I, Amer G. Ghanem, hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Computer Science & Engineering.

It is entitled:
Identifying Patterns of Epistemic Organization through Network-Based Analysis of Text Corpora

Student's name: Amer G. Ghanem

This work and its defense approved by:

Committee chair: Ali Minai, Ph.D.
Committee member: Raj Bhatnagar, Ph.D.
Committee member: Karen Davis, Ph.D.
Committee member: Carla Purdy, Ph.D.
Committee member: James Uber, Ph.D.
Identifying Patterns of Epistemic Organization through
Network-Based Analysis of Text Corpora

A dissertation submitted to the
Graduate School
of the University of Cincinnati
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in the Department of Electrical Engineering and Computing Systems
College of Engineering and Applied Science

by

Amer G. Ghanem

B.S Computer Science, University of Cincinnati, June 2005
M.S Computer Science, University of Cincinnati, Dec 2007

Committee Chair: Ali A. Minai, Ph.D.

Cincinnati Ohio, 2015
This work was supported in part by National Science Foundation CreativeIT grant IIS-0855714 and National Science Foundation INSPIRE grant BCS-1247971. The ideas or conclusions expressed in this dissertation are not those of the National Science Foundation or its employees.
Abstract

The growth of on-line textual content has exploded in recent years, creating truly massive text corpora. As the quantity of text available on-line increases, professionals from different industries such as marketing and politics are realizing the importance of extracting useful information and insights from this treasure trove of data. It is also clear, however, that doing so requires methods that go beyond those developed for classical data processing or even natural language processing. In particular, there is great need for efficient methods that can make sense of the semantic content of this data, and allows new knowledge to be inferred from it.

The research in this dissertation describes a new method for identifying latent structures (topics) in texts through the application of community extraction techniques on associative networks of words. Since humans represent knowledge in terms of associations, it is asserted that deriving topics from associative networks represents a more cognitively meaningful approach than using purely statistical patterns.

The topic identification method proposed in this thesis is called Topic Extraction through Partitioning of Lexical Associative Networks (TExPLAN). It begins by constructing an associative network of words where the strength of their association indicates the frequency of their co-occurrence in documents. Once the word network is constructed, the algorithm proceeds in two stages. In the first stage, a partitioning of the word network takes place using a community extraction method to extract disjoint seed topics. The second stage of TExPLAN uses the connectivity of words across the boundaries of seed topics to assign a relevance measure to each word in each topic, thus generating a set of topics where each one covers all the words in the vocabulary, as is the case with LDA.
The topics extracted by TExPLAN are used to define an epistemic metric space in which epistemic entities such as words, texts, documents, collections of documents, etc. can be embedded and compared. Once the dimensions are defined, the entities are visualized in two-dimensional space using multidimensional scaling. Because of its generality, the different types of entities can be analyzed jointly in the epistemic space. For this part of the thesis, we demonstrate the capabilities of the approach by applying it to the DBLP dataset, identifying similar conferences based on their locations in the epistemic space and deriving areas of interest associated with each conference. We are also able to analyze the epistemic diversity of conferences and determine which ones tend to attract more diverse authors and publications. Another part of the analysis focuses on authors and their participation in conferences. We define prominent status and answer questions about authors that have this status. We also look at the different ways an author can become prominent, and tie that to their epistemic diversity. Finally, we look at prominent authors who tend to publish documents that are relatively far from the mainstream of the conference in which they were published, and identify authors who may potentially become prominent in the future.
Acknowledgments

Getting to this point has been one of the most challenging things I have faced in my life. This journey has taken much longer than I ever anticipated, but it was filled with joy and excitement. During this journey I met many great people who I would like to express my deepest gratitude to.

First of all, I would like to thank my advisor, mentor and friend Dr. Ali Minai. Dr. Minai has believed in me and supported me from day one. Even during the down times, when my attention was directed towards other things, he kept believing that I had what it takes to finish this. I do not think I will ever be able to repay him for all this.

I would like to thank my committee member Dr. James Uber for his early advice and suggestions. He has been around since the beginning of my PhD research and was always available to discuss ideas. I would also like to thank Dr. Carla Purdy, Dr. Karen Davis and Dr. Raj Bhatnagar for their continuous support and comments to make this work better.

I would also like to thank my dear friend Muhammed Doudine for his never ending encouragement and support. He was always available to discuss research issue during very late hours and his suggestions were very helpful. I will not easily forget the late night phone calls to discuss variations to my proposed solutions and the continuous feedback that has contributed in making
this work better.

During my time at the University of Cincinnati I was extremely lucky to have met some of the finest people around. Unfortunately I will not be able to mention all of them but the ones that stand out are Amr Al Housainy, Byron Thompson, Brijesh Patel, Danial Naithani, Devesh Patel, Faris Alqadah, Fernando Senociain, Hammad Siddiqi, Hassan Al Atat, Mazen Nayfeh, Raed Assaf, Rami Masri, Sandeep Mohan, Shaminder Rai, Vikas Sondhi, and Yahya Abu Ragheb. I would like to thank all of them for their support.

In addition, I would like to thank my fellow high school friends Firas Allan, Khalil Hussain, Muhammed Shehab, and Wissam Batran. They are some of the greatest people I met and I am honored to call them friends.

I am very grateful to have one of my father’s best friends as my friend in Amin Dreidi. His encouragement was always on point and I do not ever remember calling my farther for an update about my studies without him being around to applaud and support.

It is clear that none of this would have been possible without coming to Cincinnati in the first place. Cincinnati would not have been an option without the presence of my family here. I want to thank my aunt Naima and my grandpa Abdel Qader for their endless support and encouragement. They were my family away from home and I will always be grateful to them. I would also like to thank Raed Dakar, for playing an instrumental role in getting me to the US. I would like to thank Kristen Abu Dakar for taking me to the international student office on my first trip to UC; that is where it all started.

My sincere thanks also go to Khalid Daqar, Ibrahim Dakar, Maha Dakar, and Bassam Garadah for helping me to adjust to life in Cincinnati. They were always there for the good and the bad and their advise was certainly crucial.
for me adopting to the life away from home. Maha in particular understood
the difficulty of my situation and she could relate to it better than anyone else. She would keep telling you have to make a final push till I did.

I would also like to thank my cousin and friend Dina who joined UC with me. We celebrated many milestones together and she was always around when I wanted someone to talk to or just have a lunch with. I would also like to thank my cousins Hayfa and Qasem Salameh who were always there when I needed them. I also want to acknowledge Basma, Yasmin, Rima, Dana, Aboud, Lana, Sarah, Sophie, Khalil, Fares, Hadi, Waleed, and Reham who represent the future of my family in Cincinnati. They have definitely made this journey a lot more interesting with their birthday parties and milestone celebrations.

My deepest and greatest gratitude goes to my mom and dad; my true heroes. Words can not describe how much I appreciate having them in my life. They helped me shape the way I think and gave the freedom to explore the world with unconditional support. As I mentioned earlier this journey has taken much longer than expected, and like any project, we sometimes lose interest. My parents were the driving force behind me wanting to see the finish line more than anything. I did this for them more than anything else and I am so grateful to see them content in sharing this success with me. I dedicate this work to them.

Thank you is not enough for my deer sister Abeer who was always there for me. When a lot of people doubted this day will ever come, she was the first to stand behind me and encourage me. If there is one thing I regret, is not being around to spend more time with her. I hope one day we get to make up for the missed time. I am also thankful to her husband Imad Zagha who
was there to take care of her and my parents while I was away. He is the brother I never had. I would also like to acknowledge my nieces Celina and Suzana for bringing joy to my sister and my parents. All of them being happy has contributed to my focus during my studies.

Last but certainly not the least, my wife Lina. She is the true hero behind this who I will always be in debt for. I am sure she learned more about the field of computer science than she ever liked. Having a full-time job and a research to finish is time consuming and mentally draining. She was always there to take care of me and our children Ghanem and Sama. It is clear to me, and to all others around me, that I would not have been able to do it without her support and her unconditional love. She has the toughest job ever and I would never be able to do what she does. God had blessed us with two beautiful kids that made all the struggle of work and research go away in a heartbeat. They are light in our life and without them life would certainly not be the same. My mission is to provide them with the best life possible, and this degree is one of those things.

Thank you all...
## Contents

1 Introduction

1.1 Summary of Research Tasks ........................................ 7
1.2 Domain and Data .................................................... 8
1.3 Research Contributions ............................................. 9
  1.3.1 Topic Extraction through Partitioning of Lexical Associ-
        ative Networks .................................................. 10
  1.3.2 Epistemic Space Embedding and Joint Analysis of Epis-
        temic Entities ................................................... 12
1.4 Thesis Outline ..................................................... 12

2 Background

2.1 Network Analysis ................................................... 16
  2.1.1 Complex Networks Description ................................ 17
  2.1.2 Complex Network Properties ................................... 18
  2.1.3 Network Models ................................................ 21
2.1.4 Centrality Measures ........................................ 25
2.1.5 Community Structure In Networks ....................... 28

2.2 Language Networks ........................................... 36
  2.2.1 Lexical Networks ............................................. 38
  2.2.2 Co-Occurrence Networks ................................. 44

2.3 Vector Space Models ........................................... 45
  2.3.1 Matrices in Vector Space Model ......................... 47
  2.3.2 Measuring Similarity ....................................... 50
  2.3.3 Constructing The Matrices ............................... 52

2.4 Text Clustering ............................................... 55

2.5 Topic Extraction ............................................... 58
  2.5.1 Topic Extraction via Vector Space Models ............. 58
  2.5.2 Topic Extraction via Probabilistic Generative Models . 60

2.6 Text Processing ............................................... 64
  2.6.1 Tokenization ................................................. 64
  2.6.2 Normalization .............................................. 65
  2.6.3 Annotation .................................................. 66

3 Topic Extraction - A Network Based Approach ................ 69
  3.1 TExPLAN Stage I: Term Network Partitioning ............ 72
     3.1.1 Method Description ....................................... 73
     3.1.2 NSF Abstract Dataset Partitions ....................... 77
     3.1.3 DBLP Dataset Partitions ................................. 85
     3.1.4 IJCNN Dataset Partitions ............................... 92
     3.1.5 University of South Florida Word Association Dataset
          Partitions .................................................. 98
3.2 Stage II: Obtaining Overlapping Topics ........................................ 109
  3.2.1 Algorithm Description ......................................................... 109
  3.2.2 Overlapping Topics in the NSF Dataset ................................. 113
  3.2.3 Overlapping Topics in the DBLP Dataset ............................... 116
  3.2.4 Overlapping Topics in the IJCNN Dataset .............................. 121
  3.2.5 Overlapping Topics in the University of South Florida
     Word Association Dataset ......................................................... 124
  3.2.6 Comparison of Topic Overlap in All Datasets ......................... 127

3.3 Comparison with LDA ............................................................... 129
  3.3.1 LDA Results for the NSF Dataset ......................................... 131
  3.3.2 LDA Results for the DBLP Dataset ....................................... 132
  3.3.3 LDA Results for the IJCNN Dataset ..................................... 136

3.4 Conclusion .................................................................................. 142

4 Epistemic Space Analysis ................................................................ 143
  4.1 Overview .................................................................................... 144
  4.2 Embedding in Epistemic Space .................................................... 145
    4.2.1 Embedding Documents ....................................................... 146
    4.2.2 Embedding Authors ............................................................ 147
    4.2.3 Embedding Venues ............................................................. 149
    4.2.4 Epistemic Diversity ............................................................ 150
  4.3 Epistemic Space Analysis ............................................................ 152
    4.3.1 Analysis of Venues in Epistemic Space ................................. 152
    4.3.2 Document Analysis in the Epistemic Space ............................ 159
    4.3.3 Author Analysis in Epistemic Space ..................................... 167
  4.4 Conclusion .................................................................................. 184
5 Conclusion and Future Work 193

5.1 Future Work .......................... 197

Bibliography 201

A A Multi-Agent Model of Evolving Communities and Ideas 233

A.1 Goals and Methods .................. 234
A.2 Semantic Networks and Ideas ........ 237
   A.2.1 Agent Model ..................... 238
   A.2.2 Communication and Idea Formation .... 238
   A.2.3 Social Network Formation Mode ...... 239
A.3 Community Identification Model ........ 241
   A.3.1 Idea Extraction ................. 242
   A.3.2 Results .......................... 242
   A.3.3 Experiment 1: Social Dominant .... 243
   A.3.4 Experiment 2: Mixed ............. 244
   A.3.5 Experiment 3: Semantic Dominant ... 244
   A.3.6 Degree Distribution .............. 246
   A.3.7 Community Size Distribution ...... 246
   A.3.8 Random Interactions ............. 247
A.4 Discussion .......................... 248
A.5 Conclusion .......................... 251

B Behavioral Classification of Agents on Facebook 253

B.1 Background .......................... 254
B.2 Data sets: ............................ 254
B.3 Community Extraction .................. 255
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.4 The Devotion Measure</td>
<td>256</td>
</tr>
<tr>
<td>B.5 Agent Clustering</td>
<td>257</td>
</tr>
<tr>
<td>B.6 Neural Network Classifier</td>
<td>259</td>
</tr>
<tr>
<td>B.7 Interpretation of the Classes</td>
<td>260</td>
</tr>
</tbody>
</table>
List of Figures

