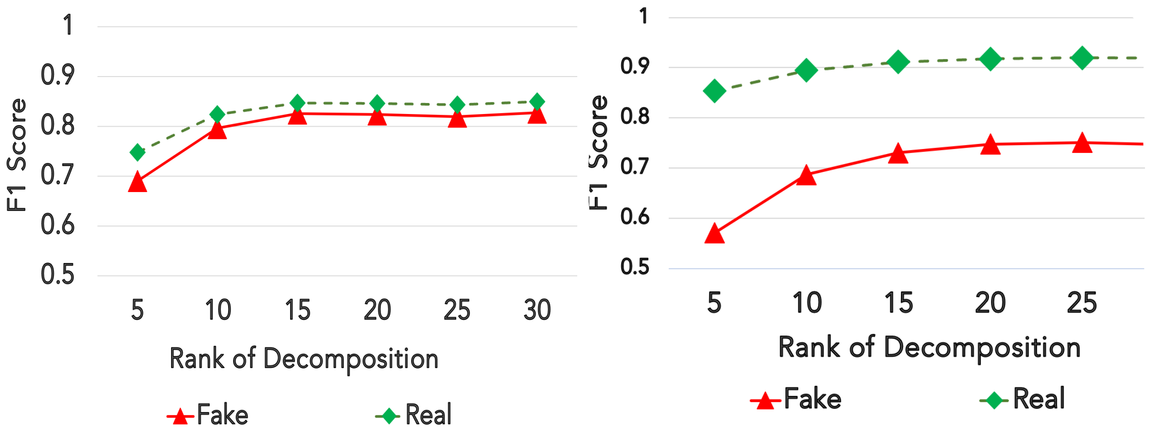
In this section, we first discuss implementation and dataset details and then report a set of experiments to investigate the effect of changing rank, resolution, and some image transformation on the performance of VizFake and then we compare it against CNN deep-learning model, text-base text-based approaches and webpage structure features.

8.3.1 Dataset description

Although collecting human annotation for misinformation detection is a complicated and time-consuming task, there exist some crowd-sourced schemes such as the browser extension "BS Detector" which provide a number of label options, allowing users to label domains into different categories such as biased, clickbait, conspiracy, fake, hate, junk science, rumor, satire, unreliable, and real. We use BS Detector as our ground truth and consider all of the nine categories above but "real" as "fake" class. We reserve a more fine-grained analysis of different "fake" categories for future work (henceforth collectively refer to all of those categories of misinformation as "fake").

We describe our crawling process in order to promote reproducibility, as we are unable to share the data because of copyright considerations. We crawled Twitter to create a dataset out of tweets published between June and August 2017 which included links to news articles. Then, we implemented a javascript code using Node.js open source server environment and Puppeteer library for automatically taking screenshots of scrolled news articles of our collected dataset.

- we took screenshots of 50K news articles equally from more than 500 fake and real domains i.e., a balanced dataset including 50% from fake and 50% real domains.
- To investigate the effect of class imbalance, we created an imbalanced dataset of the same size, i.e., 50k but this time we selected $\frac{2}{3}$ of the screenshots from real domains and $\frac{1}{3}$ of the data from fake ones.
- Although we tried to select an equal number of articles per each domain, sometimes fake domains do not last long and the number of fake articles published by them is limited. However, we show that this limitation does not affect the classification, because the result of the fake



(a) F1-Balanced

(b) F1-Imbalanced

Figure 8.3: F1 score of VizFake for different ranks when experimenting on balanced and imbalanced datasets. The best ranks for these datasets are 15 and 25 respectively.

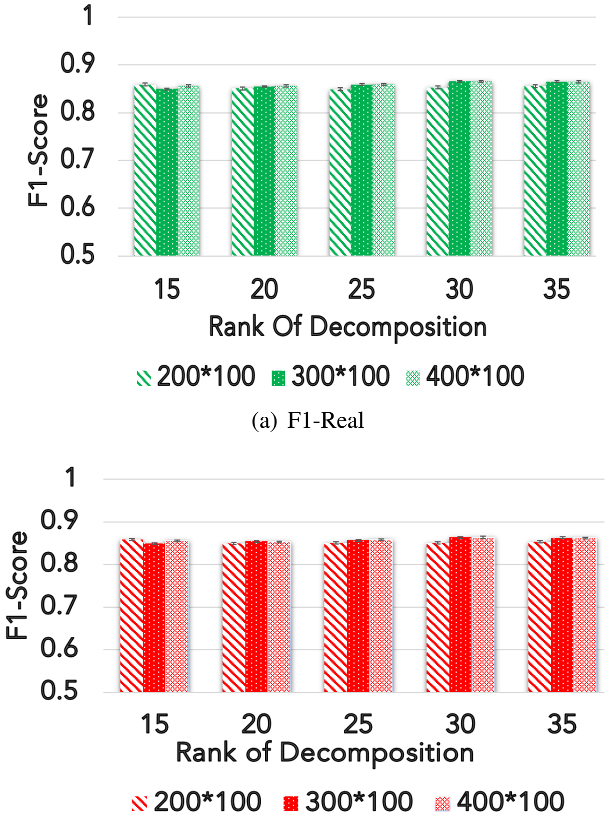
discrimination is almost same as the real class.

8.3.2 Implementation details

We used Matlab for implementing `VizFake` approach and for CP/PARAFAC decomposition we used Tensor Toolbox version 2.6 [3] [11]. For Belief Propagation, we used Fast Belief Propagation (FaBP) [89] which is linear in the number of edges. For finding the best rank of decomposition R and the number of nearest neighbors K for both balanced and imbalanced datasets, we grid searched the values between range 5-30 for R and 1-50 for K. Based on our experiments, we set R to 15 and 25 for balanced and imbalanced datasets, respectively and set K to 20 for both datasets. We measured the effectiveness of `VizFake` using widely used F1 score, precision, and recall metrics. We run all of the experiments 25 times and we report the average and standard deviation of the results for all mentioned metrics. The F1 score of different ranks for balanced and imbalanced datasets and both

[3] https://www.sandia.gov/ tgkolda/TensorToolbox/index-2.6.html

real and fake classes is shown in Fig. 8.3.



(a) F1-Real

(b) F1-Fake

Figure 8.4: F1 score of VizFake for different resolutions. F1 score increases slightly when experimenting on higher resolution images.

8.3.3 Investigating detection performance

First, we aim at investigating the detection performance of `VizFake` in discovering misinformative articles. A caveat in experimentation is that different articles even from the same domains may have different lengths, and thus screenshots of a fixed resolution may capture more or less information from different articles. However, fixed-resolution is an important prerogative for `VizFake` (and many others), thus we must use the same length for all screenshots.

Thus, we first evaluate the effect of resolution to choose a fixed setting for our model in further experiments. We experiment on screenshots of size 200 × 100, 300 × 100, and 400 × 100, and simultaneously evaluate the effect of different decomposition rank given the association with different amounts of information across resolutions. Fig. 8.4 shows the detection performance (F1 scores) across the above resolution settings and differing ranks from 15-35, using 10% seed labels in the belief propagation step.

Our experiments suggest that F1 score does increase slightly with higher resolutions and decomposition ranks, but the increases are not significant. We hypothesize that the invariance to changes in resolution is due to the fact that coarse-grained features like number of ads, positions of images in the article, and the overall format of the writing is still captured even at lower resolutions and the detection is not heavily reliant on the fine-grained features of the articles as shown in Fig. 8.5. This finding is promising, as it suggests valuable practical advantages in achieving high performance (88% F1 score) even using very low resolution or even icon size images and significant associated computational benefits. Thus, unless specified, in further experiments, we use 200×100 images.

8.3.4 Investigating sensitivity to image transformation

Next, we investigate different image-level Transformation to evaluate performance under such settings. Firstly, we consider the importance of colors in the creation of latent patterns and the role they play in the classification task via grayscaling. Next, we explore how vectorizing the channels of color screenshots improves the performance.

We first try to convert the color screenshots into grayscale ones using the below commonly used formula in image processing tasks [82]:

$$P = R \times (299/1000) + G \times (578/1000) + B \times (114/1000)$$

where P, R, G, and B are grayscale, red, green, and blue pixel values, respectively. Next, we create a 3-mode tensor from all grayscale screenshots and apply `VizFake`.

Likewise, to investigate the effect of vectorizing channels of color screenshots, we created another 3-mode tensor by vectorizing each channel matrix. The detection performance using grayscale and vectorized channel tensors in comparison to our standard 4-mode tensor (from color screenshots) are shown in Fig. 8.6. Given these different input representations, we again evaluate `VizFake` on different rank decompositions. As shown, in contrast to grayscaling, vectorizing the channels slightly improves the F1 scores.

We hypothesize the rationale for similar grayscale performance to the base 4-mode color model is that several important aspects like number of ads, image positions, writing styles (e.g., number of columns, font) are unaffected and still capture the overall look of the webpage (see Fig. 8.5) and thus producing consistent performance. The performance improvement for vectorizaing can be explained as follows: By vectorizing an image, we treat an image as a single observation, or a point in high dimensional pixel space. As a result, we are calculating all possible combinations of pixel statistics, both near and faraway statistics. On the other hand, when we consider an image as a matrix, every different image column is treated as an independent observation, and each pixel only covaries with pixels in the rows and the columns and we are not able to capture all possible pairwise statistics. [165] offers a relevant discussion on vectorization, albeit using subspace arguments rather than latent factor imposed constraints. Overall, the minor changes in the F1 score show

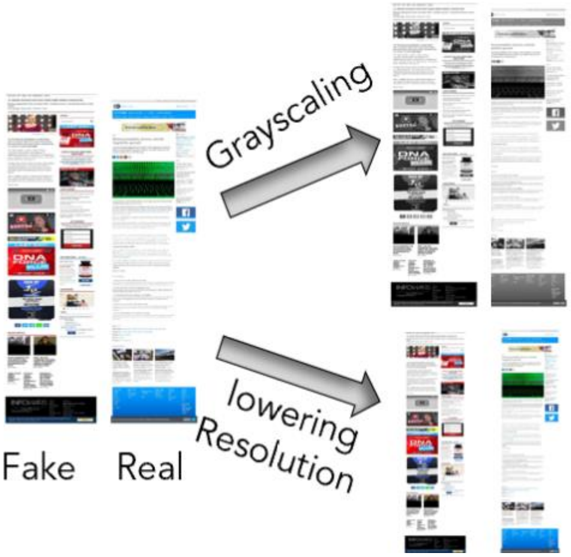


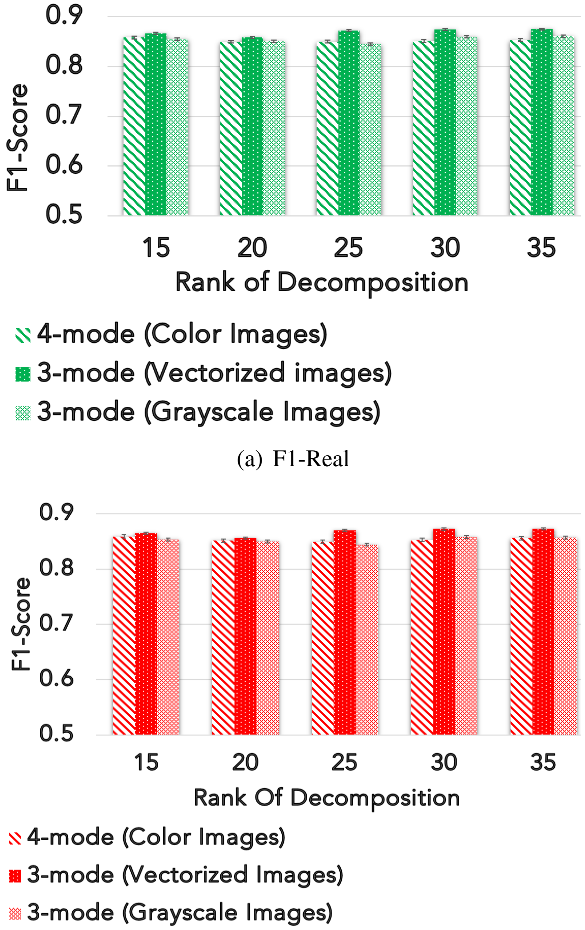
Figure 8.5: An example of grayscaling and changing the resolution on overall look of screenshots.

that `VizFake` is robust against common image transformations, suggesting practical performance across various color configurations and image representation schemes.

8.3.5 Investigating sensitivity to class imbalance

Next, we investigate sensitivity of `VizFake` to class imbalance, as is often the case in practical settings. We create a dataset of size 50k with a 1:2 fake to real article split. We then assume that the known labels are reflective of the class distribution, and use stratified sampling to designate known labels for the belief propagation step. Fig.. 8.7 shows the F1 scores on both balanced and imbalanced data for different percentages of known labels.

As we expect, the F1 score of the fake class drops when we have a scarcity of fake screenshots in the seed label population. Conversely, the F1 score of the real class increases in comparison to a balanced dataset due to more real samples. However, even under the scarcity of fake samples,



(a) F1-Real

(b) F1-Fake

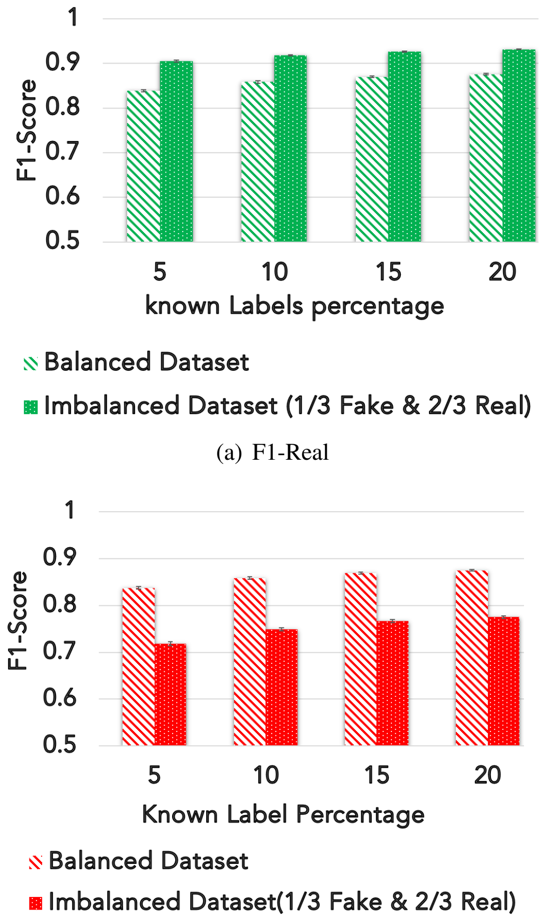
Figure 8.6: F1 score of 4-mode tensor modeling created out of color screenshots against 3-mode tensors out of vectorized and grayscale screenshots for different ranks.

the F1 score using just 5% of the data is around 70% and using 20% the F1 score is almost 78%, suggesting considerably strong results for this challenging task. Overall, changing the proportion of fake to real articles does expectedly impact classification performance. However, performance on the real class is actually not significantly affected.