3.1 Term network of the NSF dataset ................. 78
3.2 NSF dataset term network properties ................. 79
3.3 Terms with high clustering coefficient values in the NSF term network 80
3.4 NSF dataset term network centrality measures ............. 81
3.5 NSF dataset terms with high centrality values ............. 82
3.6 Term network of the NSF dataset with community labeling ..... 83
3.7 NSF Dataset Topics ........................................ 84
3.8 Term network for the DBLP dataset .................... 86
3.9 DBLP dataset term network properties ................. 87
3.10 DBLP terms with degrees in the second peak of the degree distribution 88
3.11 DBLP dataset term network centrality measures ............. 89
3.12 DBLP Terms with High Centrality Values ................. 90
3.13 Term network of the DBLP dataset with community labeling .... 90
3.14 DBLP dataset topic word clouds ....................... 91
3.15 Term network of the IJCNN dataset .................... 94
3.16 IJCNN dataset term network properties .................. 95
3.17 IJCNN dataset term network centrality measures ............. 96
3.18 IJCNN Terms with High Centrality Values ................. 97
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.19</td>
<td>Term network of the IJCNN dataset with community labeling</td>
<td>97</td>
</tr>
<tr>
<td>3.20</td>
<td>Word clouds for IJCNN dataset topics</td>
<td>98</td>
</tr>
<tr>
<td>3.21</td>
<td>Term network of the University of South Florida word association norms dataset</td>
<td>100</td>
</tr>
<tr>
<td>3.22</td>
<td>University of South Florida word association norms term network properties</td>
<td>102</td>
</tr>
<tr>
<td>3.23</td>
<td>University of South Florida word association norms term network centrality measures</td>
<td>103</td>
</tr>
<tr>
<td>3.24</td>
<td>Word clouds for high centrality words in the University of South Florida word association norms term network</td>
<td>104</td>
</tr>
<tr>
<td>3.25</td>
<td>Term network of the University of Florida word association dataset with community labeling</td>
<td>104</td>
</tr>
<tr>
<td>3.26</td>
<td>Wordclouds topics 1-16 extracted from the University of South Florida word association norms dataset</td>
<td>106</td>
</tr>
<tr>
<td>3.27</td>
<td>Wordclouds for topics 17-32 extracted from the University of South Florida word association norms dataset</td>
<td>107</td>
</tr>
<tr>
<td>3.28</td>
<td>Wordcloud for topic 33 extracted from the University of South Florida word association norms dataset</td>
<td>108</td>
</tr>
<tr>
<td>3.29</td>
<td>Wordclouds for overlapping topics extracted from the NSF dataset</td>
<td>113</td>
</tr>
<tr>
<td>3.30</td>
<td>Word entropies for the NSF dataset</td>
<td>114</td>
</tr>
<tr>
<td>3.31</td>
<td>Entropy value vs. hub for terms in the NSF dataset term network</td>
<td>114</td>
</tr>
<tr>
<td>3.32</td>
<td>Node property values vs. entropy for terms in the NSF dataset term network</td>
<td>116</td>
</tr>
<tr>
<td>3.33</td>
<td>Word-clouds for overlapping topics extracted from the DBLP dataset</td>
<td>117</td>
</tr>
<tr>
<td>3.34</td>
<td>DBLP dataset word entropies</td>
<td>118</td>
</tr>
<tr>
<td>3.35</td>
<td>Entropy Distribution for NSF and DBLP datasets</td>
<td>118</td>
</tr>
</tbody>
</table>
3.36 Cumulative percentage of words with entropy values up to specific levels in the NSF and DBLP datasets ........................................ 119
3.37 Entropy value vs. hub for terms in the DBLP dataset term network .................................................. 119
3.38 Node property values vs. entropy for terms in the DBLP dataset term network ........................................ 120
3.39 Wordclouds for overlapping topics extracted from the IJCNN dataset .................................................. 122
3.40 IJCNN dataset word entropies ................................................................. 122
3.41 Hub value vs. entropy for terms in the IJCNN dataset term network .................................................. 123
3.42 Node property values vs. entropy for terms in the IJCNN dataset term network ........................................ 124
3.43 University of South Florida dataset word entropies ................................................................. 125
3.44 Hub vs. Entropy for Nodes in the USFWAN Dataset Term Network ........................................ 126
3.45 Node property values vs. entropy for terms in the USFWAN dataset term network ........................................ 127
3.46 Cumulative Percentage of Words with Entropy Values for All Datasets 128
3.47 Determining the number of topics of LDA for the NSF dataset .................................................. 131
3.48 NSF LDA Topics K=5 ................................................................. 132
3.49 Determining the Number of Topics of LDA for the DBLP .................................................. 133
3.50 DBLP LDA Topics K=3 ................................................................. 133
3.51 DBLP LDA Topics K=5 ................................................................. 134
3.52 DBLP LDA Topics K=9 ................................................................. 135
3.53 DBLP LDA Topics K=5 for all the words in the corpus .................................................. 137
3.54 Determining the Number of Topics of LDA for the IJCNN .................................................. 138
3.55 IJCNN LDA Topics K=6 ................................................................. 138
3.56 IJCNN LDA Topics K=13 ................................................................. 139
3.57 IJCNN LDA Topics K=17 ................................................................. 140
4.1 Distances distribution of the cosine distance between the ATP-DB and ATP-TB for all the authors ........................................ 148
4.2 Number of Documents vs The Distance between the two Author Topic Profiles .................................................................. 149
4.3 Topic Profiles for each Conference ..................................................... 153
4.4 Conference Locations in 2-D after applying MDS .......................... 154
4.5 DBSCAN Best $\epsilon$ .................................................................................. 154
4.6 Topic Values for Areas of Interest .......................................................... 158
4.7 MDS for Conference Areas ................................................................. 159
4.8 Conference to Documents Distance Distribution ........................ 160
4.9 Area of Interest to Documents Distance Distribution .................. 161
4.10 Document Diversity Distribution for Areas of Interest .................. 163
4.11 Author Properties of Far Documents .................................................. 164
4.12 Far Documents Distance vs. Authors’ Number of Publications ...... 166
4.13 Document Diversity of Close and Far Documents ............................ 166
4.14 Authors Statistics .................................................................................. 168
4.15 Global Author Entropy vs. Average Publication Entropy for authors 169
4.16 The difference between GAE and APE vs. number of publications for authors .............................................................. 169
4.17 Global Author Entropy and Average Publication Entropy vs. publication statistics for authors ............................................. 170
4.18 Global Author Entropy and Average Publication Entropy vs. publication statistics for prominent authors .............................. 171
4.19 Entropy distributions for the top 10 authors in the DBLP dataset 173
4.20 Entropies for the top 10 authors vs. publication statistics ............... 174
4.21 Top 10 authors’ topic profiles .............................................................. 175
4.22 MDS epistemic space views of publications for Author 15142 . . . 176
4.23 MDS epistemic space views of publications for Author 19495 . . . 176
4.24 MDS epistemic space views of publications for Author 2346 . . . . 177
4.25 MDS epistemic space views of publications for Author 700 . . . . 177
4.26 MDS epistemic space views of publications for Author 3137 . . . . 178
4.27 MDS epistemic space views of publications for Author 22593 . . . 178
4.28 MDS epistemic space views of publications for Author 10019 . . . 179
4.29 MDS epistemic space views of publications for Author 9300 . . . . 179
4.30 MDS epistemic space views of publications for Author 4930 . . . . 180
4.31 MDS epistemic space views of publications for Author 1395 . . . . 180
4.32 Author-conference distance distribution . . . . . . . . . . . . . . . . 182
4.33 Author-Conference AoI distance distribution . . . . . . . . . . . . 183
4.34 Author entropy distribution in conferences . . . . . . . . . . . . 184
4.35 Author entropy in areas of interest . . . . . . . . . . . . . . . . . . 185
4.36 Publication entropies for prominent authors with atypical publications 186
4.37 Topic profiles for prominent authors with atypical publications . . 187
4.38 Publications in the Space for Author 11198 . . . . . . . . . . . . . 188
4.39 Publications in the Space for Author 20456 . . . . . . . . . . . . . 188
4.40 Publications in the Space for Author 8725 . . . . . . . . . . . . . . 189
4.41 Publications in the Space for Author 3114 . . . . . . . . . . . . . . 189
4.42 Publications in the Space for Author 7365 . . . . . . . . . . . . . . 190
4.43 Publications in the Space for Author 3283 . . . . . . . . . . . . . . 190
A.1 Three embedded ideas, ABC, BDE, and CEF generate a latent idea BCE(shown in solid blue) . . . . . . . . . . . . . . . . . . . . . . . . . 238
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.2</td>
<td>The initial social network, with agents organized in a small-world formation</td>
<td>240</td>
</tr>
<tr>
<td>A.3</td>
<td>The social network produced by Experiment 1. Different communities have different colors. Agents shown in black are not part of any community</td>
<td>243</td>
</tr>
<tr>
<td>A.4</td>
<td>The social network produced by Experiment 2. Different communities have different colors. Agents shown in black are not part of any community</td>
<td>244</td>
</tr>
<tr>
<td>A.5</td>
<td>The social network produced by Experiment 3. Different communities have different colors. Agents shown in black are not part of any community</td>
<td>245</td>
</tr>
<tr>
<td>A.6</td>
<td>Degree distributions for the networks generated by the three experiments</td>
<td>246</td>
</tr>
<tr>
<td>A.7</td>
<td>Community size distribution generated by the experiments</td>
<td>247</td>
</tr>
<tr>
<td>A.8</td>
<td>Final social network produced with purely random interactions among agents</td>
<td>248</td>
</tr>
<tr>
<td>A.9</td>
<td>Number of novel ideas with respect to the social weight for choosing target agents</td>
<td>250</td>
</tr>
<tr>
<td>A.10</td>
<td>Community homogeneity with respect to the social weight for choosing target agents</td>
<td>251</td>
</tr>
<tr>
<td>B.1</td>
<td>Feature vectors for 20 representative members of each class.</td>
<td>258</td>
</tr>
<tr>
<td>B.2</td>
<td>(a) Dendogram resulting from the clustering process; (b) Number of agents placed in each cluster; (c) Distribution of network clustering coefficient for each class; (d) Distribution of node degree for each class</td>
<td>258</td>
</tr>
</tbody>
</table>
B.3 Confusion matrices for the neural network classifier: (a) Data used
to train the network; (b) Data used to check for generalization during
training but not used for training directly; (c) Data not used during
training at all; (d) All data in B6. . . . . . . . . . . . . . . . . . . . . . . . . . . . 261

B.4 Classes of members based on neural network classifier . . . . . . . 261

B.5 Comparison of degree distribution . . . . . . . . . . . . . . . . . . . . . . . . . . 262

B.6 Comparison of clustering coefficient distribution . . . . . . . . . . . . . 262

B.7 Direct neural network classifier validation on a subset of B12 data . 262
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Top 5 terms in each topic for the NSF data set</td>
<td>84</td>
</tr>
<tr>
<td>3.2</td>
<td>Top 5 terms in each topic for the DBLP dataset</td>
<td>92</td>
</tr>
<tr>
<td>3.3</td>
<td>Topic Interpretation for the DBLP dataset</td>
<td>92</td>
</tr>
<tr>
<td>3.4</td>
<td>Top 5 terms in each topic for the IJCNN dataset</td>
<td>95</td>
</tr>
<tr>
<td>3.5</td>
<td>Top 5 terms in every cluster for the University of South Florida word</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>association norms dataset</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>The resulted clusters from applying DBSCAN to the conferences in</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>the DBLP dataset with $\varepsilon = 0.025$</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>The resulted clusters from applying DBSCAN to the conferences in</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>the DBLP dataset with $\varepsilon = 0.029$</td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td>The interpretation of clusters resulted from applying DBSCAN on</td>
<td>157</td>
</tr>
<tr>
<td></td>
<td>the DBLP dataset</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The growth of on-line textual content has exploded in recent years, creating truly massive text corpora. For instance, between 2009 and 2014, the content shared on Facebook by users has grown from 2.5 billion to 70 billion pieces. In the first eight months of 2015, users on Twitter shared over 203 billion tweets [2], and in one month during the the FIFA 2014 World Cup, 672 million tweets were generated [3]. As the quantity of text available on-line increases, professionals from different industries such as marketing and politics are realizing the importance of extracting useful information and insights from this treasure trove of data. It is also clear, however, that doing so requires methods that go beyond those developed for classical data processing or even natural language processing. In particular, there is great need for efficient methods that can make sense of the semantic content of this data, and allows new knowledge to be inferred from it.

In many cases, it is useful to go beyond just looking at the textual data, and to also consider the authors generating it and the communities in which they are embedded. For example, collections of scientific publications pro-
vide immense opportunities for the automated organization of knowledge and even the discovery of new ideas. However, this requires a systematic way of extracting and representing the knowledge inherent in the documents, tracking the flow of ideas, and identifying the authors who are the primary sources of innovation in the field of study. Among other things, such analysis can be used to identify factors underlying innovation and creativity in individual authors and groups of collaborators, and perhaps provide better metrics for evaluating the quality of their work than simply counting citations or measuring impact factors.

Though the research presented in this thesis focuses on a few specific topics, it is part of a larger effort to understand how new ideas emerge through the interaction of existing ideas within communities of thinking individuals. Until recently, this process could only be studied through speculative models or using relatively small-scale experiments with human subjects, but the availability of very large text corpora such as journal archives, social network feeds, etc., has now created exciting new opportunities to study the ecology of ideas.

Research has shown that knowledge is seldom distributed uniformly over a human network, but rather self-organizes into epistemic clusters of preference, expertise or belief, leading to the generation of different ideas in different communities. The pattern of communities and the distribution of ideas co-evolves in response to the behavioral choices made by the agents in the network. One especially interesting – and relatively well studied – example of social networks mediating the creation of new knowledge are communities of researchers. The availability of publications as well as citation and collaboration information makes it possible to trace ideas and patterns of social
interaction in these groups easier than in other social networks. This has led to numerous studies [155, 217, 13, 126], but even here many open questions remain.

The notion of the co-evolution of ideas and communities can be captured succinctly in two principles:

- **Communities shape ideas**: Agents in a community generate new ideas based significantly on the patterns of interaction with other agents in the community and the distribution of knowledge across these agents.

- **Ideas shape communities**: The ideas expressed by individual agents change the patterns of connectivity across the system by influencing the interaction behaviors of other agents, leading to the continual reorganization of communities.

The research in this thesis is a part of a long-term effort to understand this co-evolution of ideas and communities in large human networks, obtaining insight into what sorts of ideas could emerge from different types of communities, and learning how to structure more innovative, productive communities and organizations, and better social networking applications. A preliminary computational model for this has been implemented using a multi-agent approach [76] (see Appendix A).

The model produced several interesting insights, including the following:

- The tendency for new ideas to arise in a community depends strongly on the *epistemic diversity* of the community, and the behavioral choices made by its agents when they choose whom to interact with.
• The process of communication and behavioral adaptation always results in the emergence of several classes of agents, whose presence, in turn, shapes the epistemic and social structure of the system in the future.

These insights from the preliminary model suggest that, in order to understand the co-evolution of communities and ideas meaningfully, it is necessary to develop a framework that will allow us to characterize three different entities: the content, the users who generated it, and the venues in which they shared it. It is also important to characterize the diversity of ideas at each of these levels, i.e., whether the content covers a broad range of topics, whether the work produced by an individual author spans many topics, and whether the venues or communities where the work is presented are diverse or narrowly focused. In this thesis, we present an approach called TExPLAN, based on the idea of defining an epistemic space i.e., a metric space in which epistemic entities such as words, texts, documents, authors, journals, etc. can be embedded and compared. Since it can embed all these types of entities, this epistemic space allows analysis of each type of entity as well as the relationship between entities of different types within a unified framework.

Defining an epistemic space requires the specification of its dimensions, which should correspond to distinct dimensions of meaning. A natural way to do this is to think of these dimensions as distinct topics. Every epistemic entity can relate to one or more topics, and can thus be placed in the space defined by these topics in a meaningful way. Entities that relate to topics in similar ways will end up closer to each other in this space, and those that have different topic profiles will be placed further apart. This, the topics act
as features in defining the epistemic space.

Extracting latent semantic structure – such as topics – from document collections is not a novel problem. This problem was first tackled by the information retrieval community seeking to improve the results of matching documents to queries [49]. The main focus here was on representing document collections as term-document matrices and applying mathematical processes to extract the eigenstructure of these matrices. More sophisticated approaches known as probabilistic topic models (PTM) were proposed to improve on the early methods in extracting the latent structures. PTM techniques have been dominating this area of research in recent years. In general, PTM methods extract topics by clustering documents based on word occurrences, and these topics can then be used to search for and to cluster documents. Although different methods have different assumptions and approaches, the common denominator is that all of them model documents as mixtures of latent topics [84]. Although PTM methods have demonstrated very good results, they do have some drawbacks. One major issue with these methods is scalability. It is expensive to expand these methods to large datasets. It is also challenging to re-adjust the topics as the dataset grows without having to rerun the method again on the entire dataset. PTM algorithms also do not provide much cognitive insight, since the topics are extracted through purely statistical correlations between words in the context of documents. A major motivation for the work done in this thesis to develop a more efficient, scalable and cognitively grounded approach to extract topics from text corpora. It is important, however, that this new approach maintains or even improves the quality of the topics extracted by other PTM methods like Latent Dirichlet Allocation (LDA).
Languages are very complex things to understand and model. However, languages are used to express ideas, and it is reasonable to postulate that ideas are based on association between simpler ideas and elementary concepts. The mind is, ultimately, built on associations, and that is the fundamental basis on which knowledge is organized in the brain. Indeed, the basic neural mechanism of learning – long-term synaptic modification – is fundamentally an associative mechanism [26, 114].

Thus, the most elementary level at which linguistic knowledge can be represented is as associations between words, or lexical networks. While such networks cannot represent all the knowledge in a mind or a text, they do capture a lot of essential relationships implicitly. They also tell us a great deal about how knowledge is organized in an individual’s mind, and how their thinking can be influenced by external priming using words [45, 141, 130]. This is why considerable effort has been expended on discovering which words people typically associate with each other in several languages [145, 122, 123, 140, 99] and characterizing the structure of the resulting lexical networks [142, 191, 203, 15, 48, 129, 128, 54, 97].

It is also known that useful category information can be extracted from lexical networks [161, 5]. Given these facts, it is natural to consider lexical networks as the basis of discovering the epistemic structure of linguistic data. The approach developed in Chapter 3 of this thesis does so by building lexical networks from text corpora and analyzing its modular structure to identify the natural topics latent in the data.

Once the dimensions of the epistemic space are defined as topics extracted from the text, the next step is to embed authors, documents and venues by determining their participation in different topics. This allows ques-
1.1. SUMMARY OF RESEARCH TASKS

tions such as the following to be addressed in a systematic way:

- Who knows what?
- Who are the prominent authors in a specific area?
- What are the different paths the authors follow to reach a prominent status in different areas?
- How epistemically diverse are certain venues?
- Which authors are more epistemically diverse in their work than others?
- Are epistemically more diverse authors more or less prolific than others?
- What parts of epistemic space are well covered by authors and venues?
- What parts of epistemic space have a lot of authors but no venues, suggesting areas for a new conference or journal?

Chapter 4 of this thesis described how the various types of entities can be embedded in epistemic space, and looks at a few questions of the type listed above in order to demonstrate the general utility of the approach.

1.1 Summary of Research Tasks

To summarize, the research described in this thesis focuses on three primary tasks:

- Developing an efficient, scalable and cognitively motivated approach to extract latent topics from large text corpora.
• Using the topics to define an epistemic space where entities like authors, documents, conferences, journals, books, etc. can be embedded. The axes of this space are epistemically independent dimensions, which can be identified with the topics extracted in Task 1.

• Analyzing the characteristics of documents, authors, venues, etc., by embedding them in the epistemic space, and discovering interesting relationships between these entities based on this analysis.

1.2 Domain and Data

The work in this thesis focuses primarily on datasets of scientific publications. Such datasets are widely available and well studied, which allows us to compare our methods to other methods and understand the quality of the results. We currently have access to three datasets that differ in degree to which the documents comprising these datasets address epistemically distinct topics, and one dataset built from word associations by participants in a study at the University of South Florida.

The first data set is the DBLP Publication dataset [51]. This comes from a set of research papers published in 20 conferences from 4 different areas in Computer Science. It contains 28,569 papers which were authored by 28,702 authors. Each document in the dataset is represented only by a set of discrete words, i.e., no syntactic information is available. The second dataset is the NSF Abstracts dataset [51]. This dataset contains 129,000 abstracts from proposals which were authored by 9,989 authors. The proposals were submitted for grants in 10 research areas between the years 1990 and 2003. The third dataset comes from 2021 abstracts over several meetings of the
International Joint Conference on Neural Networks held during the period 2000 to 2011. Author information is not available for this dataset. The forth dataset used in this research is the University of South Florida Word Association Norms dataset (USFWAN) [147]. The USFWAN dataset is the largest dataset of free associations between words in the English language. The dataset was created by more than 6,000 participants who were asked to write down the first word that comes to mind when given a stimulus word. A total of more than 750,000 such queries using 5,019 stimulus words were used to obtain the data.

The four datasets described above vary significantly in the degree of topic separation. For instance, the topics in the NSF datasets are quite disjoint because the proposals were submitted to 10 distinct programs, whereas in the DBLP and IJCNN cases, the topics have much more potential overlap. The USFWAN dataset is different from the other three because it is not based on documents at all, but is derived from direct experimental probes of associations. We use the USFWAN dataset to demonstrate how the method proposed in this research is applicable to datasets that are not built from text corpora – something that is not possible with document-based methods like LDA.

1.3 Research Contributions

This research makes two main contributions as discussed below:
1.3.1 Topic Extraction through Partitioning of Lexical Associative Networks

Probabilistic topic modeling approaches are statistical methods that assume statistical distributions of words and topics in documents. In an effort to construct a more cognitive and scalable approach, we propose a method that identifies latent structures (topics) through the application of community extraction techniques on associative networks of words. Since humans represent knowledge in terms of associations, it is asserted that deriving topics from associative networks represents a more cognitively meaningful approach than using purely statistical patterns.

The topic identification method proposed in this thesis is called *Topic Extraction through Partitioning of Lexical Associative Networks* (TExPLAN). It begins by constructing a word co-occurrence network from a given text corpus, with nodes representing words and edges between two nodes indicating the event of two words appearing in the same document. The co-occurrence event can also represent the event of two words appearing in the same sentence if the dataset has that level of detail, but this is not the case for the datasets used in this thesis. The weights of the edges in the term network represent the frequency of the co-occurrence event, i.e., the number of times the two words appear in the same document. Once the word network is constructed, the method proceeds in two stages. In the first stage, a partitioning of the word network takes place using the Louvain community extraction method [27] to extract disjoint *seed topics*. The Louvain algorithm works by optimizing the modularity of partitions in the network, where modularity measures how strongly the nodes within a community are connected internally.
relative to their connections to nodes in other communities. This method was chosen because it is scalable and fast. Modularity optimization also provides a natural way of determine the number of partitions (seed topics), whereas in methods like LDA a priori knowledge about the number of topics is required. The seed topics are defined by non-overlapping groups of words, which means that each word belongs to one and only one topic. The significance of each word in its seed topic is determined by its eigenvector centrality measure calculated using the \textit{Hyperlink-Induced Topic Search} (HITS) framework \cite{HITS} that also underlies Google’s PageRank algorithm \cite{PageRank}. The second stage of TExPLAN uses the connectivity of words across the boundaries of seed topics to assign a relevance measure to each word in each topic, thus generating a set of topics where each one covers all the words in the vocabulary, as is the case with LDA.

To determine the quality of TExPLAN, the extracted topics from three datasets are compared to topics extracted by LDA. The results indicate that the quality for TExPLAN topics is comparable to LDA or somewhat better. In some cases, TExPLAN was able to extract topics which LDA was not able to find when run with similar number of topics. LDA requires the number of topics as an input, which is a liability. This is not required in the case of TExPLAN because it extracts the seed topics naturally when modularity is optimized in the word network. When comparing the run-time for TExPLAN which was implemented as a non-compiled system and a non-complied version of LDA using a lemmatizer \cite{lemmatizer}, TExPLAN was faster than LDA. TExPLAN was also able to extract topics from the University of South Florida Word Association Norms (USFWAN) dataset which was not based on documents.
1.3.2 Epistemic Space Embedding and Joint Analysis of Epistemic Entities

The topics extracted by TExPLAN are used to define an epistemic metric space in which epistemic entities such as words, texts, documents, etc. can be embedded and compared. Once the dimensions are defined, the entities are visualized in two-dimensional space using multidimensional scaling. The different types of entities are analyzed in the epistemic space. For this part of the thesis, we demonstrate the capabilities of the approach by applying it to the DBLP dataset, identifying similar conferences based on their locations in the epistemic space and deriving areas of interest associated with each conference. We are also able to analyze the diversity of conferences and determine which ones tend to attract more diverse authors and publications. Another part of the analysis focuses on authors and their participation in conferences. We define prominent status and answer questions about authors that have this status. We also look at the different ways an author can become prominent, and tie that to their epistemic diversity. Finally, we look at prominent authors that tend to publish documents that are relatively far from the mainstream of the conference in which they were published, and identify authors who may potentially become prominent in the future.

1.4 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 contains some background information about the topics discussed in this research. Chapter 3 presents the Topic Extraction through Partitioning of Lexical Associa-
tive Networks (TExPLAN) method and compares its results to those obtained through LDA. Chapter 4 discusses the epistemic space and presents the results of analysis based on epistemic space embedding. Finally, chapter 5 summarizes the work done in this thesis and touches on future work.
Chapter 2

Background

In this thesis we aim to develop a unified epistemic space in which different types of entities can be embedded in order to analyze them. The motivation is that such a space will allow us to answer questions about single and multiple entities and gain a better understanding of them. The dimensions in the epistemic space represent topics which are extracted from text corpora. Topic extraction has been studied extensively in recent years for many reasons. The availability of text data has grown tremendously in the past decade, and being able to design automated techniques to understand and represent text has become more important than ever. The work in this thesis describes a network-based approach for topic extraction from text corpora. The motivation is to have a more cognitive approach than a purely probabilistic one, and an approach that is more scalable.

The work in this thesis draws upon several areas of research ranging from complex networks to associative networks of words. As networks are central to this thesis, the first section in this chapter provides a summary of relevant ideas and concepts from the field of complex networks. The second section
focuses on linguistic networks and their applications. The third section describes vector space models of language, and the fourth section discusses text clustering. We close the chapter by looking at different methods that are used to process raw text, and put it into a format which can be used by different algorithms.

2.1 Network Analysis

The need to extract information from networks arises in many contexts, such as social network analysis [222, 73], the study of metabolic pathways [92], transportation models [19] and semantic networks [184, 104]. Mining network data can be grouped into two general areas: static network mining and dynamic network mining. In the static network mining case, the focus is on a snapshot of the network at a certain time, and the goal is to extract structural properties of the links and the nodes in the network. The work of Watts and Strogatz in [222], and Newman et al. [156] are examples of network data mining on static networks. In mining dynamic networks, the focus is on understanding the evolution of the network and the processes that lead to the observed structures. The work by Barabási et al. in [16] is considered to be a seminal example on which other many other efforts built further [21]. The focus of this research is on static networks. However, the future applications of this work can touch on dynamic evolving networks as well. In this section we aim to discuss general network concepts, and then focus on techniques and metrics for the structural analysis of networks.
2.1. NETWORK ANALYSIS

2.1.1 Complex Networks Description

Following general practice, a network is represented as a graph \( G = (V, E) \) with a set of nodes \( V \) where \( V = |V| \) is the number of nodes, and a set of edges \( E \) where \( E = |E| \) is the number of edges. The terms “node” and “vertex” are used interchangeably to indicate elements of the set \( V \), and the terms “edge”, “link” and “connection” to refer to elements from the set of edges \( E \). When the network in question is a directed network, the links are called ”arcs”. Edges in networks can be binary, where they represent the existence of the relationship between the nodes, or they could be weighted where the have a value associated with the relationship. If node \( i \) and \( j \) are connected with a link, they are adjacent (neighbors). The neighborhood of node \( i \) is defined as follows:

\[
N_i = \{ v_j : e_{ij} \in E \land e_{ij} \in E \}
\]

In a directed network the neighborhood relationship is not symmetric. If the link goes from \( i \) to \( j \), then \( j \) is a neighbor of \( i \). This does not imply that \( i \) is a neighbor of \( j \). A path in a network is defined as a sequence of nodes such that every node in the path has a link to the one after it. The path has two end-points: start, which is the first node, and end which is the last node. The length of the path is defined as the number of hops from the start node until the end node is reached. If the network is weighted, the weights of the edges are taken into account. In this case, the length of the path is the sum of all edges in the path. The distance from the node to itself is 0, and the distance between \( i \) and \( j \) is \( \infty \) if no path exists between them. When a path exists between all pairs of nodes, the network is called a connected network. In the
case of a directed network, when a path exists from each node to all other
nodes, the network is a strongly connected network. A complete network is
a network where the every pair of nodes are neighbors. A subnetwork of
size \( n \), where all nodes are neighbors is known as an \( n \)-clique, and when
such a subnetwork is not a subset of any other subnetwork satisfying these
properties, it is known as a maximal clique.

Another way to represent a graph \( G \) is to use an adjacency matrix \( A \),
where \( A_{ij} = 1 \) if there exists an edge between nodes \( i \) and \( j \), and 0 otherwise.
If edges have weights, these can be incorporated by using a weight matrix
with real-valued entries instead of a binary adjacency matrix.

### 2.1.2 Complex Network Properties

The simplest property of a node is its degree \( k \). The degree of a node is
defined to be the number of edges the node is attached to. In the case of a
directed graph, there are two degrees: in-degree \( k_i \), and out-degree \( k_o \). The
in-degree is the number of arcs that are incident on the node, and the out-
degree is the number of the arcs that go out from the node. If the graph is
not weighted, the degree is the number of neighbors for the node. When the
graph is weighted, a weighted-degree can be defined as the sum of edges
that attach to the node. The degree is a local property of the node. When
researchers want to look at the degree at the network level, they use the
degree distribution. \( p(k) \), defined as the fraction of nodes which have degree
\( k \). This can also be seen as the probability of randomly selecting a node that
has degree \( k \).

A related network property is the average node degree \( \langle k \rangle \) of the network.
This is mathematically defined as follows:
When studying networks, one important aspect to be considered is the lengths of paths between nodes. For each pair of nodes that are connected in the network, we can define the notion of shortest path and its length. As mentioned previously, the path is defined as a sequence of nodes \((v_1, v_2, v_3, \ldots, v_n)\) where between each consecutive node \(v_i\) and \(v_{i+1}\) there is an edge that connects them. In the unweighted case the length of the path is the number of edges along the way, and in the weighted case, the length can be quantified as the sum of weights for the edges along the way. The shortest path between \(i\) and \(j\) in an unweighted networks is the path with the smallest number of edges between the two nodes. In the weighted network case, it is the path with the smallest sum of weights. The length of the path can also be referred to as intervertex distance \(d_{ij}\). From the definition of a fully connected unweighted network, it follows that the shortest path is always of size 1. To get a better idea about how close the nodes are in a network, it is common to consider the mean shortest path length (MSPL), and the shortest path length distribution.

The largest shortest path in the network is called the diameter \((D)\). It is clear that \(D\) is strongly influenced by outliers. A more descriptive metric is the effective diameter proposed by Tauro et al. [208]. The effective diameter is defined as the shortest path by which 90% or more of all connected node pairs can be reached. In mathematical terms, let \(V(l)\) be the fraction of node pairs that have a shortest path of length \(l\). Then the effective diameter \(D\) is the number such that \(V(D - 1) < 0.9\) and \(V(D) \geq 0.9\). It has been found that large real-word networks exhibit small effective diameter values \([8, 9, 135,\) .

\[
\langle k \rangle = \frac{\sum_{i=1}^{V} k_i}{V}
\]
Another important, widely used structural property is the clustering coefficient. The clustering coefficient measures transitivity in networks and is used most in social networks where it answers the questions such as “how many of X’s friends are themselves friends of each other?” In many real-world networks, if there exists an edge between node $u$, and node $v$, and there is an edge between $v$ and $w$, then it is more likely that an edge exists between $u$ and $w$. Clustering coefficient $C_u$ for node $u$ is defined in equation 2.1.

$$C_i = \frac{2|e_{jk} : v_j, v_k \in N_i, e_{jk} \in E|}{k_i(k_i - 1)} \quad (2.1)$$

From this definition, we can derive the clustering coefficient $C_d$ for degree $d$ as the average clustering coefficient $C_u$ for all nodes $u$ of degree $d$. Ravasz and Barabási found that clustering coefficient is significantly lower for random networks than for real-world networks when the degree distribution is similar [180, 179]. Another important observation is that clustering coefficient $C_d$ tends to decrease when the node degree $d$ increases, and $C_d \propto d^{-1}$ [180, 179].