8.3.6 Investigating importance of website sections

One might ask, “which parts of the screenshots are more informative?” In other words, in which sections are the latent patterns formed? To answer these questions, we propose to cut screenshots into four sections as demonstrated in Fig. 8.8 and use different sections or their combinations while excluding others to create the tensor model (a type of feature ablation study). We propose to create four tensors out of the top, bottom, 2 middle sections, excluding the banner and the concatenation of the top and bottom sections, respectively. For this experiment, we used the 4-mode color tensor and screenshots of size 200×100. Thus, each section is of size 50×100. Fig. 8.9 shows F1 scores of `VizFake` on the aforementioned tensors in comparison to using complete screenshots.

The results show that by cutting the top or bottom sections of the screenshots the F1 score drops by roughly 6% and 8%, respectively. Moreover, if we cut both top and bottom sections the F1 scores decrease significantly by almost 15%. These two sections convey important information including banners, copyright signatures, sign-in forms, headline images, ads, popups, etc. We noted a considerable portion of the informativeness is included outside the banners, as the banners comprise only 10-20% of the top/bottom sections and the F1 scores when only excluding the banners are considerably better than when excluding top and bottom both. The middle sections typically consist of the text of the articles, while other article aspects like pictures, ads, and webpage boilerplate tend to be located at the top/bottom sections. Although the top/bottom sections are more informative, the two middle sections still contain important information such as the number of columns, font style, etc. because the middle sections solely, can still classify screenshots with the F1 score of 67% using just 5% of labels. By capturing all sections, we achieve significantly stronger results i.e.,



(a) F1-Real

(b) F1-Fake

Figure 8.7: F1 score of using VizFake on an imbalanced dataset (The ratio of screenshots published by fake domains to real ones is 1 : 2). On the contrary to fake class, the F1 score of real class increases due to having more samples.

83% F1 using just 5% labels. This experiment suggests that even if the screenshots are corrupted or censored for privacy considerations e.g., excluding headers and other obvious website tells, we are still capable of identifying fake/real domains using as little as 50% of the underlying images.

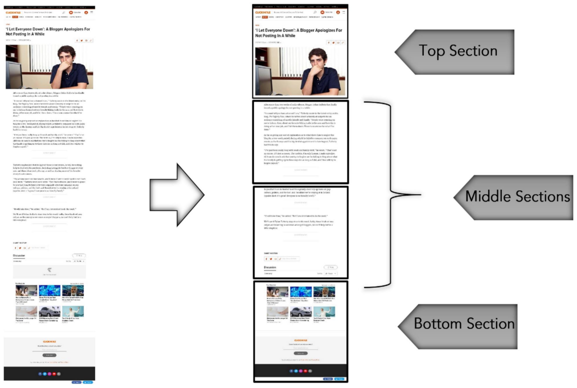


Figure 8.8: Cutting a screenshot into four sections.

	Fake Class					
	VizFake			VGG16 deep network		
%labels	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
5	**0.852±0.002**	0.860±0.005	0.844±0.004	0.799±0.008	0.823±0.027	0.779±0.039
10	**0.871±0.001**	0.880±0.003	0.863±0.005	0.816±0.003	0.842±0.014	0.793±0.018
15	**0.881±0.001**	0.890±0.002	0.873±0.003	0.837±0.001	0.883±0.009	0.795±0.009
20	**0.888±0.001**	0.896±0.002	0.880±0.003	0.849±0.009	0.884±0.023	0.818±0.034

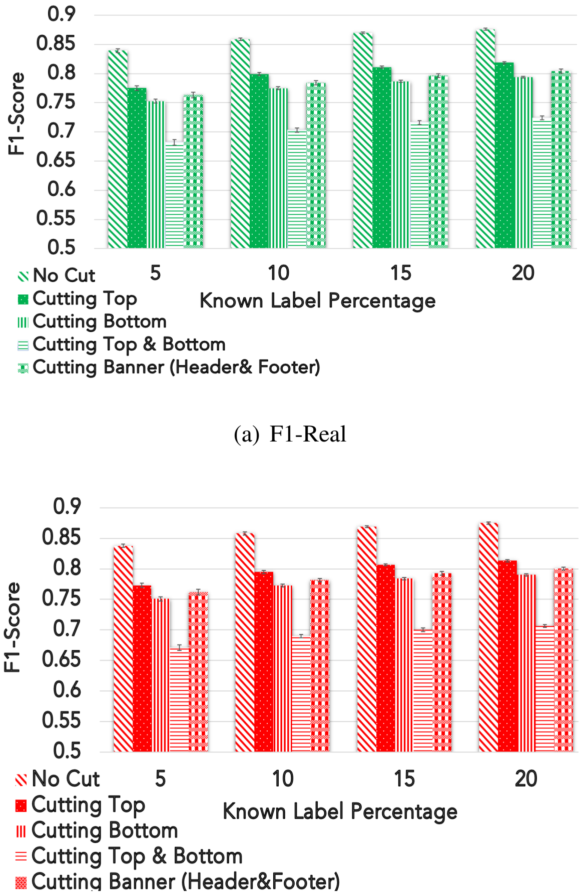
Table 8.1: VizFake outperforms VGG16 when classifying fake class e.g., F1 score (> 0.85) with only 5% of labels.

	Real Class					
	VizFake			VGG16 deep network		
%labels	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
5	**0.854±0.003**	0.847±0.003	0.862±0.006	0.809±0.007	0.790±0.021	0.830±0.039
10	**0.874±0.001**	0.865±0.004	0.882±0.004	0.827±0.003	0.804±0.010	0.851±0.019
15	**0.884±0.001**	0.876±0.002	0.892±0.003	0.852±0.002	0.813±0.005	0.894±0.010
20	**0.890±0.001**	0.882±0.003	0.898±0.003	0.860±0.005	0.831±0.021	0.892±0.029

Table 8.2: VizFake outperforms VGG16 when classifying real class e.g., F1 score (> 0.85) with only 5% of labels.

8.3.7 Comparing against deep learning models

A very reasonable first attempt at classification of screenshots, given their wide success in many computer vision tasks, is the use of Deep Convolutional Neural Networks (CNNs). To understand whether or not CNNs are able to capture hidden features that VizFake scheme cannot extract, we



(a) F1-Real

(b) F1-Fake

Figure 8.9: Changes in the F1 score when cutting different sections of the screenshots. In contrast to 2 middle sections, Cutting the top and bottom sections causes a considerable decrease in F1 scores. It seems that style defining style-defining events of the webpages are mostly focused in the top and bottom sections of the webpages.

also try CNNs for classification of screenshots. From a pragmatic point of view, we compare i) the classification results each method achieves, and ii) the runtime required to train the model in each case. In what follows, we discuss the implementation details.

VizFake configuration

We showed that the vectorized tensor outperforms 3-mode grayscale and 4-mode color tensors. So, we choose the 3-mode tensor as tensor model. We use the balanced dataset comprising 50k screenshots with resolution of 200 × 100 and finally we set the rank to 35 based on what is in Fig.. 8.6.

Deep learning configuration

Although our modest-sized dataset has considerable examples per class (25k), it is not of the required scale for current deep models; thus, we resort to deep transfer learning [117].

We choose VGG16 [143] pretrained on ImageNet [35] as our base convolutional network and modify the final fully connected layers to suite our binary classification task.We also tried some other models, they all basically perform similarly. The performance we got was indicative and also was on par with other models. So, we just report the results for VGG16 which is robust enough and the hyper-parameter optimization process is feasible in terms of time and available resources. The network is subsequently fine-tuned on screenshot images. Due to label scarcity, we want to see if the deep network performs as well as `VizFake` when there is a limited amount of labels. Thus, we experiment by fine-tuning on the same label percentage we use for `VizFake`. The remaining images are used for validation and testing.

We use the Adam optimizer [86] and search between 0.0001 and 0.01 for the initial learning rate. We apply sigmoid activation in the output layer of the network and the binary cross-entropy as the loss function. The batch sizes we experiment with ranged from 32 to 512 and we finally fixed the batch size for all experiments to 512. Batch size significantly impacts learning as a large enough batch size provides a stable estimate of the gradient for the whole dataset. [68, 144]. The convergence takes approximately 50 epochs. We note that the effort required to fine-tune a deep network for this task was tedious and included manual trial-and-error, while VizFake requires the determination of just 2 parameters, both of which produce stable performance across a reasonable range.

Comparing classification performance

Next, we compare the classification performance of VizFake against the CNN method we explained above in terms of precision, recall, and F1 score. Tables 8.1 and 8.2 show the achieved results of these metrics for VizFake and CNN model. As demonstrated, VizFake outperforms CNN especially given less labeled data. For instance, the F1 scores of VizFake for the fake class when we use only 5%-10% of the labels are 85%-87%, respectively which is 5-6% higher than the 80%-81% F1 scores from the CNN model. Thus, VizFake achieves better performance while avoiding considerable time in finding optimal hyperparameters required for tuning VGG16.

Comparing the time efficiency

We evaluate time efficiency by measuring the runtime each method requires to achieve the best results. We experiment on two settings:

The first one uses a GPU since CNN training is an intensive and time-consuming phase which typically requires performant hardware. Although using a GPU-based framework is not necessary for VizFake, we re-implemented VizFake on the same setting we use for the deep learning model to leverage the same scheme, i.e. Python using TensorLy library [88] with TensorFlow backend. Thus, we avoid influence from factors like programming language, hardware configuration, etc.

The second configuration uses a CPU and is the one we used in prior experiments and discussed in the Implementation section. Since we are not able to train the CNN model with this configuration due to excessively long runtime, we only report them for VizFake.

For both experiments, we measure the runtime of bottlenecks, i.e., decomposition of VizFake and training phase of the deep learning method. Other steps such as: K-NN graph construction, belief propagation, and test phase for CNN method are relatively fast and have negligible runtimes (e.g. construction and propagation for the K-NN graph with 50K screenshots take just 3-4 seconds). Due to our limited GPU memory, we experiment using a 5% fraction of the dataset for the GPU configuration. By doing so, we also reduce the I/O overhead that may be counted as execution time when we have to read the dataset in bashes. However, we use 100% of the dataset for the CPU setting. The technical aspects of each configuration are as follows:

Configuration 1:

- Keras API for Tensorflow in Python to train the deep network and Python using Tensorly with TensorFlow backend for VizFake.
- 2 Nvidia Titan Xp GPUs (12 GB)
- Training: 5% (2500 screenshots of size 200×100), validation: 4% (2000 screenshots)
- Decomposition: 5% (2500 screenshots of size 200×100)

Configuration 2:

- Matlab Tensor Toolbox 2.6
- CPU: Intel(R) Core(TM) i5-8600K CPU @ 3.60GHz
- Decomposition: 100% (50K screenshots)

The average number of iterations, time per iteration, and average total time for 10 runs

Resolution	Avg.# Iter.	Avg. Time/Iter.	Avg. Time
200 × 100	7.64	23.76s	181.55s
300 × 100	7.88	35.52s	279.95s
400 × 100	7.72	47.82s	369.22s

Table 8.3: Execution time (Sec.) of VizFake for different resolutions on configuration 2

Method	Avg. # Iter.	Avg. time/Iter.	Avg. Time
VizFake	**7.08**	**1.05s**	**7.64s**
CNN	50	33.08s	1654s

Table 8.4: Execution times (Sec.) of VizFake and CNN deep learning model on configuration 1.

of both methods on Configuration 1 and the same metrics for `VizFake` on Configuration 2 are reported in Tables 8.4 and 8.3, respectively.

Based on execution times demonstrated in Table 8.4, the tensor-based method is roughly 216 and 31.5 times faster than the deep learning method in terms of average time and average time per iteration, respectively. Moreover, the iterations required for `VizFake` is almost 7 times less than the epochs required for the CNN method. Note that these results are very conservative estimates since we do not consider time spent tuning CNN hyperparameters in this evaluation. Table 8.3 shows the execution time for `VizFake` on Configuration 2. Decomposing a tensor of 50k color screenshots using CPU is roughly 3 Mins for screenshots of size 200 × 100, increasing to 6 Mins for larger tensors.

Overall, the results suggest that `VizFake` is 2 orders of magnitude faster than the state-of-the-art deep transfer learning method for the application at hand, and generally more "user-friendly" for real-world deployment.

	Fake Class				
%labels	TF-IDF/SVM	Doc2Vec/SVM	GloVe/LSTM	FastText	VizFake
5	0.812±0.005	0.511±0.000	0.651±0.019	0.717±0.010	**0.844±0.004**
10	0.828±0.001	0.530±0.004	0.672±0.024	0.748±0.007	**0.863±0.005**
15	0.836±0.002	0.540±0.004	0.699±0.020	0.757±0.006	**0.873±0.003**
20	0.841±0.001	0.546±0.002	0.718±0.002	0.758±0.004	**0.880±0.003**

Table 8.5: The F1 score of VizFake for fake class, outperforms the F1 score of state of the art text-based approaches.