An especially informative measure of the detailed structure of a network is assortativity. This is a degree-degree correlation between neighbors and is usually measured by the Pearson correlation coefficient of their degrees [150, 151]. Assortativity is an informative property of real-world networks as shown by Newman [150, 151]. In his work, he showed that other properties like small-world (see below) can be produced by network models like Erdős-Rényi and Barabási-Albert but not assortativity. Assortativity of disassortativity (negative assortativity) depends on the type of network itself. For instance, biological networks are usually disassortative, while social network
are assortative.

### 2.1.3 Network Models

A significant amount of research has focused on defining network models that can be used to replicate real-world networks. In this part we describe some of different network models that are used in the analysis and understanding of complex networks.

#### Regular Networks

A regular network is a network where all nodes have the same number of neighbors. When nodes have degree \( k \), the network is known as \( k \)-regular \[181\]. Regular networks are not complex networks, but they are used as a comparison to real-world networks to get a better understanding about different properties. A standard class of regular networks are homogeneous lattices defined in a physical space, where each node connects to its \( k \) nearest geographical neighbors. Such networks have high clustering coefficients but also large MSPL.

#### Erdős-Rényi Networks

Random networks were the focus of most of the network research society before complex networks came into their own. The Erdős-Rényi (ER) model \[60, 59\] was the first random network model proposed. The \( V \) nodes in the network are connected by \( E \) edges that are drawn randomly from all possible edges. In this model the probability \( p \) of node \( i \) having a degree \( k_i \) follows a binomial distribution with \( V - 1 \) and \( p \) as parameters.
To determine the degree distribution of the network, we need to look into the number of nodes that have degree $k$ ($V_k$) for all possible values of $k$. We can derive the expected value of $V_k$ from equation 2.1.3 as follows:

$$E(V_k) = VP(k_i = k) = \frac{K}{V - 1} p^k (1 - p)^{V-1-k}$$  

From equation 2.2 we can see that the distribution of $V_k$ approaches a Poisson distribution with mean value equals to equation 2.2. Erdös-Rényi networks typically have small MSPL and low clustering coefficients.

Researchers in the social sciences have realized that real-world networks have some random links, but they follow some sort of natural rule that allows them to evolve to what they are [175, 176, 177]. This has led the community of researchers to develop network models that represent real networks more accurately.

**Small-World Networks**

The proposal by Watts and Strogatz (WS) model [221] of small-world networks is often regarded as the first significant conceptual breakthrough in understanding complex networks. This concept of small-world captures the idea that, though nodes in real-world networks are highly clustered, they are also not far from each other in terms of path length. This was initially discovered by Milgram in his study of acquaintance networks in the United States [135]. Milgram suggested that usually there is a path of length six between most people in the US. This has been known as the “six degrees of separa-
tion” concept, and was the driving force behind the work done by Watts and Strogatz.

The small-world network is characterized by a parameter, $p$, which controls the randomness of the network. To build a network of size $V$, the WS model starts by building a ring lattice of size $V$, where every node is connected to its $k$ nearest geographical neighbors. Then a fraction $p$ of the edges are randomly rewired. When ($p = 0$) the network is a lattice, and when ($p = 1$) the network is completely random. Even fairly small values of $p$ lead to some nodes connecting to distant nodes, thus reducing the MSPL of the network but leaving the clustering coefficient largely unchanged – thus producing the small-world effect.

**Scale-Free Networks**

Scale-Free networks are networks in which the degree distribution follows a power law, i.e., the number of nodes $V$ of a degree $k$ ($V_k$) is given by $V \propto k^{-\gamma}$ where $\gamma$ is the power law degree exponent. Degree distributions of this nature have been identified in many real-world networks, including biological networks [92], social networks [6], phone networks [63], and many other real-world networks. In most of these networks, the value of $\gamma$ takes is in the range $2 \leq \gamma \leq 3$. For instance, Albert and Barabasi has determined that the in-degree distribution exponent of the Internet is 2.1, and the out degree exponent is around 2.4 [8], whereas Faloutsos et al. showed that networks of autonomous systems have a $\gamma \approx 2.4$ [63].

In scale-free networks, there is a small number of nodes that have very high degrees compared to the average node. Such nodes are termed hubs and play important roles in the network depending on the domain. Barabás
and Albert were the first to look into scale-free networks systematically [17], and proposed a plausible model by which growing networks could come to have scale-free architecture. This contrasted with other models, where the nodes were assumed to be fixed a priori. The BA model proposed by Barabás and Albert postulated that new nodes enter the system with an inherent quota of attachment sites, and use these to connect to existing nodes such that the probability of connecting to a node is proportional to the degree of that node [17]. This concept is known as preferential attachment, and it has been used to explain the properties of real networks such as the World Wide Web, and social networks [17]. The model starts with a core number of nodes $V_0$ and then adds a node at every step. Every new node added links to one or more nodes from the existing one. Based on the principle of preferential attachment, when the new node selects a target node, it does so with a probability that depends on the target node’s degree. In mathematical terms, the probability $p_e(j, i)$ of new node $j$ connecting to existing node $i$ is given as:

$$p_e(j, i) = \frac{k_i}{\sum_k k_k} \quad \text{(2.3)}$$

The MSPL in scale-free networks is smaller even than in the ER model for a network of the same size. One important difference between the BA model and the WS model is the fact that the clustering coefficient $C$ in the BA model is independent of the network size, and is usually much smaller than in small-world networks, unless some features are added to the preferential attachment prescription [203].

The BA model has opened the door for many new modifications that were able to produce networks which are closer to real-world ones. Some of these focused on modifying equation 2.3 to a nonlinear one [231]. Other focused
2.1. NETWORK ANALYSIS

on reattaching links dynamically [41]. Ravasz and Barabási have also shown that some scale-free networks with high clustering coefficient tend to produce a hierarchical organization [179]. To model this behavior, they proposed the Ravasz-Barabási model, which creates networks that have hierarchical structures [179]. This model is interesting when looking into extracting hierarchical communities in networks.

2.1.4 Centrality Measures

A significant amount of research in the network field has focused on determining the importance of nodes in networks. Different centrality measures identify different roles played by nodes in the network. In a hierarchical network, the root node is the most central node, but in other cases determining the most important node is not trivial. The research in this area has focused on four main centrality measures:

Degree Centrality

The degree centrality measure is the simplest out of all centrality measures. The notion behind it is that nodes with higher degree are more central. This is often justifiable in both unweighted and weighted networks, since highly connected nodes play the role of hubs in many networks.

Closeness Centrality

In this measure the focus is more on the proximity of the node to all other nodes in the network. A more central node is a node that is closer to other nodes. Two nodes can have the same degree, but their distance to all other
nodes in the network can be different. In mathematical terms, this measure is defined as follows:

\[ C_C(n_i) = \frac{V - 1}{\sum_{j=1}^{V} d(n_i, n_j)} \]  

(2.4)

The ranking in this case is based on the average distance to all other nodes in the network.

**Betweenness Centrality**

Betweenness centrality is one of the most frequently used centrality measures and was first proposed by Freeman [74]. In this case, a node is considered central if it appears on many shortest paths between nodes. In mathematical terms betweenness centrality is defined as follows:

\[ C_B(n_i) = \sum_{j \neq i \neq k} \frac{\delta_{jk}(i)}{\delta_{jk}} \]

where \( \delta_{jk} \) is the number of shortest paths between nodes \( j \) and \( k \), and \( \delta_{jk}(i) \) is the number of shortest paths between \( j \) and \( k \) that pass through \( i \). To be able to compare networks to each other, betweenness centrality values are normalized by dividing the result of equation 2.1.4 by \((n - 1)(n - 2)\).

**Eigenvector Centrality**

Another widely used centrality measure is the eigenvector centrality proposed by Bonacich [28]. The idea behind this centrality measure is to determine the importance of the node by looking at the importance of its neighbors. It is clear that, in most cases, not all connections are equal. For example, on the Internet pages referenced by or linking to important pages tend to be more
important and useful themselves. In a social network, it is more important to be connected to influential people than non-influential ones. Of course the number of connections a node has is important, but when two nodes have the same number of connections, to whom they are connected makes a difference. In mathematical terms:

\[
C_E(n_i) = \frac{1}{\lambda} \sum_{j=1}^{V} A_{ij} C_E(n_j) \tag{2.5}
\]

where \(\lambda\) is a constant, and \(V\) is the number of nodes in the network. If we define all node centralities in a vector form \(C_E = (C_E(n_1), C_E(n_2), ..., C_E(n_V))\) then we can write equation 2.5 in the matrix form:

\[
\lambda C_E = AC_E \tag{2.6}
\]

where \(A\) is the adjacency matrix defined earlier. From equation 2.6 we can see that the \(C_E\) is an eigenvector for \(A\) with eigenvalue \(\lambda\).

Eigenvector centrality is the basis of the famous PageRank algorithm proposed by Page and Brin [32]. Google search engine is built on top of this basic concept and it ranks pages on the Internet on that basis. In directed networks, eigenvector centrality splits up into that based on incoming edges for each node, and that based on outgoing edges. These two types of centrality are termed the authority and hub values of the node [103]. High authority nodes are those to which many important nodes point, while high hub value nodes are those that point to many important nodes.
2.1.5 Community Structure In Networks

So far we have looked at complex networks at two different levels; the node level, and the network as a whole. These two levels are referred to as micro and macro, respectively. At the micro level, we are interested in defining node properties such as degree, centrality, etc., while at the macro level, we look into properties for the whole network such as degree distribution mean shortest path length. There this a third level known as the meso level which focuses on understanding groups of nodes and their interaction. In this section we review some of the work that targets the meso-level attributes of a network.

One of the areas that has attracted a lot of attention in network analysis is community extraction [169, 64, 170, 162, 152]. This is important because communities – or highly connected subsets of networks – often represent meaningful units in networks. In social networks, they correspond to actual communities, in citation networks communities can be seen as areas of interest [57, 47]. In a similar fashion, communities of web pages on the Internet could reflect areas of interest, whereas in semantic/epistemic networks they can represent topics, ideas or categories. The ability to extract groups of nodes, which can be seen as communities, is of a significant value because it also allows us to gain a better understanding about the network structure and dynamics.

Historically, there have been two main approaches to extracting communities in networks. The first approach is known as graph partitioning, and the other as community structure detection or block modeling. The first approach has been used mainly in the computer science field where researchers tried
2.1. NETWORK ANALYSIS

to apply it to VLSI designs and parallel computing issues [58, 71]. The second approach was mainly the focus of physicists, sociologists, and applied mathematicians with the aim to apply it on issues related to biological and social networks [153, 225, 220]. In reality, both approaches try to achieve the same thing, though differences in the application domains end up dictating different technical solutions.

Graph Partitioning

Graph partitioning is the approach used by computer scientists where the goal is to partition the network into \( c \) groups of a roughly equal size where the number of edges between groups is minimized – also called the min-cut problem. For example, one problem is to determine how to group a set of tasks for parallel computing where interprocessor communications is minimized. Such problems were usually approached by dividing the set of nodes into two partitions, and then partitioning the new partitions iteratively. The most cited techniques to solve the problem are the spectral bisection method [70, 172] and the Kernighan-Lin algorithm [101]. The spectral bisection method works by extracting the eigenvalues of the Laplacian of the graph, whereas the Kernighan-Lin algorithm uses a greedy method to optimize the number of edges between the groups. All the algorithms of this nature suffer from the fact that they divide the groups into two subgroups. To get more than two subgroups, one has to divide the subgroups further. Thus, if the best partition of a network is of a size other than a power of 2, these methods fail to find it.

Another technique to solve the network partitioning problem is hierarchical clustering [188]. In this technique a method is developed to measure the
similarity between two nodes based on the network topology. Once a similarity measure is determined, then one can start by connecting nodes starting from the most similar, and then adding the rest in a decreasing order of similarity constrained by a threshold. The resulting network is different from the original one and is based on the similarity that was calculated from the original network. In these algorithms, the starting point is one where every node is its own community, and the last step merges all nodes under one big community. The process is hierarchical, and a dendrogram is produced. A user can then cut the dendrogram at any level and extract the communities. One drawback of this method is that it assumes communities are separated because weak links happen to join them, and does not account for the possibility that two communities are connected with a few very strong links.

One of the first methods to address this problem was the method proposed by Girvan and Newman [79] (GN). The GN method tries to extract natural clusters by eliminating edges between communities, where edges between communities do not have to be weak. The algorithm pictures the problem as understanding the traffic between the nodes in the network. If one needs to move from one node in one community to another node in another community, then one must cross one of the few edges that link the two communities; those edges can be seen as "bottlenecks". The algorithm works by identifying the edges that link communities and removing them. To identify an edge between communities, the algorithm calculates the edge betweenness of all the edges, where the edge betweenness is the number of shortest paths between any pair of nodes that run across the edge. Once edge betweenness is calculated for all edges, the algorithm removes the edge with the highest value, and then recalculates the edge betweenness
again for all edges. The process continues to eliminate all edges, and can be see as a hierarchal process that starts at the root, rather than leaves. Once the dendrogram is created, it can be cut at any level extracting the different communities. The method proposed by Grivan and Newman is more useful than the spectral bisections methods described earlier because it allows extraction of an arbitrary number of communities. It is also more useful than the traditional hierarchical methods because it extracts more natural clusters. However, similar to all the other methods described so far, it does not give us any insight on the number of communities in the network. One disadvantage of this algorithm is its speed. In the worst case this algorithm needs $O(mn)$ time where $m$ is the number of edges, and $n$ is the number of nodes. Tyler et al. [213] proposed a modification to the algorithm that reduces the time in calculating betweenness but has a worse accuracy. It works by calculating a partial edge betweenness by identifying an appropriate sample of the nodes. That way, the algorithm does not have to calculate edge betweenness on all nodes. Radicchi et al. [173] propose a different approach that runs faster, and that is done by calculating loops instead of edge betweenness. Loops are calculated locally, therefore, they reduce the time needed by the algorithm to run. The hypothesis is that edges that link communities do not usually belong to a significant number of short loops.

One common issue with the methods described so far is that they do not shed any light on the number of communities that exist in the network. Clearly such methods are problematic when the problem is identifying the structure of the network, and trying to determine how many subgroups exists.
Community Structure Detection

In the community structure detection approach, the problem of detecting communities in the network is looked at from an analytical perspective. The questions these methods try to answer are: “Are there any communities in the network?” and “If yes, how many?”

To address this problem, Newman and Girvan proposed a modification to the GN method where they evaluate the divisions by a measure called modularity [154]. Modularity measures the statistical organization of edges between communities. The motivation behind this measure is to determine the fraction of the edges in the network that link nodes within the same community. Assuming that a network is divided into $k$ communities, let the matrix $E$ be a $k \times k$ matrix where $E_{ij}$ is the fraction of edges in the network that connect nodes from community $i$ to nodes in community $j$. The fraction of edges in the network that connect nodes in the same community are given by the trace of matrix $E \cdot TrE = \sum_i E_{ii}$. It is clear that the stronger the trace of the matrix, the more community structure there is, but the trace on its own does not give an indication of the natural community structure. To incorporate that into the calculation, Newman and Girvan defined $a_i = \sum_j E_{ij}$ to be the fraction of edges in the network that connect to all the nodes in community $i$. The final definition of modularity is given as:

$$Q = \sum_i (E_{ij} - a_i^2) = TrE - \|E^2\|$$

(2.7)

where $\|E\|$ is the sum of all the elements in matrix $E$. In this definition of modularity, we get the fraction of edges that connect nodes within the same community minus the expected fraction of edges that connect nodes between
different communities. The value of $Q \in [-1, 1]$ where $Q = 0$ when the the fraction between nodes in the same community corresponds to the random case. A value closer to 1 indicates strong community structure.