	Real Class				
%labels	TF-IDF/SVM	Doc2Vec/SVM	GloVe/LSTM	FastText	VizFake
5	0.814±0.004	0.511±0.000	0.650± 0.028	0.650± 0.030	**0.862±0.006**
10	0.829±0.005	0.520±0.001	0.680±0.005	0.707± 0.016	**0.882±0.004**
15	0.836±0.003	0.526±0.002	0.698±0.013	0.712±0.010	**0.892±0.003**
20	0.842±0.001	0.534±0.006	0.712±0.009	0.728±0.009	**0.898±0.003**

Table 8.6: The F1 score of VizFake for real class, outperforms the F1 score of state of the art text-based approaches.

8.3.8 Comparing against text-based methods

Even though the main goal of this work is to explore whether or not we can leverage the overall look of the serving webpage to discriminate misinformation, we compare the classification performance of VizFake with some well-known text-based approaches to investigate how successful is the proposed approach in comparison to these widely used methods. We compare against:

- **tf-idf** term frequency–inverse document frequency method is one of the widely used methods for document classification. tf-idf models the importance of words in documents. We create a tf-idf model out of screenshots text and then we leverage SVM for classification.

- **Doc2Vec/SVM** a shallow 2-layers neural network proposed by Google [93]. Doc2Vec is an extension to word2vec and generate vectors for documents. Again, we use SVM classifier.[4]

- **fastText** a proposed NLP library by Facebook Research. fastText learns the word representations which can be used for text classification. It is shown that the accuracy of

[4] https://github.com/seyedsaeidmasoumzadeh/Binary-Text-Classification-Doc2vec-SVM

fastText is comparable to deep learning models but is considerably faster than deep competitors[5] [18].

- **GloVe/LSTM** a linear vector representation of the words using an aggregated global word-word co-occurrence. We create a dictionary of unique words and leverage Glove to map indices of words into a pre-trained word embedding [**?**]. Finally, we leverage a LSTM classifier[6] pre-trained on IMDB and fine-tune it on our dataset. We examined embedding length in range 50-300 and finally set it to 300. The tuned batch size and hidden size are 256, 64 respectively.

The experimental results of the aforementioned methods are given in Table. 8.5 and Table. 8.6. As demonstrated, the classification performance of VizFake reported in these tables, outperforms the performance of the shallow network approaches i.e., Doc2Vec/SVM and fastText as well as the deep network approach i.e., GloVe/LSTM which shows the capability of VizFake in comparison to neural network methods in settings that there is a scarcity of labels. The tf-idf representation along with SVM classifier leads to classification performance close to the proposed visual approach which illustrates that visual information of the publishers is as discriminative as the best text-based approaches.

8.3.9 Comparing against website structure features

A question that may come to mind is ”why not using website features instead of screenshots?” To address this question, we repeat the proposed pipeline i.e., decomposition, K-NN graph, and belief propagation this time using HTML tags crawled from the serving webpages. To this end, we create

[5] https://github.com/facebookresearch/fastText

[6] https://github.com/prakashpandey9/Text-Classification-Pytorch

an article/tags matrix then we decompose this matrix using Singular Value Decomposition ($\mathbf{X} \simeq \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$) and leverage matrix $\mathbf{U}$ which corresponds to articles pattern to create a K-NN graph and propagate the labels using FaBP. The result of this experiment is given in Table.8.7. As illustrated in Table.8.7, using HTML tags is highly predictive which is another justification for using the overall look of the webpages. The question raises now is that "Why not just using website features for capturing the overall look, especially when the classification performance is better?" Here is some reasons for using screenshots instead of website features:

- HTML source of the domain is not always available or even if we gain access to the source, the page may be generated dynamically and as a result, the features that can be informative are probably non-accessible scripted content. This is why the HTML source of our dataset provided us with features mainly related to the high-level structure of the domain shared between different screenshots.
- HTML feature extraction requires tedious web crawling and data cleaning processes and is difficult to separate useful features from useless ones. Taking screenshots is easy and can be done fast and online needless to extra resources or expert knowledge for web crawling.
- Even if we have access to the HTML source and be able to separate useful features in an efficient way, these features do not give us any information about the content of the web events such as images, videos, ads, etc. If we are to conduct article-level labeling or even section level labeling (usually just some part of an article is misinformative) we will miss a lot of useful information when we use HTML features while screenshots capture such details.

Given the reasons above, the screenshots are not only as informative as textual content, but also are

%labels	5	10	15
Fake	0.977±0.0004	0.983±0.0002	0.985±0.0002
Real	0.977±0.0004	0.983±0.0003	0.985±0.0002

Table 8.7: Performing proposed pipeline on HTML-Tags of articles. The result justifies that HTMLs only contain domain features which is shared between all articles of that domain.

preferred over time-consuming and often less informative HTML features.

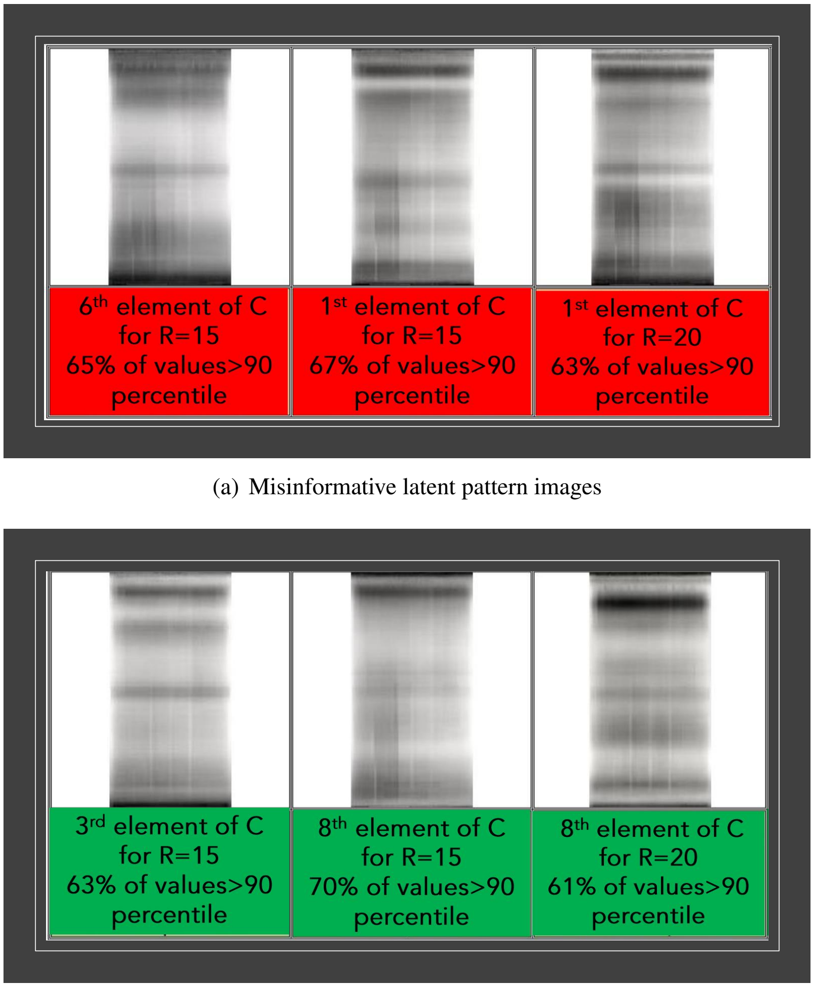
8.3.10 Exploratory analysis

The tensor representation of `VizFake` is not only highly predictive in semi-supervised settings, but also lends itself to exploratory analysis, due to the ease of interpretability of the decomposition factors. In this section, we leverage those factors in order to cluster domains into coherent categories (misinformative or not), in an unsupervised fashion. Each column of the screenshot embedding **C** indicates the membership of each screenshot to a cluster, defined by each of the rank-one components (for details on how to generally interpret CP factors as clustering, see [118]). Each one of the clusters has a representative latent image, which captures the overall intensity in different parts of the image indicating regions of interest that are participating in generating that cluster. To obtain this image, we compute the outer product of column vectors of matrices corresponding to pixels and channels i.e., **A** and **B** for the vectorized tensor and scale it to range 0-255 which provides us with R latent images. We then annotate the images based on the ground truth only to verify that the coherent clusters correspond to fake or real examples. We investigate the interpretability of these latent images by taking the 90th percentile majority vote from the labels of articles with high score in that latent factor. The details of clustering approach is demonstrated in Algorithm. 8.

Examples of latent images corresponding to misinformative and real classes are illustrated in Fig. 8.10. The darker a location of an image, the higher degree of “activity” it exhibits with respect to that latent pattern. We may view those latent images as “masks” that identify locations of interest within the screenshots in the original pixel space. In Fig. 8.10, we observe that latent images corresponding to real clusters appear to have lighter pixels, indicating little “activity” in those locations. For example, the two latent images resulted from rank 15 decomposition are lighter than latent images for the fake class, also the same holds for rank 20. Moreover, as illustrated in Fig. 8.10, darker pixels are more concentrated at the top and the bottom parts of the images which are wider for misinformative patterns and corroborate our assumption about having more objects, such as ads and pop-ups, in fake news websites. As mentioned, such objects are more prevalent at the top and the bottom of the websites which matches our observation here and the cutting observation we discussed earlier. As shown in Fig. 8.9, cutting the bottom and top sections lead to more significant changes in performance than cutting just the banner which also confirms our assumption about informativeness of these sections. This experiment not only provides us with a clustering approach which is obtained without labels and correlates with existing ground truth but also enables us to define filters for misinformation pattern recognition tasks in form of binary masks, that identify locations of interest within a screenshot, which can further focus our analysis.

8.3.11 Limitations of the work

As discussed earlier, collecting annotation for misinformation detection is a complicated and time-consuming task and as we increase the granularity of the labels from domain level to articles level and even article sections it becomes harder and harder. Moreover, the majority of available ground truth resources like “BS Detector” or ”NewsGuard” provide labels pertain to domains rather than articles. Despite this disparity, it is shown in several works [66, 195] that the weakly-supervised



(a) Misinformative latent pattern images

(b) Real latent pattern images

Figure 8.10: Examples of the cumulative structures of all articles corresponding to factors with the majority of misinformative/real labels. Contrary to the real class, images of misinformative class have darker pixels i.e., the dark portion of the image is wider.

task of using labels pertaining to domains, and subsequently testing on labels pertaining to articles, yields negligible accuracy loss due to the strong correlation between the two targets. However, as mentioned in the webpage structure section, there are useful article-level information like web events content that can be taken advantage of when we have grainier labels and capturing them causes a drop in performance because they may be considered as noise when working with domain

Algorithm 8 Exploratory analysis

```
**Input:**A, B and C Factor Matrices
**Output:** Latent pattern images
\\ scale the result to values between 0-255
$min = 0; max = 255$
$\mathbf{a_{ij}} = \frac{(\mathbf{a_{ij}} - min(\mathbf{a_{ij}}) \times (max - min)}{(max(\mathbf{a_{ij}}) - min(\mathbf{aij}))} + min$
$\mathbf{b_{ij}} = \frac{(\mathbf{b_{ij}} - min(\mathbf{b_{ij}}) \times (max - min)}{(max(\mathbf{b_{ij}}) - min(\mathbf{bij}))} + min$
for $\underline{i = 1 \cdots R}$ do
    $\mathbf{X}^{i}_{cumulative} \simeq \mathbf{a}_i \circ \mathbf{b}_i$
    $top_n{}^{i}$ = top $(100 - \alpha)$ percentile values $c_i$
    $\mathbf{X}^{i}_{cumulative}$ =Label-majority-Vote($top_n{}^{i}$)
end
```

level labels. We defer the study of obtaining and using finer-grained labels for future work.

8.4 Related work

8.4.1 Visual-based misinformation detection

The majority of work proposed so far focus on content-based or social-based information. However, there are few studies on visual information of articles. For instance, in [59,124] the authors consider user image as a feature to investigate the credibility of the tweets. In another work, Jin et al. [74] define clarity, coherence, similarity distribution, diversity, and visual clustering scores to verify microblogs news, based on the distribution, coherency, similarity, and diversity of images within microblog posts. In [150] authors find outdated images for the detection of unmatched text and pictures of rumors. Gupta et al. in [57] classify fake images on Twitter using a characterization analysis to understand the temporal, social reputation of images. On the contrary, we do not focus on the user aspect, i.e., profile image or metadata within a post e.g., image, video, etc. Thus, no matter if there is any images or not, VizFake captures the overall look of the article.

8.5 Conclusions

In this chapter, we leverage a very important yet neglected feature for detecting misinformation, i.e., the overall look of serving domain. We propose a tensor-based model and semi-supervised classification pipeline i.e., `VizFake` which outperforms text-based methods and state-of-the-art deep learning models and is over 200 times faster, while also being easier to fine-tune and more practical. Moreover, `VizFake` is resistant to some common image transformations like grayscaling and changing the resolution, as well as partial corruptions of the image. Furthermore, `VizFake` has exploratory capabilities i.e., it can be used for unsupervised soft-clustering of the articles. `VizFake` achieves F1 score of roughly 85% using only 5% of labels for both real and fake classes on a balanced dataset and an F1 score of roughly 95% for real class and 78% for the fake class using only 20% of ground truth on a highly imbalanced dataset.

9

Deepfake Detection with Multilinear Projection

Generative neural network architectures such as GANs, may be used to generate synthetic instances to compensate for the lack of real data. However, they may be employed to create media that may cause social, political or economical upheaval. One emerging media is ”Deepfake”. Techniques that can discriminate between such media is indispensable. In this chapter, we propose a modified multilinear (tensor) method, a combination of linear and multilinear regressions for representing fake and real data. We test our approach by representing Deepfakes with our modified multilinear (tensor) method and perform SVM classification. Our proposed approach achieves promising results while using few frames per video.

9.1 Introduction

Recent advances in Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs) embedded in applications like Zao[1], DeepFakes web β[2], Face Swap by Microsoft[3], DeepFaceLab[4] etc. have led to a broad usage of AI-synthesized media a.k.a. ”Deepfake” [5]. Other automated manipulation techniques are Face2Face, FaceSwap, NeuralTextures, and FaceShifter [127].

Due to the potential misuse of Deepfakes e.g., fake pornography, fake news, and financial or political fraud, they have become a major public concern. Thus, different techniques have been introduced to discriminate Deepfakes from pristine videos. Interested reader is refered to [127] for more details on the aforementioned methods.

Prior Deepfakes detection can be categorized as [152] approaches that classify based on (a) physical or physiological causal factors which are not well presented in Deepfakes e.g., eye blinking [98] and heart rate [67], or (b) artifacts in imaging factors e.g., relative head pose to the camera position [188], and (c) data-driven techniques that do not leverage specific cues and directly train a deep learning model on a large set of real and Deepfake videos [5, 29].

[1] https://www.zaoapp.net/

[2] https://deepfakesweb.com/

[3] https://www.microsoft.com/en-us/garage/profiles/face-swap/

[4] https://awesomeopensource.com/project/iperov/DeepFaceLab

[5] The term Deepfakes has been widely used for deep learning generated media, but it is also the name of a specific manipulation technique in which face of one person is replaced by another one. To distinguish these, we denote said method by DeepFakes in the entire paper.

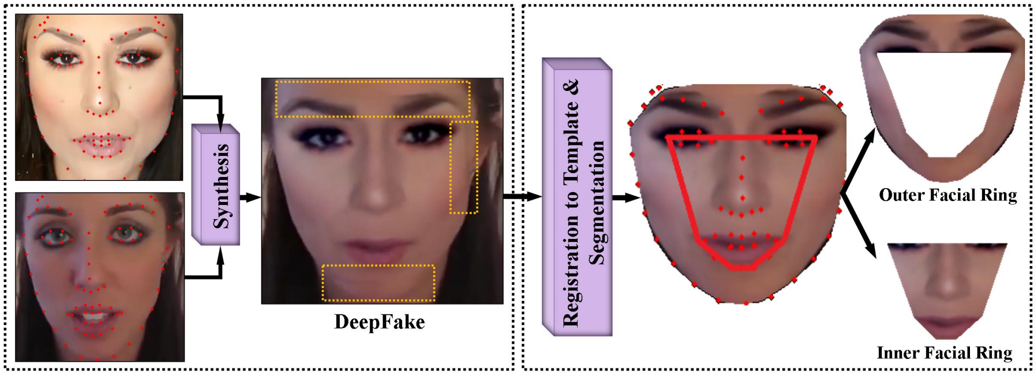


Figure 9.1: Deepfake technique replaces a person's appearance in an existing image or video with someone else's appearance [127]. This process introduces artifacts specially around the cropping boundaries estimated by facial landmarks. We propose segmenting the output face into inner and outer facial rings. The artifacts are mainly concentrated in outer facial ring.

We hypothesize that Deepfakes contain artifacts localized either in transition areas between facial images, or contain discrepancies in the overall facial appearance. We concentrate our analysis on the transition areas of the face henceforth referred to as outer facial ring, Fig. 9.1. We segment the outer ring from a facial image that has been registered to a template based on facial landmarks detected by a pretrained model [83]. The outer ring is analyzed with a modified face recognition tensor model [158, 160] that computes real and fake data representations.

We employ a multilinear a.k.a. tensor framework which decomposes basis components of outer facial rings into real and fake class representations. Later on we leverage the derived representation of classes to classify the test frames using a linear SVM. Summarily, our major contributions are as follows:

- **Segmenting face into regions of interest**: We propose Segmenting face into facial parts and leverage parts with high concentration of artifacts to distinguish Deepfakes.
- **Proposing a multilinear representation of Deepfakes for classification:** we employ a multilinear approach to represent Deepfake and real class information and then leverage them for

classification.

9.2 Proposed method

A DeepFake is a synthesizing product of two real faces. More precisely, in DeepFake generation process, face of a real person a.k.a. target is synthesized by another face a.k.a., source. This process, usually introduces some artifacts, specially around the cropping edges of source face including eyes and eyebrows Fig. 9.1. Due to the fact that a Deepfake face is a mixture of source and target faces, Sometimes it is not distinguishable from the source and this similarity results in misclassification of the video. In this work, we propose to segment faces into parts henceforth referred to as facial inner and outer rings Fig. 9.1. We define the outer ring as a facial part that comprises the blending boundaries that are mostly the non-facial pixels. We leverage this remaining region i.e., outer ring which has the highest concentration of introduced artifacts as a cue for Deepfakes detection. An example of this process is demonstrated in Fig. 9.1. This cue is very promising specially when the manipulation masks are not available.

In what follows, we discuss our proposed multilinear pipeline for detecting the Deepfakes.

9.2.1 Step 1: Vectorizing video frames

Vasilescu [163, Appndix A] argues that in most cases, it is preferable to vectorize an image and treat it as a single observation rather than a collection of independent column/row observations. By vectorizing an image, we treat an image as a point in high dimensional pixel space and calculate all possible combinations of pixel statistics, both near and faraway statistics. On the other hand, when we consider an image as a matrix, every image column (row) is treated as an independent observation, and column (row) covariances are computed. each pixel only covaries with pixels of the same row and the same column. and we are not able to capture all possible pairwise statistics [163]. Having this in mind, we also follow the same strategy and vectorize the frames and create a vector for each one of the video frames in the dataset.

9.2.2 Step 2: Finding eigenfaces of each class

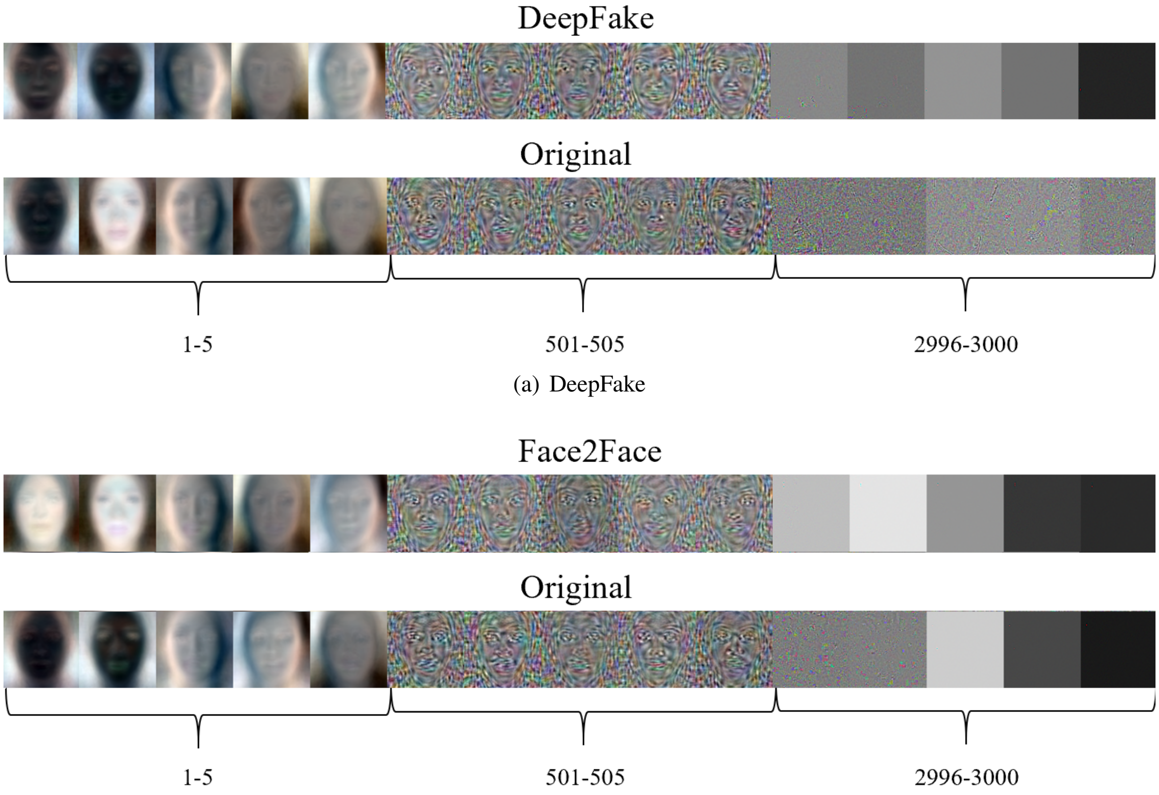
Eigenfaces are eigenvectors when the images are human face. The eigenfaces are derived from the covariance matrix of the pixel distribution over the high dimensional face space. The eigenfaces represent a basis set of all faces used to construct the covariance matrix. So far, the eigenfaces have been successfully leveraged for many facial image related tasks [153]. Leveraging eigenfaces allows for dimensionality reduction such that a smaller set of basis vectors represent the original training faces. Classification could be achieved by comparing how different faces are represented by the basis set of the corresponding class. Examples of eigenfaces are demonstrated in Fig. 9.2

Based on principle component terminology, the eigenfaces are equal to basis vectors of PCA decomposition. Therefore, by staking the vectorized frames of each class, we create two separate matrices and decompose them using SVD to capture the eigenfaces of the corresponding class as follows:

$$\mathbf{D}_{\text{real}} = \mathbf{U}_{\text{real}}\boldsymbol{\Sigma}_{\text{real}}\mathbf{V}^{\text{T}}_{\text{real}} = \mathbf{B}_{\text{real}}\mathbf{V}^{\text{T}}_{\text{real}} \tag{9.1}$$

$$\mathbf{D}_{\text{fake}} = \mathbf{U}_{\text{fake}}\boldsymbol{\Sigma}_{\text{fake}}\mathbf{V}^{\text{T}}_{\text{fake}} = \mathbf{B}_{\text{fake}}\mathbf{V}^{\text{T}}_{\text{fake}} \tag{9.2}$$

Where $\mathbf{B}_{\text{real}}$ and $\mathbf{B}_{\text{fake}}$ are basis matrices and $\mathbf{V}_{\text{real}}$ and $\mathbf{V}_{\text{fake}}$ are the normalized coefficient matrices of



(a) DeepFake

(b) Face2Face

Figure 9.2: Top 3000 eigenfaces corresponding to original and manipulated videos of `DeepFakes` and `Face2Face` datasets.

the corresponding classes.

9.2.3 Step 3: Leveraging tensor framework to decompose eigenfaces into underlying factors

As seen in previously mentioned tensor is an effective framework for decomposing a set of observation into underlying factors. After reducing the dimentionality of observations using eigenface representation of the classes, we propose leveraging a three-mode tensor where the first mode i.e., measurement mode represents the pixels of an eigenface, the second mode corresponds to the eigenfaces and the third mode is the class mode i.e., DeepFake vs. real. We propose using an M-mode

SVD which as we discussed earlier decomposes a tensor into M orthonormal matrices ($M = 3$), and a core tensor which governs the interaction between these spaces. Since the first mode is the measurement mode, We only calculate the M-mode SVD of the tensor by flattening the second and the third modes as follows:

$$\begin{aligned} \mathcal{D} &\simeq \mathcal{Z} \times_1 \mathbf{U}_{\text{p}} \times_2 \mathbf{U}_{\text{f}} \times_3 \mathbf{U}_{\text{c}} && (9.3) \\ &= \mathcal{T} \times_2 \mathbf{U}_{\text{f}} \times_3 \mathbf{U}_{\text{c}} && (9.4) \end{aligned}$$

where the $\mathbf{U}_{\text{c}}$ comprises underlying vector representation of original and fake classes. Moreover, the core tensor $\mathcal{T}$ is the signature of this dataset and shows interactions of orthonormal subspaces. Later on, we leverage this signature to project the test frames into the subspaces we derive here.

9.2.4 Step 4: Embedding the class representations in a higher three dimensional space

Applying M-mode SVD results in a mode matrix $\mathbf{U}_{\text{c}} \in \mathbb{R}^{2\times2}$ that spans the class representations. We embed the vector class representations into a higher dimensional space to increase the class separability of the test data. We embed the row vectors of $\mathbf{U}_{\text{c}}$ into $\mathbb{R}^3$,

setting the third coordinate of the real and fake class to +1 and −1 respectively, and normalizing the vector length to 1.

9.2.5 Step 5: Multilinear projection of an incoming frame into the orthonormal vector spaces

As mentioned above, the core tensor of each decomposition is the signature of the decomposed space which governs the interaction of constituent factors. we leverage the core tensor and perform a multilinear projection of the incoming frame into the subspaces we derived in the previous step. Let say we have the vectorized frame **d**. If **d** is supposed to be in the same subspaces we derived, then

$$\mathbf{d} = \mathcal{T} \times_2 \mathbf{f}^{\mathrm{T}} \times_3 \mathbf{c}^{\mathrm{T}} \tag{9.5}$$

where the vectors **f** and **c** are the coefficient vector representations of a video frame **d** in the orthonormal subspaces that are governed by the extended core tensor $\mathcal{T}$. The goal is to find out weather the class coefficient vector **c** is more similar to the vector representation of real class or Deepfake class. To this end, we estimate **c** representation vector by employing the multilinear projection algorithm [157, 171] that decomposes a vectorized observation, **d** into a set of latent vector representation, $\mathbf{r}_{\mathrm{n}}$ that corresponds to the constituent factors of data formation. The basic multilinear projection is the M-mode SVD/CP decomposition of $\mathcal{T}^{\dagger_1} \times_1 \mathbf{d}^{\mathrm{T}}$ which can be expressed mathematically as

$$\underbrace{M\text{-mode SVD/CP}\left(\mathcal{T}^{\dagger_1} \times_1 \mathbf{d}^{\mathrm{T}}\right)}_{\text{Multilinear Projection}} \simeq \mathbf{r}_{\mathrm{f}} \circ \mathbf{r}_{\mathrm{c}} \;\Rightarrow\; \mathbf{d} \simeq \left(\mathcal{T} \times_2 \mathbf{r}_{\mathrm{f}}^{\mathrm{T}} \times_3 \mathbf{r}_{\mathrm{c}}^{\mathrm{T}}\right)$$

where $\mathcal{T}^{\dagger_1}$ is mode-1 pseudo-inverse of $\mathcal{T}$ that in matrix notation is expressed as $\mathbf{T}^{\dagger}_{[1]}$, and $\mathbf{r}_{\mathrm{c}}$, $\mathbf{r}_{\mathrm{f}}$ are estimates of vectors $\mathbf{c}$ and $\mathbf{f}$ from eq.(9.5), respectively.