Given this measure of modularity, the optimal set of communities can be found by maximizing modularity instead of iterative removal of edges[153]. However, optimizing modularity is a very expensive problem computationally. It would take an exponential amount of time to explore the space of all possible divisions and therefore, it is not scalable for large networks. The best alternative to this to use approximation methods like genetic algorithms or simulated annealing. In his work, Newman uses a greedy method that appears to generate good results. The method starts by placing every node in its own community and then merges communities that result in better modularity values. This is a hierarchical approach where a dendogram can be constructed all the way from the leaves to the root. Once the process is done, the algorithm can make a cut in the dendrogram where the value of $Q$ is maximized. In the case of weighted networks, the modularity value $Q$ can be defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[ w_{ij} - \frac{w_i w_j}{2m} \right] \delta(c_i, c_j)$$

where $c_i$ is the community to which node $i$ is currently assigned, $w_{ij}$ is the weight from node $j$ to node $i$, $w_i = \sum_j a_{ij} =$ total incoming weight for node $i$, $m = \sum_i w_i =$ total weight of the network, and $\delta(c_i, c_j) = 1$ if $c_i=c_j$, 0 otherwise.

The fastest method of approximation for modularity optimization was proposed by Clauset et al. [44]. This method merges communities recurrently in a way where the resulting communities optimize modularity. However, the resulting modularity values tend to be smaller than modularity values pro-
duced by simulated annealing [86]. One other drawback of this method is that it tends to generate large communities that contain a large portion of the nodes, thus slowing down the process. The work by Wakita and Trsurumi in [218] defines some workarounds where the sizes of the communities stays small, resulting in a faster algorithm.

Some of the methods we have described so far were applied to large networks. For example, one of the large networks is a protein interaction network where the number of nodes is 30,739 [163]. Another, and much larger network, is a 5.5 million node network from a Japanese social networking site [218]. It is true that these are large networks, but they are still far from networks like Facebook that has around 1.49 billion users, or the 20 billion pages that Google searches every day.

A faster and more scalable algorithm for community extraction was proposed by Blondel et al. [27]. This algorithm builds a hierarchical structure of communities where it merges two communities only if there is a positive gain in modularity. The algorithm starts by assigning every node to its own community, and then for each node $i$ it looks for neighbors $j$ and determines if moving $i$ to the community of $j$ will gain modularity. If the gain in modularity is positive, then $i$ is moved to the same community as node $j$. Once the process converges and no node can be moved with a gain in modularity, the algorithm enters the second phase. In the second phase, a new network is constructed whose the nodes are the communities obtained from the first step. The edge between two newly formed communities, which are nodes in the new network, is the sum of all the edges that link nodes from the corresponding communities. Then the algorithm from step 1 is applied to this network, and the process is repeated recursively for higher hierarchical lev-
2.1. NETWORK ANALYSIS

This algorithm is fast because calculating the gain in modularity when node $i$ is moved to community $C$ can be calculated as follows:

$$
\Delta Q = \left[ \frac{\sum_{in} + k_{i,in}}{2m} - \left( \frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[ \frac{\sum_{in}}{2m} - \left( \frac{\sum_{tot}}{2m} \right)^2 - \left( \frac{k_i}{2m} \right)^2 \right]
$$

where $\sum_{in}$ is the sum of all the edge weights in $C$, $\sum_{tot}$ is sum of all the weights for links affiliated to the nodes in community $C$. $k_i$ is the edge weights sum of all the edges affiliated with $i$, $k_{i,in}$ is the sum of all the edge weights that link nodes in $C$ to node $i$, and $m$ is the total sum of weights for all the edges in the network. This algorithm is simple to implement and extremely fast. It can also be applied to massive networks as shown in [27].

All the algorithms that we have described so far produce non-overlapping communities. In some cases, having the ability to extract overlapping communities is very important. In social networks for example, individuals usually belong to more than one community. Many algorithms have been proposed to extract overlapping communities in network. Palla et al. [163] propose a clique propagation framework, which is used to extract communities from networks such that community overlap is possible. This method uses the notion of $k$-clique to define a clique of size $k$, and two cliques are $k$-connected if they share $k - 1$ nodes. Finally a community is defined as a $k$-clique percolation cluster, which is the union of all $k$-cliques that are $k$-clique-connected.

Another very effective algorithm is link clustering [4], which groups links instead of nodes. The algorithm starts by creating an edge-network for which the nodes are edges in the original network. The nodes in the edge-networks are connected with edges where the weight reflects the similarity between the two edge-nodes. The similarity between two edge-nodes is calculated as follows:
Once the edge-network is constructed, a non-overlapping partition algorithm is applied to extract non-overlapping communities of edge-nodes. This translates into non-overlapping partitions for the edges in the original network. Finally, every edge community is transferred into a node community by simply using all the nodes that are a part of the edges in the corresponding community. Since nodes can have multiple edges, they can belong to multiple communities. The result of this method is an overlapping set of node communities.

In this section, we gave a short overview of complex networks and some of their properties. We described some analytical properties that allow us to gain a better understanding about their topology. We also described an important subject in network analysis known as community extraction. In this work we aim to extract communities of terms from term networks, so it was important to give an overview of the community extraction methods that can help us achieve this goal. In the next section we describe linguistic networks in general and give an overview of some models that are used to construct and analyze them.

### 2.2 Language Networks

Processing textual data has been the major focus of research within many fields, including artificial intelligence, information retrieval, stylometrics, cryptography, psycholinguistics, etc. In recent years, the amount of available text has grown tremendously due to many factors. The emergence of the
Internet and Web 2.0 sites has allowed users to generate even more content. In addition to that, most, if not all, content generated in scientific research has been digitized. This explosive growth in available content has led many researchers to believe that studying languages from an information processing and computational viewpoint is more important than ever. Languages are very complex systems to understand and model. They are made up of a large number of words that can be related in many different ways. Words are sometimes ambiguous and constantly changing. New words appear from time to time, while others die out. Relationships between words can have many dimensions ranging from structural relations to semantic relations. Studying languages gets even more complicated when approaching it from a cognitive perspective. The process by which linguistic content is generated is effected by many aspects like the organization of knowledge in the brain, affective state, and environmental constraints. All this complexity is managed by one of the most complex systems known to us: our brains!

Most of the work that has been done in linguistics and psycholinguistics has focused on understanding this complexity. However, when it comes to understanding the structure of a language, a different approach is required. If we want to know how similar the English language is to Arabic, both need to be represented in a way where we can compare them, identify their similarities, and study their dynamics. It has been proposed that languages can be represented very generally as language networks [7]. The study of language networks can be classified into two categories: (1) the research that focuses on understanding the structure and evolution of language networks, and (2) the application of language networks for automating different tasks such as text summarization, and information retrieval. In this section we aim to give
an overview of the different efforts used to represent languages as networks and understand their characteristics.

### 2.2.1 Lexical Networks

*Lexical networks* is a broad term that covers networks in which nodes are words (lexemes). These networks can vary in size depending on the language and the target of study. To model a lexical network of an individual, we access to their *mental lexicon*, i.e., the words they are familiar with. It has been found that for high school students the size of such a lexicon is over 100,000 words [137]. It is quite fascinating when we consider the speed with which high school students can retrieve words and associate them with others. Understanding how these words are represented and retrieved is a whole area of science on its own right. The work by Collins and Quillian is considered one of the first efforts to study information retrieval time from semantic memory [46]. In their work, they focused on developing a tree structure that captures the relationship between concepts and sub-concepts. For instance, an elephant is a mammal, and mammals are animals. The goal of such studies was to model the local structure of concepts. However, attempts to understand the global structure did not take place until later stages when researchers started to use complex networks as a representation [85, 96, 207, 215, 192].

In the efforts described above, the relationship between words in the network varied from phonological similarity, to parts-of-speech similarity. In this part, we will review examples of such networks.
Lexical Networks based on Phonological Similarity

Phonology is the area that focuses on how sounds are used in languages. When networks are built based on phonological similarity, two words are similar if they sound similar. For example, the word coast will be similar to the word ghost, whereas bear and elephant will not. Understanding how words are related phonologically is important in studying the mental lexicon and linguistic systems in general [10, 81, 168, 202]. There has been a large body of research that studies phonological relationships of words by modeling them in complex networks. One of the most famous efforts was done by Kapatsinski where he created a complex network known as the Phonological Neighborhood Network (PNN) to study the structure of the mental lexicon. In his network, two words are connected if they share at least two thirds of their phonemes. Kapatsinski built his network from the Hoosier Mental Lexicon [159]. The PNN network has high clustering coefficient and long path lengths. Although high clustering coefficient is a property of small-world networks, however long paths is not. Other efforts in understanding the mental lexicon by modeling phonological based complex networks can be found in [85, 96, 207, 215]

Orthographic Based Lexical Networks

In orthographic networks words are connected by edges that have weights which reflect the similarity in the spelling between the two words [36]. These types of lexical networks have been used to study spelling mistakes and corrections. An example of this is work by Choudhury et al. which analyzes three languages: English, Hindi and Bengali [42]. Their SpellNet system
demonstrated that the average weighted degree can predict the probability of spelling mistakes in the language. The have also shown that these networks have high clustering coefficient and exponential degree distribution.

**Semantics-Based Lexical Networks**

In this class of networks, nodes are words, but the relationship between words is based on semantic similarity. Defining semantic similarity can take multiple forms, and have different usage. Below we discuss two examples of building semantics-based networks.

**Dictionaries and Thesauri** The most well-known semantic lexical networks is WordNet [65]. In networks like WordNet nodes are concepts and the edges between them represent a semantic relationship. Those relationships can be of different types such as:

- synonym e.g., automobile and car
- antonym, e.g., good and bad
- hypernym-hyponym, e.g., mammal and cat
- meronym-holonym, e.g., branch and tree

An interesting issue that these networks can help with is polysemy – when a word such as “bank” or “trunk” has multiple meanings. Looking at the network as a whole, polysemic words represent “short cuts” that give the network a small-world characteristic. It is also found that words with high clustering coefficient are often polysemic.
In other work, complex networks were built from concepts in a thesaurus and analyzed to determine the consistency and quality of the thesaurus itself [132]. For example, to determine the quality of the hypernym-hyponym relationship, the network should locally have a hierarchal structure. On the other hand, a polysemy relationship should produce networks with that are more small-world. Examples of work like this can be found in [192]

**Feature-Based Similarity**  Another way of building semantic networks of words is based on the similarity between their descriptive and functional features. Such features have been used successfully to explain models of memory [131]. One of the most influential efforts in this area was done by McRae et al. in what is known as Feature Product Norms[131]. Their effort was centered around collecting features for words from participants. Participants were given some words, and they were asked to list some characteristics which they believed were important for the words. The result was to build a vector representation for each word, where each entry in the vector represents a feature, and the value represents how many participants had associated the word with that feature. For this network, the number of concepts was 541 and the semantic similarity between them was calculated using the cosine similarity of their feature vectors. The cosine similarity between two words $v$ and $w$ is calculated as follows:

$$sim(v, w) = \frac{v_1w_1 + v_2w_2 + \ldots + v nw_n}{|v||w|}$$

(2.9)

where $v_k$ and $w_k$ are the values of feature $k$ in the words, respectively. This gives a value of 1 for perfect similarity, -1 for complete dissimilarity and 0 when the words are orthogonal in their features. When constructing the
network, each concept is a node, and the edges are links between concepts that have a similarity value greater than zero. The weights of the edges are the cosine similarities between the concept pairs.

Associative Networks In these networks, words are nodes and links are linguistic associations observed from participants in an experiment. Through a long data-collection process, Nelson et al. produced what is known as the *University of South Florida Free Association Norms* (USF-FA) \[146\]. USF-FA is the largest association dataset for the English language and was produced by 5,019 participants. Participants in this study were asked to write down the first word that came to their mind when given a certain word. Such data, collected over all participants, was then compiled to give ordered lists of associates for each of the cue words. The associations can be used implicitly to define a network with words as nodes. This type of a network is different than the one proposed by McRae et al. because it is not based on an explicit set of features, and the links represent association rather than similarity. For example, “hair” and “comb” have no similarity of features, but are strongly associated. The USF-FA network is also more complex because participants can use any relationship that ties words together. Some may chose to write down a word based on a conceptual relationship. For example, when given the word *car*, they may write down *bus*. Others may choose to write down words based on a causal relationships like *sick* and *fever*. Naturally this results in a directed network. However, since the number of cues used is necessarily smaller than the total number of words elicited as associated, the directionality is strongly influenced by the experiment, and researchers have often ignored directionality in the study properties related to these networks.
Association is the primary mechanism for humans to retrieve information from our memory. This has led researchers to investigate the network properties that lead to efficient information retrieval. Two main network properties that play a role in effective association and information retrieval are local clustering and short path lengths. Researchers have described the small-world property of word networks as an indication of efficient information retrieval and management [143]. Local clustering is essential for effective associations, and short path lengths are necessary for fast search in the information space [204, 192].

One of the most comprehensive efforts to understand semantics-based lexical networks was by Steyvers and Tenenbaum [204]. In this work the authors analyzed three large-scale semantic networks built from WordNet, Roget's Thesaurus and the USF-FA norms. They have demonstrated how these networks have a small-world structure and strong local clustering. In addition to that, the networks have power law degree distributions. These are typical properties of other real-world networks such as the World Wide Web. The authors propose that these properties reflect the way the networks evolved, and propose an algorithm as a model for this. Recently, the word on associative networks has been extended by Kenett et al [97], who repeated the FSU association association experiment but allowed subjects to generate multiple associates for each cue.

The majority of the work in understanding semantic networks was done on the English language. However, association networks are available in other languages [37, 66, 134, 67], and researchers have demonstrated that properties of real-world networks such as high clustering coefficient and short path lengths exist in other languages [100].
2.2.2 Co-Occurrence Networks

Co-occurrence networks can be seen as a type of associative networks where words are connected when they appear together in normal usage. Such networks are well studied in an area called collocation analysis \[194, 205, 206]. In collocation analysis the goal is to study semantic similarity based co-occurrence in text. The motivation is that co-occurrence is an example of a semantic association between the words regardless of any grammatical relation. In most cases, networks are constructed by connecting terms that are close to each other in texts. For instance, one way of constructing such a network is by connecting two words if they are \(d\) words apart. For example, if \(d = 3\), then each word in a text is connected to the three previous words, and to the next three words. Defining the value of \(d\) is important because if it is too short, it can miss on important associations, and if it is long, it can associate words that are not really related \[68\].

One of the early efforts to construct such networks was done by Ferrer and Solé where they analyzed the British National Corpus (BNC) \[68\]. Two co-occurrence networks were constructed from the data set: (1) words are connected if they are in the sentence and at the most three words apart, (2) connections between words are considered if they happen to appear together in the first network more often than what is expected randomly. In both networks, the small-word property was apparent. The top 5000 connected words also showed a power law degree distribution. Other researchers built on this and showed that real-world network properties exist in co-occurrence networks for other languages \[29\], and other corpora \[88\]. Dorogovtsev and Mendes utilized these findings to propose a model for network growth that
was able to replicate these properties [53].

In the previous examples, and others [107], the network was constructed from a collection of documents, and used to extract properties about the whole dataset. In other similar work, the network was constructed from single author's documents in an effort to understand their writing style [11].

In this section we gave an overview of linguistic networks and some of their applications. The work discussed in this thesis will build lexical networks from text similar to what is descried in section 2.2.2. The aim is to use word networks to extract communities of words that can be interpreted as topics. In the next section we describe some of the efforts in representing text using a Vector Space Model.