9.2.6 Step 6: Classifying an incoming frame

Up to this step, we have the vector representation of each classes in addition to class coefficients of the incoming frame. We use a linear Support Vector Machine (SVM) and estimate the decision boundaries using validation frames and then leverage the defined boundaries for classification of test frames. An overview of the proposed approach is demonstrated in Algorithm 9.

9.2.7 Dimensionality reduction in step 3

As mentioned earlier, factor matrix $\mathbf{U}_f$ comprises underlying structures of basis vector continent. Despite the fact that we construct our predictive model by approximating discriminating regions, still there are many shared components which getting rid of them make the model more distinguishable. Since we are interested in noisy regions i.e., artifacts, we propose truncating components of the core tensor $\mathcal{T}$ which correspond to top values of $\mathbf{U}_f$ and keeping lower value components as representatives of noisy parts. In the next section, we will show how this truncation boosts the classification performance of the proposed framework. An example of truncating components corresponding to the second mode of a 3-mode tensor is depicted in Fig. 9.3.

Algorithm 9 DeepFake Detection Algorithm

```
Input : $\mathbf{D}_{\text{real}}, \mathbf{D}_{\text{fake}}$ were centered by subtracting the mean of the real training data,

  1. Preprocessing and data tensor organization:
     $[\mathbf{U}_{\text{real}}, \mathbf{S}_{\text{real}}, \mathbf{V}_{\text{real}}] \Leftarrow \text{svd}(\mathbf{D}_{\text{real}})$
     $[\mathbf{U}_{\text{fake}}, \mathbf{S}_{\text{fake}}, \mathbf{V}_{\text{fake}}] \Leftarrow \text{svd}(\mathbf{D}_{\text{fake}})$
     $\mathcal{D}(:,:,\mathbf{1}) = [\mathbf{U}_{\text{real}}\mathbf{S}_{\text{real}}]$
     $\mathcal{D}(:,:,\mathbf{2}) = [\mathbf{U}_{\text{fake}}\mathbf{S}_{\text{fake}}]$

  2. Training data decomposition:
     $\mathcal{T} \times_2 \mathbf{U}_{\text{f}} \times_3 \mathbf{U}_{\text{c}} \Leftarrow M\text{-mode SVD}(\mathcal{D})$

  3. Embed the class representations in the higher three dimensional space and set the third co-
     ordinate of the real and fake class to +1 and $-1$ respectively. Hence, $\mathbf{U}_{\text{c}} \in \mathbb{R}^{2\times 2}$ now has
     dimensionality $\mathbb{R}^{2\times 3}$. Normalize the rows of $\mathbf{U}_{\text{c}}$ to have length 1.

  4. Computer the extended core
     $$\mathcal{T} := \mathcal{D} \times_2 \mathbf{U}_{\text{f}}^{\text{T}} \times_3 \mathbf{U}_{\text{c}}^{\dagger} \qquad (9.6)$$

  5. Centering: validation and test data is centered by subtracting the mean of the real training
     data.

  6. Test data decomposition of a centered $\mathbf{d}_{\text{test}}$ :
     $$\mathbf{d}_{\text{test}} \simeq \mathcal{T} \times_2 \mathbf{r}_{\text{f}}^{\text{T}} \times_3 \mathbf{r}_{\text{c}}^{\text{T}} \Leftarrow \text{Multilinear Projection}(\mathcal{T}, \mathbf{d}_{\text{test}})$$

  7. Finding linear SVM decision boundaries using validation set

  8. classifying all $\mathbf{d}_{\text{test}} \in$ test set
```

9.3 Experimental evaluation

In this section, we first introduce the dataset and benchmark on this dataset and then we discuss the implementation details and the experimental evaluation.

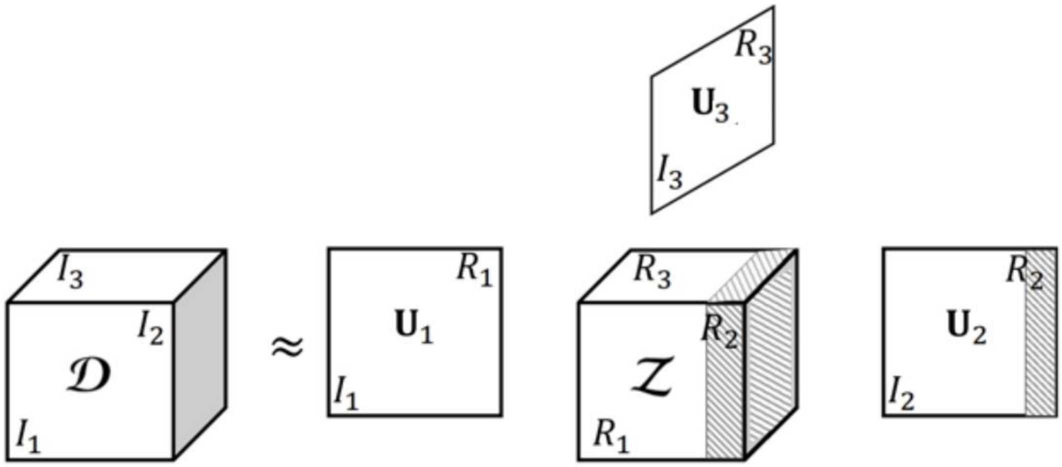


Figure 9.3: M-mode SVD decomposition of a 3-mode tensor. Some of the singular values corresponding to the components of the second factor matrix i.e., second mode of tensor $\mathcal{D}$ are truncated.

9.3.1 Dataset description

One of the most popular and widely used databases for image or video forgeries detection is FaceForensics++[6] which first was introduced in 2018 [126]. FaceForensics++ comprises more than 500,000 frames from 1000 youtube videos that contain mostly frontal faces [127]. This dataset also includes 1000 videos which are the manipulated version of the original onesand have been manipulated by four automated face manipulation methods: Deepfakes, Face2Face, FaceSwap and NeuralTextures. All original and manipulated videos have constant frame rate of 30 fps and have been compressed lossless with H.264. Moreover, the videos are split up into train set of size 720, validation set of size 140 and test of 140 videos. binary classification scenario on this dataset. The state-of-the-art benchmark on FaceForensics++ is available in GitHub[7]. In this work, we experiment

[6] https://github.com/ondyari/FaceForensics

[7] http://kaldir.vc.in.tum.de/faceforensics-benchmark/

on images manipulated by DeepFake technique.

9.3.2 Implementation

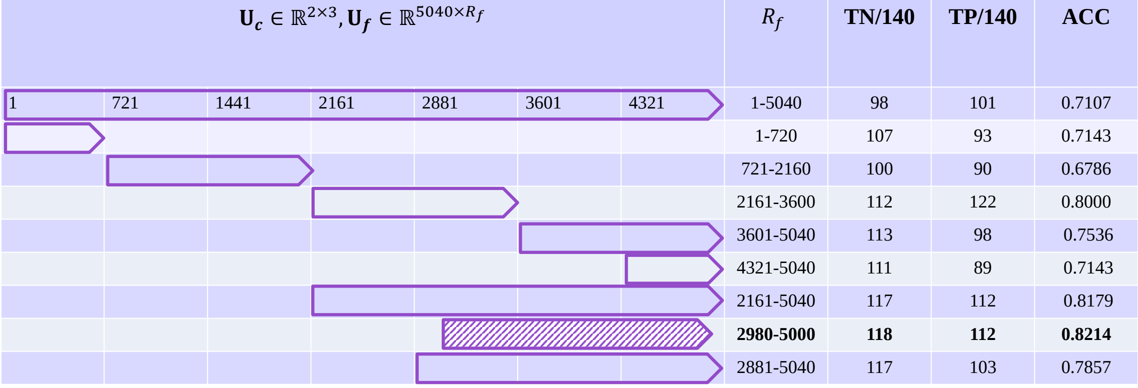
Our work was implemented in MATLAB partially using Tensor Toolbox version 2.6. [12, 13]. Since all videos have constant frame rate 30 fps, we extracted up to 7 frames for each video by snapping almost one frame per each 30 seconds using OpenCV library in Python. Moreover, for detecting facial landmarks, we used pretrained dlib face detector[8] which is created using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and sliding window detection scheme [83]. For the second step, we calculated the SVD rank r where the r is equal to ”number of train videos $\times 7 = 720 \times 7 = 5040$” for all experiments. The intuition behind this estimation is to have an individual component for each frame. Moreover, in contrast to many deep learning approaches for Deepfake detection, our approach does not require GPU base configuration and both train and test steps can be executed on an ordinary CPU based configuration. The description of the CPU based configuration we experimented on is as follows: Intel(R) Core (TM) i5-8600K CPU @3.60GHz,CentOS Linux 7 (Core) operating system and 40GB RAM memory.

9.3.3 Evaluation

Classification performance

Classification performance of our proposed multilinear framework when we keep all of the components as well as when we truncate different ranges of components, is illustrated in Fig. 9.4. In this

[8] http://dlib.net/face_landmark_detection.py.html


$\mathbf{U}_c \in \mathbb{R}^{2\times 3}, \mathbf{U}_f \in \mathbb{R}^{5040\times R_f}$
1
721
1441
2161
2881
3601
4321

R_f	TN/140	TP/140	ACC
1-5040	98	101	0.7107
1-720	107	93	0.7143
721-2160	100	90	0.6786
2161-3600	112	122	0.8000
3601-5040	113	98	0.7536
4321-5040	111	89	0.7143
2161-5040	117	112	0.8179
2980-5000	**118**	**112**	**0.8214**
2881-5040	117	103	0.7857

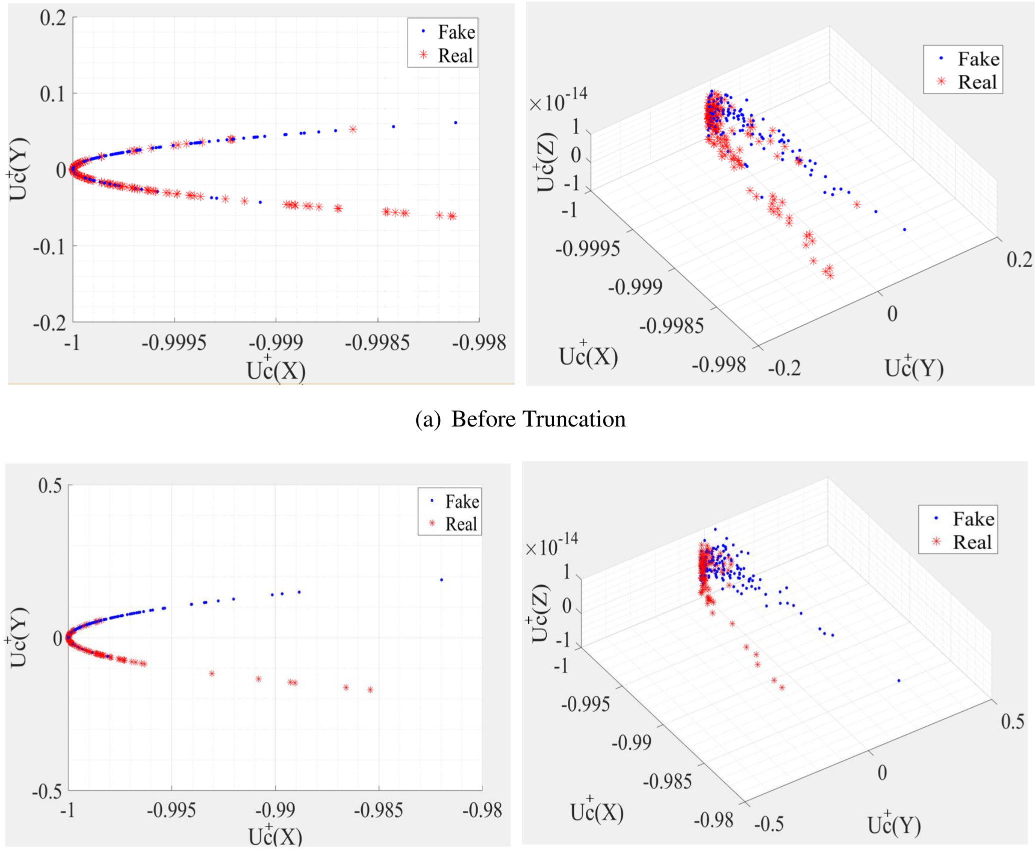
Figure 9.4: Dimensionality reduction experiments video frames compressed with quantization 23. Truncating top 2979 and bottom 40 components of the core tensor corresponding to factor matrix $\mathbf{U}_c$ increases the classification performance. Significant component mostly represent high level structures while insignificant ones may represent noise e.g., artifacts which we want to leverage as a discriminating feature.

Fig, TN, TP, and ACC. denote true negative, true positive, and accuracy respectively.

As demonstrated, truncating top 2980 and bottom 40 components, significantly improves the classification accuracy. In this work, we aim to find discriminating representations for outer ring of real vs Deepfake videos introduced by synthesizing artifacts. Thus, we hypothesize the noisy components i.e., components with insignificant values may represent those artifacts. So, by truncating the top components, we avoid high level facial structures and only keep those that correspond to what we aim to capture i.e., artifacts, for the classification. Moreover, the last 40 components are the most insignificant ones that might be introduced by noises other than synthesizing artifacts. Anyhow, keeping components in range $2980-5000$ results in around 0.82% accuracy.

Effects of truncation on class representations

To clarify the efficacy of truncation, we depict the PCA coefficients of column vectors of $\mathbf{U}_\mathbf{c}^+$ for test frames before and after applying truncation. The distribution of the PCA coefficients is demon-



(a) Before Truncation

(b) After truncation (2980-5000)

Figure 9.5: Distribution of class coefficients before and after truncation. As illustrated, after truncation, data points of each class get closer to each other and as a result, the number of outliers decreases significantly and and the classes are more linearly separable. In these plots, there are scale differences in the axes but in reality the distributions are nearly straight lines.

strated in Fig.9.5. As shown, truncating the undiscriminating components, makes coefficients of each class more similar and as a result put them closer to each other. Specially in case of samples that are located in outer parts of the semicircle i.e., outliers. In other words, the representations after truncation are more linearly separable than those before applying the truncation.

9.4 Conclusions

In this chapter, we leverage the region that we hypothesize has the highest concentration of artifacts, the face outer ring, for classification of Deepfakes using our proposed multilinear framework. Our preliminary results show that using only the outer facial ring we achieve 82% accuracy. However, the outer face ring, the inner face ring and the entire face can be treated as either items in a weighted ”bag of parts”, or as items in a weighted hierarchy of parts on which a compositional hierarchical tensor factor analysis can be performed [166, 167].

10

Conclusions and Future Work

In this chapter, conclusions and future works are presented separately sorted by each chapter.

10.1 Chapter 3: Content-based Techniques for Misinformation Detection

Summary In this chapter, we propose a tensor-based semi-supervised framework for misinformation detection. We propose three different tensor-based models which decomposing them into factor matrices provides us with patterns that indicate different categories of news. We leverage a tensor-based modeling along with graph-based representation and label propagation technique to detect misinformation in a semi-supervised manner. Experimental results on real-world datasets illustrates that our approach outperform many state-of-the art content based methods in terms of accuracy even when we use very few known labels. We contribute to different areas of fake news detection research:

Data-oriented contributions We contribute to the data-oriented fake news research by creating a large-scale multi-class dataset which comprises more than a million tweets and more than 400K articles. This dataset includes detailed information about tweets, articles and users. Each article is labeled as one of the categories proposed by crowed source B.S. Detector.

Feature-oriented contributions We contribute to the feature oriented research by introducing a novel cue i.e., higher order co-occurrences of the words in form of co-occurrence tensors to distinguish fake content from real one. We show how higher order co-occurrence tensor i.e. `TTA` could be leveraged for extracting latent patterns that discriminate real content from fake articles.

Model-oriented contributions We contribute to the model-oriented fake news research by developing a novel tensor-based semi-supervised model for detecting misinformation from article contents which outperforms previous work while using only 1% of labeled data.

Application-oriented contributions Top word co-occurrence patterns of different categories of fake news captured by our proposed TTA could be leveraged by fact checkers to diffuse the spread of misinformation while reviewing unknown articles.

Future work Higher order words co-occurrences shows great promise for variety of text related tasks. For instance, as we discussed in chapter 4, TTA modeling not only is applicable in document classification task, but also could be leveraged for data augmentation as well 4. We believe that the co-occurrences tensors introduced in this chapter could also be exploited for a variety of other Natural Language Processing tasks. In future, we will investigate capabilities of the TTA model in other NLP-based applications.

Resulting publications

- **Abdali., S.**, Bastidas G., G., Shah., N., Papalexakis., E., E., Tensor Embeddings for Content-based Misinformation Detection with Limited, Springer International Publishing 2020
  https://link.springer.com/chapter/10.1007/978-3-030-42699-6˙7
- Bastidas G., G., **Abdali., S.**, Shah., N., Papalexakis., E., E., Semi-supervised Content-based Detection of Misinformation via Tensor Embeddings, ASONAM, Barcelona, Spain. 2018
  https://arxiv.org/pdf/1804.09088.pdf
- Gisel Bastidas Guacho, **Sara Abdali**,

- Bastidas G., G., **Abdali., S.**, Papalexakis., E., E., Semi-supervised content-based fake news detection using tensor embeddings and label propagation, SoCal NLP Symposium. 2018 https://www.cs.ucr.edu/ epapalex/papers/socal-nlp18.pdf

10.2 Chapter 4: `Vec2Node` A Tensor-based Augmentation Technique for Few Shot Learning

Summary In this chapter, we propose a novel tensor-based technique i.e., `Vec2Node`, for augmenting textual datasets and classifiers leveraging local and global information in corpus. `Vec2Node` leverages tensor-based data augmentation with self-training and consistency learning. Our experiments demonstrate that synthetic data generated by `Vec2Node` are interpretable and significantly improve the classification accuracy of text classifiers in few shot settings. Our contributions are:

Model-oriented contributions We create a novel tensor embedding based data augmentation technique for text classification in few shot settings where there is very few labels. We incorporate tensor-embedding, self training and concept drift learning to create a robust framework for improving the performance of text classifiers. As shown in this chapter, the proposed `Vec2Node` not only is applicable to news article classification, but also could be leveraged for other text classification tasks such as sentiment analysis. It could also be used for decreasing the class imbalance when there are rare classes in the dataset.

Future work `Vec2Node` shows great promise in terms of improving classification performance and interpretebility of generated samples. A possible direction to build upon this work is to incorporate Vec2Node framework into deep text generators such as GPT3 and leverage tensor embedding to produce interpretablity for generated samples. We reserve this direction for future work.

Resulting publication

- **Abdali., S.**, Mukherjee., S., Papalexakis., E.,Vec2Node: Self-training with Tensor Augmentation for Text Classification with Few Labels Manuscript under Review 2021.

10.3 Chapter 5: A Hybrid Summarization Approach for Extracting Misinformative Key Phrases

Summary In this chapter, we propose a hybrid approach for extracting key phrases of different categories of misinformation in order to recognize misinformative parts and diffuse further spreed of misinformation. Our experimental results illustrates that using the hybrid approach not only removes the redundant phrases, but also achieves shorter and more understandable phrases in comparison to when we only extract key phrases of original tweets. Our contributions are two folds:

Model-oriented contributions We propose a hybrid pipeline that leverages both extractive and abstractive summarization techniques. More precisely, we leverage BERT transformer to summarize tweets that share a certain type of misinformation into a shorter and more abstract summary and then extract key phrases using RAKE algorithm.

Application-oriented contributions In this chapter, we propose an intervention method which aims to extract key phrases of tweets including a link to a fake article to help both users and fact checkers to recognize and discriminate similar topics, hate or bias languages and use the list of keywords to 1) highlight suspicious key phrases and 2) use the list of key phrases as a baseline for recognizing similar content in unlabeled tweets.

Future work In this chapter, we observe a considerable improvement in terms of quality of extracted keywords. To improve the performance of summarization even more, we may use tensor based co-occurrences graph that we proposed in chapter 3. We showed how leveraging tensor embedding enables us to capture higher order co-occurrences which results in capturing more contextual information. This may improve the ranking process of RAKE algorithm.

10.4 Chapter 6: HiJoD A Tensor-based Ensemble Technique for Misinformation Detection

Summary In this chapter, we propose HiJoD, a 2-level decomposition framework that integrates different aspects of an article i.e., content, social context and domain information towards more precise discovery of misinformation. We show that HiJoD not only is able to detect misinformation in a semi-supervised settings even when we use only 10% of the labels, but also is an order of magnitude faster than similar ensemble approaches in terms of execution time. Our main contributions in this chapter are four-fold:

Data-oriented contributions we extracted Tweets hashtags and HTML sources and a variety of other information about the domains and added them to the dataset we generated earlier. As is proposed in this chapter, HTML source information could be leveraged as an estimation of overall look of the websites that serve the articles.

Feature-oriented contributions In this chapter we introduce the following features:

- Social context in form of co-occurrences of words in an article and hashtags assigned to them
- Publisher webpage information in form of frequencies of HTML tags

we combined these features with content based feature we introduced earlier i.e., higher order word co-occurrences and use the combination as a manifold feature for detecting misinformation.

Model-oriented contributions We develop a three-stage semi-supervised approach with a novel hierarchical Joint decomposition technique i.e., `HiJoD`. We first decompose each model separately to find the latent patterns of the articles with respect to the corresponding aspect, then we use a strategy to find shared components of the individual patterns. Next, we leverage these manifold patterns to model articles and their similarities using KNN graph and semi-supervised belief propagation technique.

Application-oriented contributions As far as the diffusion fake news research is concerned, we show that there is a difference between the latent manifold patterns of fake news and real news articles. Leveraging theses patterns enables us to discriminate fake content from the reliable one and possibly use them for early detection of misinformation.

Future work In future, we aim to explore more discriminating features and investigate the performance and scalability of our proposed technique while adding more features. misinformation. Another direction to build upon this work is to customize and apply our proposed `HiJoD` for detecting misinformation in multimodal platforms.

Resulting publication

- **Abdali., S.**, Shah., N., Papalexakis., E., E :Semi-SupervisedMulti-aspect Detection of Misinformation using Hierarchical Joint Decomposition, ECML/PKDD 2020, Ghent, Belgium2020 https://www.springerprofessional.de/en/semi-supervised-multi-aspect-detection-of-misinformation-using-h/18900038 https://arxiv.org/abs/2005.04310

10.5 Chapter 7: K-Nearest Hyperplanes Graph (KNH) for Misinformation Detection

Summary In this chapter, we introduce a novel multi-aspect modeling of the articles i.e. K-Nearest Hyperplane Graph (KNH) by generalizing the classic KNN graph. Our main contribution is as follows:

Model-oriented contributions We propose hypernodes to represent articles. Hypernodes are defined by vectors that represent an article with respect to different aspects. We also propose a novel embedding representation that encodes the relative position of the hypernodes in the space which could be leverages for classification of news articles. We experiment on two real world datasets. We observe that not only KNH outperforms the KNN graph in terms of F1 score, but also is significantly more robust against improper rank representations.

Future Work This chapter shows great promise of KNH for multi-aspect modeling. In the future, we plan to extend our investigation on 1) higher dimensional mathematical formulation of hypernodes, which requires an additional discussion and we skipped it here due to the space limitation, 2) capability of hypernodes i.e., common spaces for extrapolating missing/unknown features, and 3) defining hyperedges that consider higher order relationships between the subspaces. We reserve all the aforementioned directions for future work.

Resulting publication

- **Abdali., S.**, Shah., N., Papalexakis., E., KNH: Multi-View Modeling with K-Nearest Hyperplanes Graph for Misinformation Detection. TrueFact workshop-KDD 2020 https://arxiv.org/pdf/2102.07857.pdf

10.6 Chapter 8: Tensor Emdedding for Misinformation Detection from Website Screenshots

Summary In this chapter, we propose leveraging screenshots of news articles for misinformation detection. We propose a tensor-based model and semi-supervised classification pipeline i.e.,`VizFake` which outperforms text-based methods and state-of-the-art image based deep learning models and is over 200 times faster, while also being easier to fine-tune and more practical. Moreover, `VizFake` is resistant to some common image transformations like grayscaling and changing the resolution, as well as partial corruptions of the screenshot. Our contributions are fourfold:

Data-oriented contributions We implemented a toolbox to automatically take screenshots of the scrolled articles. Leveraging this toolbox, we have collected more than 60K article screenshots. We have also labeled them into one of the categories of fake news using B.S. Detector crowd sourcing toolbox.

Feature-oriented contributions We propose a very important yet neglected feature for detecting misinformation, i.e., the overall look of serving domain. We hypothesise that unreliable sources tend to be visually messy and full of advertisements and popups while trustworthy domains often look professional and ordered. We use screenshots of the webpages to capture these visual differences between real and fake articles.

Model-oriented contributions we propose a novel tensor-based semi-supervised model i.e., `VizFake` for classification of screenshot images. `VizFake` is fast, efficient, robust to image resolution, and missing image segments, and more importantly data-limited. Moreover, `VizFake` has exploratory capabilities i.e., it could be used for unsupervised soft-clustering of the articles.

Application-oriented contributions In this chapter, we show that there is a difference between latent pattern images of fake and real classes. We demonstrate that contrary to the real class, images of fake class have darker pixels i.e., the dark portion of the image is wider. These patterns could be leverages for unsupervised classification of articles and misinformation diffusion research.

Future Work Collecting annotation for misinformation detection is a complicated and time-consuming task and as we increase the granularity of the labels from domain level to articles level and even article sections it becomes harder and harder. Moreover, the majority of available ground truth resources like “B.S. Detector” or ”NewsGuard” provide labels pertain to domains rather than articles. However, there are useful article-level information like web events content that can be taken advantage of when we have grainier labels and capturing them causes a drop in performance because they may be considered as noise when working with domain level labels. We defer the study of obtaining and using finer-grained labels for future work.

Resulting publications

- **Abdali., S.**, Gurav., R., Menon., S., Fonseca., D., Entezari., N., S., Shah., N., Papalexakis., E., E: Identifying Misinformation from Website Screenshots , AAAI International Conference on Web and Social Media (ICWSM) 2021
  https://ojs.aaai.org/index.php/ICWSM/article/view/18036/17839

10.7 Chapter 9: Deepfake Detection with Multilinear Projection

Summary In this chapter, we propose a novel approach for detecting Deepfake videos. We propose a novel cue i.e., facial outer ring or the region that we hypothesize has the highest concentration of warping artifacts. Then, We propose a multilinear framework for classification of Deepfake videos using facial outer rings. Our results show that using only the outer facial ring and a handful of frames we achieve admissible accuracy. Our main contributions are as follows:

Feature-oriented contributions we propose facial outer ring i.e., the blending boundaries by estimating the blending region and non-facial pixels using facial landmarks and canceling out the inner blending parts. We leverage this remaining region i.e., facial outer ring which has the highest concentration of introduced artifacts as a cue for Deepfakes video detection.

Model-oriented contributions we propose a novel multilinear model to represent Deepfake and real class information and then we apply a multilinear projection technique to detect Deepfakes using only 7 frames per video.

Application-oriented contributions We show that there is a difference between the eigenfaces of deepfakes and original videos specially those corresponding to noisy elements. These eigenfaces could be leveraged to differentiate manipulated frames from pristine ones.

Future Work The outer facial ring, the inner face ring and the entire face can be treated as either items in a weighted ”bag of parts”, or as items in a weighted hierarchy of parts on which a compositional hierarchical tensor factor analysis can be performed [166, 167]. Another direction for achieving higher accuracy, is to use binary masks released by [127]. The binary mask can be leveraged for precise segmentation of the frames into regions of interest. We reserve the aforementioned directions for future work.