2.3 Vector Space Models

Representing different aspects of languages has been a challenging problem in the field of information retrieval. The complexity of language has imposed many limitations to fully utilizing machines for tasks like search and recommendations. In this section we describe an idea known as the Vector Space Models (VSM) which allows representation of words and documents in such a way suitable for automation.

The intuition behind VSMs is to embed documents in a metric space such that documents that are similar to each other are close in proximity. This concept was first introduced by Salton et al. in a system known as SMART [185]. SMART is an information retrieval system where documents are points in a space, and queries are also points in the same space. When a query is submitted, the system determines its location in the space, and returns a
list of documents (points) based on the distances to the query. The returned
documents are sorted in a decreasing order, thus selecting the most similar
documents first. This basic idea has been the used in many modern search
systems [124].

This type of a representation is desirable because it allows the machine to
extract knowledge without the manual work needed to construct ontologies.
For example, Rapp was able to measure the similarity between words in the
British National Corpus (BNC) without the need of an ontology such as Word-
Net [178]. Ontologies are available for languages like English and Spanish,
but not necessarily for other languages. VSMs provide the ability to measure
word similarities directly from raw text, which is easily available. It has been
shown that VSMs do very well when the task at hand is to determine the
similarity between words, documents, or corpora [124, 164, 178, 211, 144].

Researchers have found VSMs interesting because it aligns well with
what is called the distributional hypothesis of meaning. In this hypothesis,
the assumption is that words which happen to appear in similar documents,
tend to have similar meanings [69, 87, 118, 49]. Most of the algorithms built
on this hypothesis represent words and documents using vectors and matrices.
It is important to clarify that not all models that use vectors and matrices
are considered VSMs. In a VSM, the vectors and matrices typically represent
the frequencies of events. For instance, in a document by term matrix, the
entries represent how many times a word appeared in a document. On the
contrary, in cases where we represent lexical graphs using semantic similar-
ity, etc., are not considered VSMs. though they may be related. It is this idea,
which bases the vectors and matrices on event frequency, that strongly ties
VSMs to the distributional hypothesis.
Examples of using vectors for point representations has been used in many areas such as clustering and classification. They are usually referred to us feature vectors. In most cases, features have nothing to do with frequency. For instance, features for a table could be *is flat* and *has 4 legs*; those do not represent any frequencies. In VSMs, however, frequencies are used as features [229].

Vectors have also been used in recommendation systems where usually the system has a matrix of users as rows, and all items as columns. In this scenario, the user vector is a list of values \((i_1, i_2, ..., i_n)\) where \(i_x\) is the number of times the user bought item \(x\) [182, 31, 117]. Vectors have also been used in cognitive science research in areas such as prototype theory[183, 108]. In idea in this theory is that in some cases, members in categories have a different degree of membership; some are more central than others. In this case, members could be vectors, and a set of members can be seen as categories [158, 197]. Psychometrics is another area where vectors are used to represent questioners [199]. In these examples the matrices and vectors are not VSM because the are usually not built from event frequencies. In the next section we discuss methods like Latent Semantic Analysis (LSA) [49, 109] and Hyperspace Analogue to Language (HAL) [121, 120] that are built on VMSs. In fact, it has been shown that, for theoretical and empirical reasons, systems like HAL and LSA are good models for human cognition [111].

### 2.3.1 Matrices in Vector Space Model

Researchers have suggested that it is possible to infer meaning from vectors of word usage frequencies and other text statistics. This is known as the
statistical semantics hypothesis and was first introduced by Warren Weaver in 1955 [223]. This hypothesis was the driving force behind other hypotheses that motivated different matrix representations in VSM, which are described below.

The Term-Document Matrix

The purpose of the Term-Document matrix \( X \) is to represent the occurrence of terms, i.e., words, in the vocabulary in every documents of a corpus to determine document similarities. Each row in \( X \) represents a term vector, and each column represents a document vector. The entry \( X_{ij} \) represents the number of times word \( i \) appeared in document \( j \). Each column vector is built from the bag of words that appear in it. The term bag is derived from mathematics where a bag is defined as a set that allows an element to appear more than once (multiset). Aggregating this bag to identify the counts of each elements creates a vector where each value is the number of times a certain element appeared. For instance bag \( b = \{ w_1, w_1, w_1, w_2, w_3, w_3, w_3 \} \) contains three distinct elements \( w_1, w_2, \) and \( w_3 \). For bag \( b \) we can derive vector \( x = \langle 3, 1, 2 \rangle \) by putting the frequencies of element \( w_1, w_2 \) and \( w_3 \) in this order. Note that the order of the entries in the set does not matter, but the same order must be used for all documents in the corpus. Essentially, each document is placed in a vector space whose dimensions are defined by the words in the vocabulary.

Assuming that \( x \) represents the bag of words for a document, a set of all the bags of words for all the documents in the corpus can be seen as the matrix \( X \). If the document collection has \( n \) documents and \( m \) words, then the matrix is \( X = m \times n \), and \( X_{ij} \) is the frequency of word \( i \) in document \( j \). In
general, \( X \) is usually a very sparse matrix and most of its entries are zeros.

This concept of bags ties in with a hypothesis based on the statistical semantics known as the *bag of words hypothesis*. In this hypothesis the intuition is that we can determine the similarity of two documents by looking at the distribution of words in them. Salton et al. based their work on this hypothesis. The assumption is that the term-document matrix gives us an idea about what the document is about. If we think in terms of topics, then the idea is that the topic of the document will be the driving force of selecting the words which appear in it. Salton et al.'s work was the first of its kind to make this assumption. Later techniques like *Latent Dirichlet Allocation* (LDA) are also built on this assumption.

**The Word-Context Matrix**

In the term-document matrix the goal is to determine the similarity of documents. Deerwester et al [49] suggest that determining word similarities can be achieved by focusing on the row vector in \( X \) instead of the column vector. In the word-context matrix, the rows are words and the columns are different contexts such as documents, chapters, or sentences. The motivation behind this is that word usage patterns might reveal insights about their meaning [69]. This has led to the development of the *distributional hypothesis*. The intuition behind this hypothesis is that when words appear in contexts that are similar, they usually have similar meanings [87]. There has been a significant amount of effort that uses this hypothesis as justification to use VSMs to detect the similarity between words [120, 116, 160, 61]

* LDA will be described in more details late in this chapter
The Pair-Pattern Matrix

In this matrix, the rows are pairs of words, and the columns represent the ways these pairs appear together. For example, a pair can be nurse : doctor, and the patterns of their co-occurrence would be things like “nurse assist doctor” and “nurse work with doctor”. A hypothesis known as the extended distributional hypothesis proposes that the similarity in patterns’ meanings can be detected by the similarity of the pairs they share [116]. The authors in [116] used this hypothesis as a justification to measure column similarity in the pair-pattern matrix to determine how similar two patterns were. Their method was able to conclude that “A resolves B” is similar to “B is solved by A”. In another effort that used the same matrix, the goal was to determine the semantic similarity between word pairs [211]. Turney et al. proposed a new hypothesis known as the latent relation hypothesis which suggests the similarity in the patterns for word pairs can suggest things about the pairs’ semantic similarity. The focus in this hypothesis is on the row vectors in the pair-pattern matrix, rather than the column vector.

2.3.2 Measuring Similarity

The three matrices discussed so far are used to answer different types of questions. The term-document matrix has been used mainly to determine the similarity between documents, whereas the term-context matrix has been used to detect similarity between words. In both cases, similarity is calculated between two vectors and the goal is the assign a real value that captures the alignment in properties between them. If two vectors are identical, the the similarity should be maximum, and if they have nothing in common it should
be at its lowest value. One of the most popular methods for measuring the similarity between two vectors is to take the cosine of the angle between them. The cosine of the angle between two vectors is their inner product after the vectors are normalized to unit length. In mathematical terms, let \( \mathbf{x} \) be the vector \( \langle x_1, x_2, ..., x_m \rangle \) and \( \mathbf{y} \) be the vector \( \langle y_1, y_2, ..., y_m \rangle \). The cosine for angel between them is then measured as:

\[
\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||}
\] (2.10)

In calculating the cosine, the length of the vector does not matter, its orientation that does. If the degree between the two vectors is 180 degrees (i.e., they point to opposite directions) then the cosine is -1. If the degree is 0 (i.e., they are orthogonal), then the cosine is 0, and when the angle is 0 (i.e., they point in the same direction), the the cosine is 1. In the case of VSMs, the values are always positive (they are frequencies), therefore, the actual range of cosine is [0, 1] instead of [-1, 1]. The cosine value actually represents the dissimilarity between vectors. To convert this to a similarity metric, one can use two approaches:

\[
sim(\mathbf{x}, \mathbf{y}) = \frac{1}{\text{dist}(\mathbf{x}, \mathbf{y})}
\]

\[
sim(\mathbf{x}, \mathbf{y}) = 1 - \text{dist}(\mathbf{x}, \mathbf{y})
\]

The cosine similarity measure is only one example of similarity measures that have been used with the VSM models. Measures ranging from geometric distances to measures from information theory have been suggested [95, 115, 224]. In a comprehensive study, where multiple measures were
used, it was shown that the cosine measure had the best overall results [35]. However, there has been a strong suggestion that when the measures are properly normalized, changing them has a low impact on the results [214].

### 2.3.3 Constructing The Matrices

The matrices presented in the previous section go through multiple steps of processing to make them ready for VSM methods. The processes are meant to improve the quality of results obtained from VSM. They are not strictly required to run VSM methods, but it has been shown that they improve the outcomes of such methods.

Assuming that the final list of tokens has been extracted from the text corpus †, the first step is to construct the frequency matrix. In the term-document matrix, each entry captures how many times a certain event occurred. “Event” in this case means a word appearing in a document. These types of matrices tend to be sparse, so a sparse matrix representation is ideal to avoid unnecessary memory usage [78]. This process is similar to building the other matrices, the only difference being the definition of an event. In the term-context matrix, for example, the event is defined as the number of times the word appears in a certain context.

Once this network is constructed, the elements need to be weighted. The aim is to give a higher weight to events that are considered unexpected. The motivation behind this is that rare events are better indication of similarity between two vectors than expected events. Such events have more information encapsulated in them based on Shannon’s information theory [189]. A widely used method, and perhaps the most popular one, is term frequency-

---

†A detailed description to generate the final tokens is discussed in section 2.6
inverse document frequency (tf-idf) weighting [198]. The notion behind this weighing system is that a word gets a high tf-idf value in a document if it has high frequency (high $tf$ value) in the document, and low frequency in all other documents (low $df$ value). When $df$ is low, the inverse of it $idf$ is high. In mathematical terms the $tf-idf$ value for term $t$ in document $d$, which is a member of the document collection $D$, is defined as:

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

where $tf(t, d)$ is the frequency of term $t$ in document $d$, and $idf(t, D)$ is defined as:

$$idf(t, D) = \frac{N}{|\{d_r \in D : t \in d_r\}|}$$

where $N$ is the number of documents in corpus $D$. Is clear that this measure favors long documents, as they will end up have a higher $tf$ value. A variation to this method was proposed to take into account the length of the document [196]. The $tf(t, d)$ can be then calculated as follows:

$$tf(t, d) = 0.5 + \frac{0.5 \times f(t, d)}{\max\{f(t, d) : t \in d\}}$$

In a study to determine the impact of this weighing scheme, it was shown that the $tf-idf$ weighing produced better results than raw frequencies [186].

Although tf-idf has been widely used and is the preferable scheme of weighin in the Information Retrieval field, other methods like Pointwise Mutual Information (PMI) and Positive Pointwise Mutual Information (PPMI) have also been proposed [43, 210, 157].
The term weighing step can also be used in the feature engineering space, where features with near-zero weight values can be removed. An example of such an approach can be found in [72].

One issue with the metrics defined so far is high dimensionality of the vector spaces. For instance, when comparing documents, the number of words can be very large. One way to deal with this issue is to filter out that words that have low weight values across all documents. Another approach was proposed in [49] that is built on the idea of Singular Value Decomposition (SVD). SVD can be used to reduce the dimensions of matrix $X$, and therefore, improve the efficiency of calculating the similarity between the vectors. SVD is a mathematical method that decomposes $X$ into three matrices $U \Sigma V^T$. In $U$ and $V$ the columns have unit length (orthonormal) and are orthogonal. Therefore, $U^T U = I$ and $V^T V = I$. $\Sigma$ is a diagonal matrix where the values in $\Sigma$ are known as singular values [80]. When the top $k$ values from $\Sigma$ ($\Sigma_k$) are considered, the resulting lower-dimensional matrix $\hat{X}$ is obtained by performing the matrix multiplication $U_k \Sigma_k V_k^T$. Here, $U_k$ and $V_k$ are created by selecting the $k$ corresponding vectors from their original matrices $U$ and $V$. In this setting, matrix $\hat{X}$ is the best matrix of rank $k$ that minimizes $||\hat{X} - X||$ (the approximation errors).

SVD has been used in the field of Information Retrieval in many cases like discovering latent meanings [49], noise reduction [178], high-order co-occurrence [109], and sparsity reduction [216].

In this section have discussed described an algebraic model for semantic representation known as vector space models (VSM). VSM is one of the most used techniques to represent words and documents in a way that allows
researchers to extract word and document meanings automatically. A more detailed description about VSM and its application can be found in [212]. In the previous section, we described network representations for text, which can also be used to extract semantics from texts. In the next section we review some of the techniques that leverage the VSM to extract topics from collections of documents.

2.4 Text Clustering

The idea behind text clustering is to group documents according to their content. The basis of most text clustering algorithms is to use terms as features, and then group documents based on a similarity measure. Different similarity measures have been proposed [30], with cosine similarity as the most widely-used [106].

In most of the basic text clustering techniques, the number of clusters is known a priori to the algorithm. For example, when K-means and expectation maximization (EM) [20] are used for text clustering, the number of clusters is an input the clustering algorithm, and the goal is to minimize the distance between the data points within the same cluster while maximizing the distance between clusters. Clusters are usually represented by the mean or the weighted average of their points, and the distance between documents is calculated using one of the similarity measures in [30].

In addition to the variation in the similarity measure, text clustering algorithms can use different clustering techniques. The most used techniques in clustering text documents are partitioning, hierarchical, and density-based algorithms.
Partitioning Algorithms

In partitioning algorithms, the documents are partitioned into clusters such that each document belongs to only one cluster. Each cluster is represented by a cluster centroid, which typifies all the documents that belong to the cluster. The most common example of this is K-means clustering. Another interesting partitioning algorithm is Fuzzy c-means [228], which calculates a degree of membership for each document for every cluster. The work in [195] also shows that the fuzzy c-means clustering algorithm is more stable, and produces better results on most data sets.

Hierarchical Algorithms

Hierarchical algorithms have been extensively applied in the area of text clustering, e.g. [75]. In some cases, hierarchical methods can have a top-down approach, where the algorithm starts with one cluster that contains all documents, and then systematically splits the cluster into sub-clusters. The sub-clusters are then split further down, until each document is in a cluster of its own. In other cases, hierarchical algorithms can follow a bottom-up approach, where the clustering starts with each document being its own cluster and clusters are merged until all documents belong to one cluster.

Other than the direction of clustering, one other main difference between hierarchical algorithms is the similarity measure between the clusters. For example, SLINK [190] uses the single linkage method where the similarity between two clusters is the similarity between the two most similar documents. Note that the similarity between two documents could be any of the similarity measures we discussed in section 2.3.2, such as cosine similarity. One
2.4. TEXT CLUSTERING

other measure for merging or splitting clusters (depending on the problem) is CLINK [50]. In CLINK, the similarity between the clusters is the similarity between the most dissimilar pair of documents, where each document belongs to one of the two clusters. Comprehensive reviews of hierarchical clustering are provided in [201, 230].