Resulting publication

- **Abdali., S.**, Vasilescu., O. A, Papalexakis., E., Deepfake Representation with Multilinear Regression MIS2-KDD: The Second International MIS2 Workshop: Misinformation and Misbehavior Mining on the Web. 2021

  https://arxiv.org/pdf/2108.06702.pdf

10.8 List of Publications

- **Abdali., S.**, Mukherjee., S., Papalexakis., E.,Vec2Node: Self-training with Tensor Augmentation for Text Classification with Few Labels Manuscript under Review 2021.
- **Abdali., S.**, Vasilesco., O. A, Papalexakis., E., Deepfake Representation with Multilinear Regression MIS2-KDD: The Second International MIS2 Workshop: Misinformation and Misbehavior Mining on the Web. 2021
- **Abdali., S.**, Gurav., R., Menon., S., Fonseca., D., Entezari., N., S., Shah., N., Papalexakis., E., E: Identifying Misinformation from Website Screenshots , AAAI International Conference on Web and Social Media (ICWSM) 2021
- **Abdali., S.**, Shah., N., Papalexakis., E., E Semi-SupervisedMulti-aspect Detection of Misinformation using Hierarchical Joint Decomposition, ECML/PKDD, Ghent, Belgium 2020
- **Abdali., S.**, Shah., N., Papalexakis., E., KNH: Multi-View Modeling with K-Nearest Hyperplanes Graph for Misinformation Detection. TrueFact workshop-KDD 2020
- **Abdali., S.**, Bastidas G., G., Shah., N., Papalexakis., E., E., Tensor Embeddings for Content-based Misinformation Detection with Limited, Springer International Publishing 2020
- Bastidas G., G., **Abdali., S.**, Shah., N., Papalexakis., E., E., Semi-supervised Content-based Detection of Misinformation via Tensor Embeddings, ASONAM, Barcelona, Spain. 2018
- Bastidas G., G., **Abdali., S.**, Papalexakis., E., E., Semi-supervised Content-based Fake News Detection using Tensor Embeddings and Label propagation, SoCal NLP Symposium. 2018

Bibliography

[1] Mining misinformation in social media, author=Liang Wu, Fred Morstatter, Xia Hu, Huan Liu, booktitle=AAAI, year=2016,.

[2] Sara Abdali, Gisel G. Bastidas, Neil Shah, and Evangelos E. Papalexakis. Tensor Embeddings for Content-Based Misinformation Detection with Limited Supervision, pages 117–140. Springer International Publishing, Cham, 2020.

[3] Sara Abdali, Gisel G. Bastidas, Neil Shah, and Evangelos E. Papalexakis. Tensor Embeddings for Content-Based Misinformation Detection with Limited Supervision. Springer International Publishing, Cham, 2020.

[4] Evrim Acar, Yuri Levin-Schwartz, Vince Calhoun, and Tulay Adali. Acmtf for fusion of multi-modal neuroimaging data and identification of biomarkers. 08 2017.

[5] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. 09 2018.

[6] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting World Leaders Against Deep Fakes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, page 8, Long Beach, CA, June 2019. IEEE.

[7] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery, 29:626–688, 2014.

[8] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. Text summarization techniques: A brief survey, 2017.

[9] B. W. Bader and T. G. Kolda. Matlab tensor toolbox version 2.6. Available online, February 2015.

[10] Brett Bader and Tamara Kolda. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. ACM Trans. Math. Softw., 32:635–653, 01 2006.

[11] Brett W. Bader and Tamara G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. Transactions on Mathematical Software, 32(4):635–653, December 2006.

[12] Brett W. Bader and Tamara G. Kolda. Efficient MATLAB computations with sparse and factored tensors. SIAM Journal on Scientific Computing, 30(1):205–231, December 2007.

[13] Brett W. Bader, Tamara G. Kolda, et al. Matlab tensor toolbox version 2.6. Available online, February 2015.

[14] Md Jawadul Bappy, Cody Simons, Lakshmanan Nataraj, B. Manjunath, and Amit Roy-Chowdhury. Hybrid lstm and encoder-decoder architecture for detection of image forgeries. IEEE Transactions on Image Processing, PP, 01 2019.

[15] J.S. Baruni and Dr. J.G.R . Sathiaseelan. Keyphrase extraction from document using rake and textrank algorithms. 2020.

[16] K. G. Binmore. The foundations of topological analysis: A straightforward introduction. 1981.

[17] Brandon C Boatwright, Darren L Linvill, and Patrick L Warren. Troll factories: The internet research agency and state-sponsored agenda building. Resource Centre on Media Freedom in Europe, 2018.

[18] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606, 2016.

[19] Allan Borodin, Rafail Ostrovsky, and Yuval Rabani. Subquadratic approximation algorithms for clustering problems in high dimensional spaces: Theoretical advances in data clustering (guest editors: Nina mishra and rajeev motwani). Machine Learning, 56, 07 2004.

[20] A. Braunstein, M. Mézard, and R. Zecchina. Survey propagation: An algorithm for satisfiability. Random Struct. Algorithms, 27(2):201–226, September 2005.

[21] R. Bro. Parafac: Tutorial and applications. In In Chemom. Intell. Lab Syst., Special Issue 2nd Internet Cont. in Chemometrics (INCINC'96), volume 38, pages 149–171, 1997.

[22] B.S. Detector. http://bsdetector.tech/, 2019.

[23] Matteo Cardaioli, Stefano Cecconello, Mauro Conti, Luca Pajola, and Federico Turrin. Fake news spreaders profiling through behavioural analysis notebook for pan at clef 2020. 01 2021.

[24] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. Psychometrika, 35:283–319, 1970.

[25] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In Proceedings of the 20th International Conference on World Wide Web, WWW '11, pages 675–684, New York, NY, USA, 2011. ACM.

[26] Pfeffer Juergen Stempeck Matt Castillo Carlos, El-Haddad Mohammed. Characterizing the life cycle of online news stories using social media reactions. Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW, 04 2013.

[27] Fabio Celli. Adaptive personality recognition from text. 2013.

[28] Yimin Chen, Nadia Conroy, and Victoria Rubin. Misleading online content: Recognizing clickbait as “false news”. 11 2015.

[29] François Chollet. Xception: Deep learning with depthwise separable convolutions. pages 1251–1258, 2017.

[30] Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. Computational fact checking from knowledge networks. PloS one, 10(6):e0128193, 2015.

[31] I. Cook. Oxford dictionary of statistics. page 104, 2002.

[32] Dianne Cyr. Website design, trust and culture: An eight country investigation. ECRA, 12, 11 2013.

[33] L. de Lathauwer, B. de Moor, and J. Vandewalle. A multilinear singular value decomposition. SIAM J. of Matrix Analysis and Applications, 21(4):1253–78, 2000.

[34] L. de Lathauwer, B. de Moor, and J. Vandewalle. On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors. SIAM J. of Matrix Analysis and Applications, 21(4):1324–42, 2000.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, pages 248–255, 2009.

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL, 2019.

[38] Yingtong Dou, Kai Shu, Congying Xia, Philip Yu, and Lichao Sun. User preference-aware fake news detection. pages 2051–2055, 07 2021.

[39] Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. Self-training improves pre-training for natural language understanding, 2020.

[40] M. A. Rasmussen E. Acar, A. J. Lawaetz and R. Bro. Structure-revealing data fusion model with applications in metabolomics. In IEE EMBS, 2013.

[41] T. Kolda E. Acar and D. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. In KDD Workshop on Mining and Learning with Graphs, 2011.

[42] A. Elgammal and C. S. Lee. Separating style and content on a nonlinear manifold. In In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), volume I, page 478–485, 2004.

[43] Soheil Esmaeilzadeh, Gao Peh, and Angela Xu. Neural abstractive text summarization and fake news detection. 03 2019.

[44] María Espinosa, Roberto Centeno, and Álvaro Rodrigo. Analyzing user profiles for detection of fake news spreaders on twitter - notebook for pan at clef 2020. 09 2020.

[45] Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In ACL'12, page 171–175, 2012.

[46] E. Fersini, Justin Armanini, and Michael D'Intorni. Profiling fake news spreaders: Stylometry, personality, emotions and embeddings. In CLEF, 2020.

[47] Z. Pan G. Liu, M. Xu and A. E. Rhalibi. Human motion generation with multifactor models. In Computer Animation and Virtual Worlds, page 351–359, 2011.

[48] Deepali K. Gaikwad and C. Namrata Mahender. A review paper on text summarization. 2016.

[49] Fabio Gallo, Gerardo Simari, Maria Vanina Martinez, and Marcelo Falappa. Predicting user reactions to twitter feed content based on personality type and social cues. Future Generation Computer Systems, 110, 11 2019.

[50] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. Discrete Applied Mathematics, 42:177–201, 04 1993.

[51] A. Gelman and G. Imbens. Why ask why? forward causal inference and reverse causal questions. 2013.

[52] G. B. Guacho, S. Abdali, N. Shah, and E. E. Papalexakis. Semi-supervised content-based detection of misinformation via tensor embeddings. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pages 322–325, Aug 2018.

[53] David Guera and Edward Delp. Deepfake video detection using recurrent neural networks. pages 1–6, 11 2018.

[54] A. Gupta, H. Lamba, and P. Kumaraguru. $1.00 per rt #bostonmarathon #prayforboston: Analyzing fake content on twitter. In APWG eCrime Researchers Summit, pages 1–12, Sept 2013.

[55] Aditi Gupta and Ponnurangam Kumaraguru. Credibility ranking of tweets during high impact events. In Proceedings of the 1st Workshop on Privacy and Security in Online Social Media, PSOSM '12, pages 2:2–2:8, New York, NY, USA, 2012. ACM.

[56] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy. In Proceedings of the 22Nd International Conference on World Wide Web, WWW '13 Companion, pages 729–736, New York, NY, USA, 2013. ACM.

[57] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. pages 729–736, 05 2013.

[58] Manish Gupta, Peixiang Zhao, and Jiawei Han. Evaluating Event Credibility on Twitter, pages 153–164.

[59] Manish Gupta, Peixiang Zhao, and Jiawei Han. Evaluating event credibility on twitter. SIAM International Conference In Data Mining, pages 153–164, 04 2012.

[60] Shashank Gupta, Raghuveer Thirukovalluru, Manjira Sinha, and Sandya Mannarswamy. Cimtdetect: A community infused matrix-tensor coupled factorization based method for fake news detection. pages 278–281, 08 2018.

[61] Jiawei Han, Micheline Kamber, and Jian Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[62] Momchil Hardalov, Ivan Koychev, and Preslav Nakov. In search of credible news. page 172–180, 2016.

[63] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. volume 16, page 84, 1970.

[64] Philipp Hartl and Udo Kruschwitz. University of regensburg at checkthat! 2021: Exploring text summarization for fake news detection. 09 2021.

[65] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation, 2020.

[66] Stefan Helmstetter and Heiko Paulheim. Weakly supervised learning for fake news detection on twitter. In IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining(ASONAM), pages 274–277, 2018.

[67] Javier Hernandez-Ortega, Ruben Tolosana, Julian Fierrez, and Aythami Morales. Deepfakeson-phys: Deepfakes detection based on heart rate estimation, 2020.

[68] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In NIPS, pages 1731–1741, 2017.

[69] P. W. Holland. Statistics and causal inference: Comment: Statistics and metaphysics. In American Statistical Association, 1986.

[70] Benjamin D. Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. volume abs/1703.09398, 2017.

[71] Seyedmehdi Hosseinimotlagh and Evangelos E. Papalexakis. Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. 2017.

[72] Yuchi Huang, Qingshan Liu, Fengjun Lv, Yihong Gong, and Dimitris N. Metaxas. Unsupervised image categorization by hypergraph partition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33:1266–1273, 2011.

[73] Z. Jin, J. Cao, Y. G. Jiang, and Y. Zhang. News credibility evaluation on microblog with a hierarchical propagation model. In 2014 IEEE International Conference on Data Mining, pages 230–239, Dec 2014.

[74] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian. Novel visual and statistical image features for microblogs news verification. Transactions on Multimedia, 19(3):598–608, March 2017.

[75] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. News verification by exploiting conflicting social viewpoints in microblogs. 2016.

[76] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651, 2016.

[77] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification, 2016.

[78] Kai K. Shu and H. Liu T. Le, D. lee. Deep headline generation for clickbait detection. In IEEE International Conference on Data Mining(ICDM), pages 467–476, 2018.

[79] S. Wang K. Shu, A. Sliva and H. Liu. Beyond news contents: the role of social context for fake news detection. In WSDM '19 Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pages 312–320, 2019.

[80] S. Wang D. Lee K. Shu, L. Cui and H. Liu. Exploiting tri-relationship for fake news detection. arXiv preprint arXiv:1712.07709, 2017.

[81] S. Wang D. Lee K. Shu, L. Cui and H. Liu. defend: Explainable fake news detection. In Proceedings of 25th ACM SIGKDD international conference on Knowledge discovery and data mining, 2019.

[82] Christopher Kanan and Garrison Cottrell. Color-to-grayscale: Does the method matter in image recognition? PloS, 7:e29740, 01 2012.

[83] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 1867–1874, 2014.

[84] J Kietzmann, L Lee, I McCarthy, and T Kietzmann. Deepfakes: trick or treat? Business Horizons, December 2019.

[85] Gi Yeon Kim and Youngjoong Ko. Graph-based fake news detection using a summarization technique. In EACL, 2021.

[86] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.

[87] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. SIAM Review, 51(3):455–500, September 2009.

[88] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. Tensorly: Tensor learning in python. The Journal of Machine Learning Research, 20(1):925–930, 2019.

[89] Danai Koutra, Tai-You Ke, U. Kang, Duen Chau, Hsing-Kuo Pao, and Christos Faloutsos. Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms. In Machine Learning and Knowledge Discovery in Databases (ECML/PKDD), volume 6912 of Lecture Notes in Computer Science, pages 245–260. 2011.

[90] S. Wang D. Lee K.Shu, D. Mahudeswaran and H. Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. arXiv preprint arXiv:1809.01286, 2018.