Density-Based Algorithms

Density-based algorithms are another set of algorithms that have been proposed for text clustering. In this approach, clusters are required to have the minimum number of documents where the similarity between them is higher than a specific threshold. This approach does not require a priori knowledge of the number of clusters, and is able to detect clusters of different sizes. One main assumption made by these algorithms is that clusters that have a low density can be considered outliers. Although this assumption can allow for a high tolerance of noise, it also makes the quality of the results dependent heavily on the density threshold. One other factor in the quality of the results is the similarity measure between the documents. Although the Euclidean distance seems to be the most popular distance measure used in this approach, it tends to be problematic for sparse, high dimensional data sets.

The Density-Based Spatial Clustering and Application with Noise (DBSCAN) is one of the most widely-used examples of this [62]. Similar to all other algorithms that use this clustering technique, DBSCAN does not require a priori knowledge of the number of clusters. The inputs to the algorithm are two parameters, the density threshold, and the maximum number of points in each cluster.
2.5 Topic Extraction

A useful way to think about a corpus of documents is that the documents cover a fine set of unknown topics, each of which has its own semantics and therefore implicitly specifies a pattern of word usage. Topic extraction is a computational process by which this latent structure can be discovered by analyzing the pattern of word occurrence in the documents comprising the corpus.

Following the VSM definition, a document in most topic extraction approaches is represented by a vector of real numbers, where each number represents the frequency of a term appearing in the document (the event). Terms in this scenario can be seen as features of documents, and the corpus can be represented by a word-document matrix $F = W \times D$. In $F$ each entry $F_{ij}$ is the frequency of word $i$ in document $j$. Several approaches for topic extraction are briefly described below.

2.5.1 Topic Extraction via Vector Space Models

Deerwester et al. were the first to propose a method that utilizes singular value decomposition (SVD) to extract latent structure in text corpora. The method, known as Latent Semantic Indexing (LSI), was developed to improve the matching of documents against queries in Information Retrieval [49]. The intuition behind the method is that semantic associations can be inferred by representing documents in high-dimensional space. The hope was that by placing documents in a space where the variance is maximized, it would be possible to retrieve documents that are more relevant to the given query.

\[^{\dagger}\text{A more detailed description of constructing such a matrix is given in section 2.3.3}\]
Using $\mathbf{F}$, the space where documents are embedded is defined by the words in the corpus. Deerwester et al. proposed the use of a singular value decomposition (SVD) approach on the term-document matrix to extract a subspace where the variance in the documents is maximized. This process produces new features that are a linear combinations of the original term features.  

In LSI, before SVD is applied, the matrix $\mathbf{F}$ goes through a transformation to construct matrix $\mathbf{G}$. This transformation is usually done by applying an entropy normalization for word distribution over documents [82]. Every entry in $\mathbf{G}$ is calculated as:

$$G_{ij} = \log(F_{ij} + 1)(1 - H_i)$$

where $H_i$ is the normalized entropy distribution for word $i$ over all the documents, and is calculated as:

$$H_i = -\frac{\sum_{j=1}^{D} \frac{F_{ij}}{F_i} \log \left( \frac{F_{ij}}{F_i} \right)}{\log D}$$

Once $\mathbf{G}$ is obtained, SVD is then applied on it to get a vector for every word in the reduced space. As the goal of LSI was to improve indexing in Information Retrieval, a more general theory known as *Latent Semantic Analysis* was proposed by Landauer and Dumais which applied the methods of LSI to the broader goals of topic extraction and word prediction [109]. In word prediction, LSA was able to predict what context a word comes from fairly well.

LSI, and more generally LSA, can be seen as methods to improve the matching between documents. With a similar perspective, Lund and Burgess

---

§SVD is described with more details in section 2.3.3
proposed a method known as *Hyperspace Analogue to Language* (HAL) that have been used to determine the similarity between words [120]. The intuition was that words that appear together are usually semantically related. In HAL, a co-occurrence matrix $W$ of words is constructed where the rows and columns are the words in the document collection. The entry $W_{ij}$ represents the number of times the words $i$ and $j$ co-occur. In determining co-occurrence, the authors define a moving window of $r$ words. The words that appear within the window $r$ are considered co-occurring, and are captured in matrix $W$. For every word $w$ in the corpus, there is one row and one column in $W$ that, respectively, capture co-occurrences before and after the word. Concatenating the row and the column gives a vector $v$ of length $2N$ representing all the co-occurrences for $w$. The vector $v$ can be seen a point in a high-dimensional space of size $2N$. As the length of $v$ can become very large, the algorithm considers only the 100-200 components that show the highest variance. The authors in [120] demonstrated that this was sufficient to produce good results.

### 2.5.2 Topic Extraction via Probabilistic Generative Models

To better evaluate the performance of LSA, Papadimitriou suggested that one needs to develop a generative probabilistic model and determine if LSA is able to recover the characteristics of the generative model from the data [165]. Hofmann argued that LSA lacks a statistical foundation, and therefore, he proposed a probabilistic LSA (PLSA) based on the likelihood principle [89]. In PLSA documents are mixtures of latent topics, and each word in a document is generated from a single topic. PLSA ignores the position of
words in the document and treats each document as a bag of word \(^\dagger\).

PLSA is based on a statistical model known as the *aspect model* \([90, 187]\) and it aims to associate a latent variable (topic) \(z \in Z = z_1, z_2, ..., z_k\) with each observation of a word \(w \in W = w_1, w_2, ..., w_M\) in document \(d \in D = d_1, d_2, ..., d_N\). To define this as a generative process, the probability of obtaining document \(d\) and word \(w\) is defined as:

\[
p(d, w) = p(d) \sum_z p(w|z)p(z|d)
\]

where \(p(d)\) is the probability of selecting document \(d\), \(p(w|z)\) is the probability of selecting word \(w\) given given latent variable \(z\), and \(p(z|d)\) is the probability of selecting latent variable \(z\) from document \(d\). In this setting \(d\) and \(w\) are conditionally independent given \(z\).

PLSA determines \(p(d)\), \(p(w|z)\), and \(p(z|d)\) by maximizing the log-likelihood function

\[
\mathcal{L} = \sum_{d \in D} \sum_{w} w_w \log p(d, w) = \sum_{d \in D} \sum_{w} w_w \log P(d, w) \tag{2.11}
\]

where \(F_{wd}\) is the number of times word \(w\) appears in document \(d\).

One issue with PLSA is that it does not provide a probabilistic model at the document level. It does assume that every document is a mixture of topics, but it does this only for the training documents. Therefore, there is not a well-defined way to apply the method to non-observed documents. Also, not having a probabilistic model at the document level requires the number of parameters to grow linearly with the corpus size. This increase in the number of parameters leads to over-fitting and causes issues in determining the probability for documents that do not exist in the training set. Blei et al. proposed a method, known as *Latent Dirichlet Allocation* (LDA), that introduces

\(^\dagger\)This follows from the bag of word hypothesis discussed in section 2.3.1
a probabilistic model at the document level [25]. LDA also uses the a bag of word approach. Like PLSA, it assumes that a single document can have multiple topics, and each topic is a distribution over words. LDA is different than PLSA in that it assumes that the distribution of topics in document $i$ ($\theta_i$) has a Dirichlet prior with parameter $\alpha$, and the distribution of words over topic $k$ ($\phi_k$) has a Dirichlet distribution with parameter $\beta$. For each document $d$, every topic $k \in K$ has a proportion of $\theta_{dk}$, and $\theta_d$ is the proportion of all topics in document $d$. If $z_{dn}$ is the topic of word $n$ in document $d$, $z_d$ is the topic assignment of all words in $d$, and $w_{dn}$ is the $n^{th}$ word in document $d$, the generative process for LDA is then defined as

$$p(\beta, \theta, z, w) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left( \prod_{n=1}^{N} p(z_{dn}|\theta_d) p(w_{dn}|\beta, z_{dn}) \right)$$ \hspace{1cm} (2.12)$$

Learning the different distributions for LDA is a Bayesian inference problem and it is solved by a variational Bayes approximation. Other techniques were also proposed that use Gibbs sampling [82], and expectation propagation [138]. LDA has become the most widely used algorithm in this space, and has been the focus of many research initiatives. A more detailed description of LDA can be found in [25]

LDA and PLSA are generative probabilistic topic modeling methods. In these methods the assumption is that documents are generated by a generative process that has latent variables. Following this assumption, the observed variables are words and documents, and the latent variables are topics. In LDA, several statistical assumptions are made about the documents. For instance, one major assumption is that documents are bags of words. However, the appearance of a word usually depends on the words before it.
To address this issue, Wallach proposed a modification in LDA where this is taken into account [219]. Another modification was proposed by Griffiths et al. where the topic model is combined with a hidden markov model [83]. Another approach relaxes the assumption that the order of document does not matter and proposes a model where topics change over time [24]

One issue with models like LDA is that the number of topics needs to be specified a priori, and this choice makes a significant difference. When analyzing document collections, the number of topics is usually unknown. One way to solve this problem is by using the Bayesian nonparametric topic model [209]. In this model the number of topics is determined during the posterior inference step. Another approach extracts a hierarchy of topics where the structure is extracted from the data [23]. Arun et al. has proposed another method for determining the number of topics in LDA based on matrix factorization [12]. In this method, a document by term matrix $D = D \times W$ is split into two matrices $M = D \times K$ and $Q = K \times W$ where

$$D = M \times Q$$ (2.13)

Here, $D$ is the number of documents in the corpus, $K$ is the number of topics, and $W$ is the number of words in the vocabulary. This approach tries to find the best number for $K$ that optimizes the quality of the split. They calculated the KL-Divergence of the distributions that are extracted from $M$ and $Q$ and determine the $K$ value that gives the lowest divergence. More details about this method can be found in [12]

In this section we described two classes of methods to extract latent structures in texts. One of them is based on the VSM model, and the other is based on probabilistic modes. In this thesis we propose a different approach
that is based on extracting communities from word networks. Therefore, hav-
ing an understanding on the topic extraction techniques described in this sec-
tion is crucial to determine the value of our approach. In the next section we
describe some methods used to process raw text and convert it into a format
that can be used by the algorithms described in this section.

2.6 Text Processing

When dealing with text mining, researchers usually get the data in the form
of free text. Getting the data this way is easy, and it provides a baseline
on what the original data looks like. To build VSM metrics, or even term-
networks, text corpora must go through some preprocessing to generate an
appropriate list of tokens. Once the list of tokens is extracted, the process
of building VSM metrics, or term-networks becomes straightforward. In this
section we describe three steps that are usually used to generate the list of
tokens.

2.6.1 Tokenization

Tokenization is the process by which we take a paragraph and extract a list
of words. Although this sounds simple, which it might be for the English
language, it is certainly a challenging process for other languages. A sophis-
ticated tokenization process can make a significant difference on the results
obtained from models like VSM. For instance, a tokenizer should be able
to handle punctuation like John’s and aren’t. It should able to extract multi-
word phrases like Kobe Bryant, global warming and South Africa [124]. This
becomes a more challenging task when the language at hand does not sep-
2.6. TEXT PROCESSING

arate words by spaces. For example, in Chinese words are not separated by spaces. For cases like this, matching text against a lexicon is one of the proposed solutions [200]. Once a list of tokens is generated, it is usually common to remove *stop words* such as articles, prepositions, conjunctions, and functional verbs such as ‘is, may, will, can, etc. A popular list of stop words was produced for the SMART system and is still widely used [185].

2.6.2 Normalization

In all languages, there are instances where different strings can have the same meanings. For example in English, the word *car* and *automobile* have a similar meaning. If the goal is to have one word that represents both, we can perform an artificial replacement to keep one of them. In this case, if we want to keep *car*, we replace every instance of *automobile* with *car*. Capitalization also introduces some issues. For instance, if the word appears at the begging of a sentence, the first letter will be capitalized. If it happens that the same word appears somewhere else, where the first letter is not capitalized, then it might be considered another word. To solve this, we usually convert all words to lower case in what is known as *case folding*. Both of these are examples of text normalization. Another type of normalization is *stemming*, which is the process by which we replace every word with its root. In the English language, words are usually made of stems. Affixes can be added to the stem to indicate plural form (book and books), or the past tense of a verb (book and booked). Stemming in English is relatively easier than other languages [94]. When stemming is performed with full attention to grammar, it is called *lemmatization*. Many stemming and lemmatization algorithms have been proposed and work fairly well [119, 171, 139]. More details about nor-
malization can be found in [212]

2.6.3 Annotation

It is common to have two different meanings for the same word. For instance, the word *bank* can mean a financial institution, or the land alongside a river. Annotation helps us deal with issues like this to improve precision. One form of annotation is using part-of-speech tagging. By tagging the word *book* as verb, we will be looking for documents that have to do with booking a trip for example. On the other hand, if it is tagged by noun, then we will be looking for reading material. Other forms of tagging include word-sense tagging where words with many meanings are tagged with the intended meaning, and parsing where words are tagged by their grammatical role[125]. In the area of Information Retrieval, annotation has produced significant improvement in search algorithms [18, 112, 40]. Annotation is also used to improve the calculation of similarity between words in a word-context matrix [164].

In this chapter we have described some of the concepts that we will be using throughout the work presented in this dissertation. We started by looking at general network concepts and then reviewed some of the work in representing languages as networks. We touched on a different technique for representing language known as the vector space model. VSM is one of the most used semantic representations in the area of Information Retrieval and is the basis for many topic extraction techniques presented in section 2.5. We closed the chapter by looking at different techniques that are used to process raw text into tokens. These tokens are then used to construct the different matrices that are used for VSM. As we mentioned in chapter 1 the goal of the dissertation is to develop the concept of a unified semantic space where
documents, authors and venues can be embedded. This extends the work in the VSM model to allow embedding of different types of entities. In VSMs, the goal is to usually embed documents or words in a semantic space such that similarity between the objects can be determined using a similarity function based on their location in the space. In our approach the goal is to embed different entity types such as authors and venues. In our model, the dimensions of the space are topics extracted from text corpora. For that, we use a network based approach where topics are defined as communities of words. In the next chapter we describe the process by which we construct a term network from text corpora, and use a community extraction method to extract communities of words that we interpret as topics.
Chapter 3

Topic Extraction - A Network Based Approach

As described in Chapter 1, the primary aim of this research is to analyze text corpora to extract an epistemic space where different epistemic entities can be embedded and related. While the idea of constructing and using such representational spaces is very general, the focus in this thesis is on corpora of plain English text, and the types of entities to be considered (see the next chapter) are mainly documents, authors and publication/presentation venues.

The representation of words in semantic spaces is a well-studied problem, as discussed in Chapter ???. The focus in these cases is on finding meaningful latent variables through dimensionality reduction methods and using these latent variables as the dimensions of the semantic space. The best-known example of such an approach is latent semantic analysis (LSA) [110], but others include [120, 93]. All these can be regarded as vector space approaches. An alternative way to identify meaningful structure in semantic space is to cluster words, but this does not necessarily yield an appropri-
ate embedding space. A much more promising method is to use one of the topic extraction models, and to use the topics as dimensions. As discussed in Chapter 2, there are several methods for automatic topic extraction, but latent Dirichlet allocation (LDA) [25] has emerged as by far the most dominant approach. It is an unsupervised approach and can, therefore, work on large, unlabeled corpora. However, it has some limitations including overly simplistic underlying assumptions, the need to pre-specify the number of topics, great sensitivity to this parameter, and general variation in results across multiple runs. LDA is also difficult to apply when the corpus consists of very short documents (e.g., tweets) or when it is a single large document. In this thesis, we propose an alternative approach to topic extraction based on the analysis of lexical association networks constructed from the corpus. One motivation is to devise a more cognitively grounded approach using the fact that brains represent semantic information largely through associations [46, 166, 127].