[91] Srijan Kumar and Neil Shah. False information on web and social media: A survey. arXiv preprint arXiv:1804.08559, 2018.

[92] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In Blai Bonet and Sven Koenig, editors, AAAI, volume 333, pages 2267–2273, 2015.

[93] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. ICML 2014, 4, 05 2014.

[94] Lee, D. D., Seung, H. S.,. Algorithms for non-negative matrix factorization. In Neural Inf. Process. Syst., page 556–562, 2001.

[95] Xi Li, Yao Shen Li, Chunhua Shen, Anthony Dick, and Anton Hengel. Contextual hypergraph modeling for salient object detection. Proceedings of the IEEE International Conference on Computer Vision, 10 2013.

[96] Xi-Lin Li, Matthew Anderson, and Tülay Adalı. Second and higher-order correlation analysis of multiple multidimensional variables by joint diagonalization. In Vincent Vigneron, Vicente Zarzoso, Eric Moreau, Rémi Gribonval, and Emmanuel Vincent, editors, Latent Variable Analysis and Signal Separation, pages 197–204, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[97] Xinzhe Li, Qianru Sun, Yaoyao Liu, Shibao Zheng, Qin Zhou, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification, 2019.

[98] Yuezun Li, Ming-Ching Chang, Hany Farid, and Siwei Lyu. In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking. 06 2018.

[99] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. CoRR, abs/1811.00656, 2018.

[100] L. Lim and P. Comon. Blind multilinear identification. In IEEE Transactions on Information Theory, page 1260–1280, 2014.

[101] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. 2017.

[102] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3730–3740, Hong Kong, China, November 2019. Association for Computational Linguistics.

[103] Eric Lock, Katherine Hoadley, J.S. Marron, and Andrew Nobel. Joint and individual variation explained (jive) for integrated analysis of multiple data types. volume 7, pages 523–542, 03 2013.

[104] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung 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. AAAI Press, 2016.

[105] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, pages 1751–1754, New York, NY, USA, 2015. ACM.

[106] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[107] S. McCloskey and M. Albright. Detecting gan-generated imagery using saturation cues. In 2019 IEEE International Conference on Image Processing (ICIP), pages 4584–4588, 2019.

[108] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: Can we trust what we rt? In Proceedings of the First Workshop on Social Media Analytics, SOMA '10, pages 71–79, New York, NY, USA, 2010. ACM.

[109] Subhabrata Mukherjee and Ahmed Hassan Awadallah. Uncertainty-aware self-training for text classification with few labels, 2020.

[110] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In AAAI, 2017.

[111] NewsGuard. https://www.newsguardtech.com, 2019.

[112] Huy Nguyen, Fuming Fang, Junichi Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. 06 2019.

[113] Huy Nguyen, Junichi Yamagishi, and I. Echizen. Use of a capsule network to detect fake images and videos. 10 2019.

[114] Huy Nguyen, Junichi Yamagishi, and I. Echizen. Use of a capsule network to detect fake images and videos. 10 2019.

[115] Ray Oshikawa, Jing Qian, and William Yang Wang. A survey on natural language processing for fake news detection. arXiv:1811.00770, 2018.

[116] C. Xiang P. Liu, X. Wang and W. Meng. A survey of text data augmentation, 2020.

[117] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2009.

[118] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. ACM Trans. Intell. Syst. Technol., 8:16:1–16:44, 2016.

[119] Papalexakis, Evangelos E. Automatic unsupervised tensor mining with quality assessment. In SIAM SDM, 2016.

[120] Georgios Pavlopoulos, Maria Secrier, Charalampos Moschopoulos, Theodoros Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis Bagos. Using graph theory to analyze biological networks. BioData mining, 4:10, 04 2011.

[121] K. Tsioutsiouliklis P.Biyani and J. Blackmer. "8 amazing secrets for getting more clicks": detecting clickbaits in news streams using article informality. In AAAI'16 Proceedings of the Thirtieth AAAI Conference on Artificial, pages 94–100, 2016.

[122] Dan Pelleg and Andrew Moore. Accelerating exact k-means algorithms with geometric reasoning. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 277–281. ACM, 1999.

[123] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. Rumor has it: Identifying misinformation in microblogs. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pages 1589–1599, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[124] Meredith Ringel Morris, Scott Counts, Asta Roseway, Aaron Hoff, and Julia Schwarz. Tweeting is believing?: Understanding microblog credibility perceptions. pages 441–450, 02 2012.

[125] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic Keyword Extraction from Individual Documents, pages 1 – 20. 03 2010.

[126] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. arXiv, 2018.

[127] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In International Conference on Computer Vision (ICCV), 2019.

[128] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics++: Learning to detect manipulated facial images. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct 2019.

[129] D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. 81:688–701, 1974.

[130] Victoria Rubin, Niall Conroy, and Yimin Chen. Towards news verification: Deception detection methods for news discourse. 01 2015.

[131] Victoria L. Rubin, Niall J. Conroy, Yimin Chen, and Sarah Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. 2016.

[132] Natali Ruchansky, Sungyong Seo, and Yan Liu. CSI: A hybrid deep model for fake news. volume abs/1703.06959, 2017.

[133] E.E. Papalexakis S. Abdali, N. Shah. Semi-supervised multi-aspect detection of misinformation using hierarchical joint decomposition. ECML/PKDD, 2020.

[134] Neil Shah, Alex Beutel, Brian Gallagher, and Christos Faloutsos. Spotting suspicious link behavior with fbox: An adversarial perspective. In Data Mining (ICDM), 2014 IEEE International Conference on, pages 959–964. IEEE, 2014.

[135] Chengcheng Shao, Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. Hoaxy: A platform for tracking online misinformation. Proceedings of the 25th International Conference Companion on World Wide Web, 2016.

[136] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. Defend: Explainable fake news detection. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining, KDD '19, page 395–405, New York, NY, USA, 2019. Association for Computing Machinery.

[137] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. SIGKDD Explor. Newsl., 19, 2017.

[138] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. CoRR, abs/1708.01967, 2017.

[139] Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. The role of user profile for fake news detection. 04 2019.

[140] Weiguang Si, Kota Yamaguchi, and M. Alex O. Vasilescu. Face tracking with multilinear (tensor) active appearance models. http://pdfs.semanticscholar.org/6c64/59d7cadaa210e3310f3167dc181824fb1bff.pdf, Jun 2013.

[141] N.D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. volume PP, 07 2016.

[142] Craig Silverman. This analysis shows how fake election news stories outperformed real news on facebook. BuzzFeed News, "November" 2016.

[143] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

[144] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. arXiv:1711.00489, 2017.

[145] Shaden Smith, Niranjay Ravindran, Nicholas D. Sidiropoulos, and George Karypis. Splatt: Efficient and parallel sparse tensor-matrix multiplication. In Proceedings of International Parallel and Distributed Processing Symposium (IPDPS), pages 61–70, 2015.

[146] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher 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, October 2013. Association for Computational Linguistics.

[147] Andreas Spitz, Dennis Aumiller, Balint Soproni, and Michael Gertz. A versatile hypergraph model for document collections. 2020.

[148] Lifan Su, Yue Gao, Xibin Zhao, Hai Wan, Ming Gu, and Jiaguang Sun. Vertex-weighted hypergraph learning for multi-view object classification. IJCAI'17, page 2779–2785. AAAI Press, 2017.

[149] Liang Sun, Shuiwang Ji, and Jieping Ye. Hypergraph spectral learning for multi-label classification. KDD '08, page 668–676, New York, NY, USA, 2008. Association for Computing Machinery.

[150] Shengyun Sun, Hongyan Liu, Jun He, and Xiaoyong Du. Detecting event rumors on sina weibo automatically. pages 120–131, 04 2013.

[151] Robert Tate. Correlation between a discrete and a continuous variable. point-biserial correlation. The Annals of Mathematical Statistics, 25, 09 1954.

[152] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. arXiv preprint arXiv:2001.00179, 2020.

[153] Mathew A. Turk and Alex P. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71–86, 1991.

[154] M. Vasilescu and D. Terzopoulos. Adaptive meshes and shells: Irregular triangulation, discontinuities, and hierarchical subdivision. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'92), page 829–832, Champaign, IL, Jun 1992.

[155] M. A. O. Vasilescu. An algorithm for extracting human motion signatures. In IEEE Conf. on Computer Vision and Pattern Recognition, Hawai, 2001.

[156] M. A. O. Vasilescu. Human motion signatures: Analysis, synthesis, recognition. In Proc. Int. Conf. on Pattern Recognition, volume 3, pages 456–460, Quebec City, Aug 2002.

[157] M. A. O. Vasilescu. Multilinear projection for face recognition via canonical decomposition. In Proc. IEEE Inter. Conf. on Automatic Face Gesture Recognition (FG 2011), pages 476–483, Mar 2011.

[158] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In Proc. European Conf. on Computer Vision (ECCV 2002), pages 447–460, Copenhagen, Denmark, May 2002.

[159] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, volume II, pages 93–99, Madison, WI, 2003.

[160] M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, volume I, pages 547–553, San Diego, CA, 2005.

[161] M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, volume I, pages 547–553, San Diego, CA, 2005.

[162] M. Alex O. Vasilescu. A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision and Machine Learning. PhD thesis, University of Toronto, 2009.

[163] M. Alex O. Vasilescu. A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision, and Machine Learning. PhD thesis, University of Toronto, 2009.

[164] M. Alex O. Vasilescu. Multilinear projection for face recognition via canonical decomposition. In Proc. IEEE International Conf. on Automatic Face Gesture Recognition (FG 2011), pages 476–483, Mar 2011.

[165] M Alex O Vasilescu. A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision and Machine Learning. PhD thesis, Citeseer, 2012.

[166] M. Alex O. Vasilescu and Eric Kim. Compositional hierarchical tensor factorization: Representing hierarchical intrinsic and extrinsic causal factors. In The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'19): Tensor Methods for Emerging Data Science Challenges, Aug. 5 2019.

[167] M. Alex O. Vasilescu, Eric Kim, and Xiao S. Zeng. Causalx: Causal explanations and block multilinear factor analysis. In 2020 25th International Conference of Pattern Recognition (ICPR 2020), pages 10736–10743, Jan 2021.

[168] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part I, volume 2350 of Lecture Notes in Computer Science, pages 447–460. Springer, 2002.

[169] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA, pages 93–99. IEEE Computer Society, 2003.

[170] M. Alex O. Vasilescu and Demetri Terzopoulos. Tensortextures. In Alyn P. Rockwood, editor, Proceedings of the SIGGRAPH 2003 Conference on Sketches & Applications: in conjunction with the 30th annual conference on Computer graphics and interactive techniques, 2003, San Diego, California, USA, July 27-31, 2003. ACM, 2003.

[171] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear projection for appearance-based recognition in the tensor framework. In IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007, pages 1–8. IEEE Computer Society, 2007.

[172] Luisa Verdoliva. Media forensics and deepfakes: an overview. arXiv preprint arXiv:2001.06564, 2020.

[173] H. Wang and N. Ahuja. Facial expression decomposition. In IEEE Inter. Conf. on Computer Vision (ICCV), volume 2, pages 65–958, 2003.

[174] Meng Wang, Xueliang Liu, and Xindong Wu. Visual classification by -hypergraph modeling. Knowledge and Data Engineering, IEEE Transactions on, 27:2564–2574, 09 2015.

[175] Meng Wang, Xueliang Liu, and Xindong Wu. Visual classification by 1-hypergraph modeling. IEEE Trans. Knowl. Data Eng., 27:2564–2574, 2015.

[176] Suhang Wang, Charu Aggarwal, Jiliang Tang, and Huan Liu. Attributed signed network embedding. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, page 137–146. Association for Computing Machinery, 2017.

[177] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. WWW '17, page 391–400, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[178] Yaqing Wang, Subhabrata Mukherjee, H. Chu, Yuancheng Tu, Miaonan Wu, Jing Gao, and Ahmed Hassan Awadallah. Adaptive self-training for few-shot neural sequence labeling. ArXiv, abs/2010.03680, 2020.

[179] Wang, William, Y. ”liar, liar pants on fire”: A new benchmark dataset for fake news detection. In ACL’17, 2017.

[180] John Wells, Joseph Valacich, and Traci Hess. What signal are you sending? how website quality influences perceptions of product quality and purchase intentions. MIS Quarterly, 35:373–396, 06 2011.

[181] Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, and De Rosal Ignatius Moses Setiadi. Review of automatic text summarization techniques methods. Journal of King Saud University - Computer and Information Sciences, 2020.

[182] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[183] Liang Wu, Jundong Li, Xia Hu, and Huan Liu. Gleaning wisdom from the past: Early detection of emerging rumors in social media. In Proceedings of the 2017 SIAM International Conference on Data Mining, pages 99–107. SIAM, 2017.

[184] Liang Wu and Huan Liu. Tracing fake-news footprints: Characterizing social media messages by how they propagate. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018.

[185] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation. CoRR, abs/1904.12848, 2019.

[186] Y. Wen K. Ramamohanarao C. Xu Y. Luo, D. Tao. Tensor canonical correlation analysis for multi-view dimension reduction. 2015.

[187] Ruoh-Nan Yan, Jennifer Yurchisin, and Kittichai Watchravesringkan. Does formality matter?: Effects of employee clothing formality on consumers’ service quality expectations and store image perceptions. Retail Distribution Management, 39:346–362, 04 2011.

[188] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. pages 8261–8265, 2019.

[189] J.S. Yedidia, W.T. Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. volume 51, pages 2282 – 2312, 08 2005.

[190] Jun Yu, Dacheng Tao, and Meng Wang. Adaptive hypergraph learning and its application in image classification. IEEE Transactions on Image Processing, 21:3262–3272, 2012.

[191] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 9054–9065. Curran Associates, Inc., 2019.

[192] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. International Conference on Learning Representations,ICLR, 2017.

[193] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 28, pages 649–657. Curran Associates, Inc., 2015.

[194] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. volume 19, pages 1601–1608, 01 2006.

[195] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. National Science Review, 5, 08 2017.