The proposed approach, termed topic extraction through partitioning of lexical associative networks (TExPLAN), is based on the idea of looking at each document as an associative network of words that is a partial, highly context-dependent “snapshot” or sample of the much more complex associative knowledge networks underlying the cognitive processing that created the document [54]. We hypothesize that, while these lexical networks represent noisy samples, important information about the organization of knowledge in the author’s mind can be gleaned from them [54]. One such piece of information is the topical structure of knowledge. The method proposed in this thesis is to identify topics with communities or modules in the lexical networks constructed from text corpora. Each community is a group of words
that tend to be used together with greater-than-expected frequency, and thus form the vocabulary of a particular topic. When the corpus represents the work of a single author, this provides insight into the author’s topical preferences, but when the corpus represents the work of many authors, the topics that emerge from the analysis of the lexical associative network indicate the consensus topical preferences of that entire group of authors. In this way, the approach can extract meaningful epistemic structure at many levels using different types of text corpora.

In Chapter ?? we discussed how community extraction approaches fall into two classes: Those that produce mutually exclusive communities (partitioning), and those that allow overlapping communities. In analyzing lexical networks, the latter approach is generally preferable, since the same word can clearly belong to multiple topics. However, topics created by mutually exclusive partitioning also have their advantages. They are easier to interpret and can be used as features without worrying about confounds created by overlap. As implied by its name, the TExPLAN algorithm is based on partitioning lexical networks, which initially yields non-overlapping topic modules. These are then processed further in the second stage of the algorithm to obtain overlapping topics. Since both types of topics are of interest, we consider each stage separately in this thesis. The topics obtained through TExPLAN are validated by comparing them with those obtained through latent Dirichlet allocation (LDA). Such comparisons over several datasets indicates that TExPLAN detects meaningful structure of quality similar to or better than LDA in a robust way.

This chapter first describes the Stage I network-based approach to extract non-overlapping topics from text corpora. We apply the method to four
different datasets and discuss the results. We then describe the Stage II process for obtaining overlapping topics from the non-overlapping ones, and discuss its outcomes. In the last section we compare our results to LDA and discuss the advantages and disadvantages of the approach.

3.1 TExPLAN Stage I: Term Network Partitioning

The motivation behind using a network-partitioning approach to extract topics is the fact that if words appear together in the same document (or proximity), then they are more likely to be related. A group of such words that occur together preferentially, and tend not to occur as much with other words, naturally represent the vocabulary of an epistemic topic. To capture this, we build a lexical associative network – henceforth called the term network – based on joint occurrence of words in the text corpus and use it to extract communities of words representing these strong interconnected topic groups. In the term network $G = (V, E)$, the set of nodes $V$ represent all the unique words in the corpus, i.e., its vocabulary, and the set of edges $E$ is the set of all edges that connect these nodes through associations. Two words are connected if they appear together in one or more documents, and the edge weight is the number of documents in which the two words co-occur. The more the two words appear together in the same documents, the more related they are, and have a higher chance of being in the same topic. All the networks described in this thesis are undirected networks unless otherwise specified, and the documents are all taken as bags of words where order is not defined. It should be noted that, while we use co-occurrence in documents to build networks, the same could be done with co-occurrence in paragraphs or
3.1. **TEXPLAN STAGE I: TERM NETWORK PARTITIONING**

sentences, or based only on proximity of occurrence [77, 54, 133, 56, 39].
This can be especially useful in corpora where each document is a long text,
or even a single very long document such as a book.

### 3.1.1 Method Description

A well-known algorithm called the Louvain algorithm is used for the initial community extraction step in TExPLAN [27]. The Louvain algorithm works by optimizing the modularity of partitions in a network, where the modularity of the partition measures how strongly the nodes within each community are connected internally relative to their connections to nodes in other communities. It is defined mathematically as follows:

$$Q = \frac{1}{2m} \sum_{ij} [w_{ij} - \frac{w_i w_j}{2m}] \delta(c_i, c_j)$$  \hspace{1cm} (3.1)

where $c_i$ is the community to which node $i$ is currently assigned, $w_{ij}$ is the weight from node $j$ to node $i$, $w_i = \sum_j a_{ij}$ = total incoming weight for node $i$, $m = \sum_i w_i$ = total weight of the network, and $\delta(c_i, c_j) = 1$ if $c_i = c_j$, 0 otherwise.

Basically, this statistic measures the “excess modularity” compared to what would be expected if nodes from the same network were grouped at random into modules of the current sizes.

The Louvain method for community extraction uses a hierarchical, recursive and greedy strategy for the optimization process, and produces a hierarchy of communities that allows the discovery of sub-communities. The method starts by placing every node in its own community and merges communities to optimize modularity locally. It then creates a new network whose nodes are the new communities and applies the same optimization process.
on it. This process is applied recursively until a hierarchy of communities with maximum modularity is created. The pseudo code for the Louvain method is shown in Algorithm 1.

**Algorithm 1** The Louvain Community Extraction Method

1: function \textsc{LouvainCommunityExtraction}(G)
2: \hspace{1em} \textbf{c} ← make every node $i \in G$ its own community
3: \hspace{1em} \textbf{while} No further changes occur \textbf{do}
4: \hspace{2em} \textbf{for} each node $i \in G$ \textbf{do}
5: \hspace{3em} $N_i \leftarrow$ neighbors of $i$ in $G$
6: \hspace{3em} $m \leftarrow$ The community with the highest modularity once $i$ is moved to it
7: \hspace{3em} \textbf{if} $m$ is not Null \textbf{then}
8: \hspace{4em} update \textbf{c} by moving $i$ to $m$
9: \hspace{3em} \textbf{end if}
10: \hspace{2em} \textbf{end for}
11: \hspace{1em} \textbf{end while}
12: \hspace{1em} return \textbf{c}
13: \textbf{end function}

For the algorithm in 1, the input $G = (V, E)$ is a network with $V$ nodes and $E$ edges. The output is a set $S_v$ of disjoint subsets of nodes where each node in $V$ belongs to one and only one subset $V_k \in S_v$.

Mathematically, the algorithm’s function is given by:

$$
\mathcal{L}(G = (V, E)) \leftrightarrow S_v = \{V_1, V_2, ..., V_{N_k}\}
$$

(3.2)

Where $G$ is a network of $V$ nodes and $E$ edges, $S_v$ is a set of $N_k$ partitions of nodes that belong to $V$ where $V_1 \cup V_2 \cup ... \cup V_k = V$ and $V_k$ is the set of nodes which belong to partition $k$.

The definition in Equation 3.2 indicates that all nodes in $V$ must be in one and only one subset $V_k \in S_v$. This follows from the algorithm since it starts by assigning every node to a singleton partition, and merges partitions hierarchically. According to this definition, each topic $k$ has a set of words $V_k$.
that only belong to topic \( k \). From this we can derive a set of edges \( E_k \) where each edge \( e_i \) belongs to \( E_k \) if both end-points of the edge are in \( V_k \). We define the membership function \( M_E(e, k) \) on edge \( e \) and community \( k \) as follows:

\[
M_E(e, k) = \begin{cases} 
1 & \text{if } v_{e_a} \& v_{e_b} \in V_k \\
0 & \text{otherwise}
\end{cases}
\]

where \( v_{e_a} \) and \( v_{e_b} \) are the two nodes connected by \( e \).

So far we have defined a term networks for a dataset and described how it is constructed. We have also defined a set of nodes \( V_k \) and a set of edges \( E_k \) for every topic \( k \). Since each topic is made of nodes and edges, we can then construct a topic network, \( G_k = (V_k, E_k) \) for every community \( k \) in the term network \( G \).

The next step in the topic extraction process is to assign an importance value to every word in its topic. This is done by measuring each node’s hub value, calculated using the Hyperlink-Induced Topic Search (HITS) method described in Section ?? . HITS calculates two values for each node; hub and authority. This algorithm is used to determine the importance of web pages on the Internet and it is the basis of the famous Google PageRank algorithm [32]. The Internet is made of directed links between pages, so in-degree and out-degree are different. Based on this, the hub and authority values are defined in terms of each other: hub value determines how many of a node’s outgoing links point to nodes with high authority, and the authority values measures how many of its incoming links p come from important hubs. Informally, authorities represent nodes with important content (thus attract-
ing many links), and hubs act as central locations from which many authority nodes can be accessed. Since the term network in our model is undirected, the difference between hubs and authorities disappears, and the value calculated corresponds to a metric called \textit{eigenvector centrality} \cite{149}, which is defined recursively as follows: Nodes with high eigenvector centrality are those that connect to other nodes with high eigenvector centrality. To keep terminology simple and intuitive, we use the term \textit{hub value} for the eigenvector centrality of a node, and use it as a measure of its significance.

One of the advantages of using this network partitioning method for topic extraction is that a priori knowledge of the number of topics (communities) is not required. On the other hand, methods like LDA require a priori knowledge of the number of topics. The Louvain method is fast and efficient, and although its computational complexity is not known, it is believed to be \textit{O}(\text{nlogn}). In fact, the Louvain method is the most widely used community extraction method for large networks due to its efficiency and accuracy. One drawback of this algorithm is the possibility of ending up at a local maxima. This is because the algorithm uses a greedy strategy in finding a solution.

To demonstrate the network-based approach for topic extraction, we applied it to four different datasets described in Section 1.2. That datasets vary in terms of complexity and have different levels of structure. In the NSF dataset, the areas of documents are quit disjoint and we expect the clustering method to do very well. The DBLP dataset has areas that are very overlapping, and we expected to be more complex than the NSF dataset. The IJCNN dataset is the most complex dataset since it really comes from one area of interest (AoI): Neural networks. As such, topics within it are likely to be much harder to find and somewhat arbitrary. The last dataset is the
University of Florida word association norms, and it is quite different from the other datasets because the notion of document is not defined at all, the data having been obtained experimentally by asking people to produce associated words in response to explicit cue words. Therefore, methods like LDA cannot be applied to this dataset at all. This serves an example of how the network-based approach for topic extraction can go further than the topic models described in Section ??

3.1.2 NSF Abstract Dataset Partitions

We applied the algorithm described above to the NSF abstracts data set described in Section 1.2. The set comprises 129,000 abstracts from proposals for basic research submitted between 1990 and 2003. The dataset provides a bag of words for each abstract and the affiliated investigators. For the purposes of this research, we extracted the set of abstracts that belong to five of the ten largest research programs:

1. ALGEBRA and NUMBER THEORY

2. BIOLOGICAL OCEANOGRAPHY

3. ECONOMICS

4. ELECT, PHOTONICS, and DEVICE TEC

5. GEOPHYSICS

As described previously, the process starts by creating a term network from the text. Two terms are connected if they appear together in the same document, and the weight of their connection is the number of documents
The resulted network has 11,844 nodes and 4,024,707 edges and, for visualization purposes, we show the top 5% of its edges and nodes in Figure 3.1.

To get a better understanding about the network structure, we start by looking at two basic network properties: degree distribution and clustering coefficient. Both of these were described in Section ??.

Figure 3.2(a) shows the degree distribution of the nodes in the network. Each bin is of size 5 and the height of the bar reflects how many of the nodes have that degree. As we can see the number of nodes with high degrees is relatively very low compared with the number of nodes with low degrees. This is also reflected in the rank plot shown in Figure 3.2 (b). In the rank plot we order the nodes with respect to their degrees in a decreasing order, and plot the degree value vs. the rank on a log-log scale.
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

In Figure 3.2(c) we plot the clustering coefficient distribution of the terms in the term network. From the figure we can see that there are 3 peaks at clustering coefficient values 0.006, 0.0085, and 0.015. The lower two cover most of the terms, and indicate that most of the terms have a low value of clustering coefficient. The third peak represents some terms with relatively higher clustering coefficient values. A word cloud of these terms is shown Figure 3.3 where the size of the word is its clustering coefficient value. As we can see these words are not really central to any specific area of interest, and they seem to be terms within small, tight-knit clusters.

We also looked at network properties related to node centrality. In Section ?? we described in more detail the following four measures of node centrality:
Figure 3.3: Terms with high clustering coefficient values in the NSF term network

- **Degree Centrality**: This is the most straightforward definition of centrality. It determines the importance of the node purely based on its degree. The node with the highest degree is the most important.

- **Betweenness Centrality**: The centrality of the node is the ratio of the shortest paths between all other nodes that pass through it.

- **Closeness Centrality**: This centrality value measures how close the node is to all other nodes in the network. The closer the node is to all other nodes, the higher the closeness centrality is for that node.

- **Eigenvector Centrality**: This centrality measure determines the influence of the node in the network. The assumption is that important nodes are connected to other important nodes.

In Figure 3.4 we show the different distribution of centrality measures. From the figure we see that the degree, betweenness and closeness centralities have similar distributions and rank plots. The degree centrality is essentially the same as the degree distribution plotted earlier. However, most nodes seem to have a closeness centrality value around 0.5. This means that most nodes are not too far and not too close to all other nodes in the network.
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

Figure 3.4: NSF dataset term network centrality measures
In Figure 3.5 we show words with the highest 1% of centrality values in the network. In general, degree, betweenness, and closeness centralities are positively correlated. From the figures we can see that the top words for these three measures are usually the same; models followed by data. For the words with the top 1% eigenvector centrality, we notice that the word theory is the highest, followed by models. This suggests potentially interesting facts about the data – especially about the role words such as models, theory and data play in NSF proposals – but we do not follow this line of investigation in this thesis.

Next, the community extraction process is applied to the term network for the NSF dataset to obtain topics. As expected, the algorithm found five distinct communities, presumably reflecting the five topics in the original dataset. Figure 3.6 shows the term network where nodes from the same topic are colored with the same color. The separation of communities is quite noticeable, reflecting the very distinctive topics inherent in the dataset.

The communities are groups of words whose interconnections within the group are stronger than the connections with other nodes in different groups.
As previously noted, the intuition is that such groups of words indicate topics. Based on this, we are able to get a separate term network for each topic, which we call the topic term network. The significance of words within each topic term network can be assigned using the hub value as discussed earlier. Table 3.1 gives the top five terms in each community, which are clearly related to the five original topics on which the dataset was based. A more complete snapshot of the discovered topics can be seen by plotting word clouds for the most significant words in each community. The size of the word in each word cloud indicates its hub value. These word clouds are shown in Figure 3.7.

As we can see from the extracted topics, each one of them maps to one of the NSF program represented in the dataset. The mapping is clear and the topic extraction method did really well. One of the reasons why the partitions
are very clear is the nature of the data. This dataset is inherently disjoint and topics do not overlap significantly at the semantic level, though individual terms may overlap.

The next dataset we apply our method to is slightly more challenging and the results are described in the next section. The goal is to demonstrate how the network-based topic extraction method is applicable to datasets that have more of an overlapping nature.
3.1.3 DBLP Dataset Partitions

The second dataset to which we applied our algorithm is the Digital Bibliography and Library Project (DBLP) dataset [51]. The DBLP dataset is a collection of publications from well-known computer science conference proceedings. In this dataset, each paper is represented by the group of words that appear in it, i.e., as a bag of words. The dataset also contains authorship information and indicates the conference where the paper was published. The publications in this dataset are from four main computer science areas: databases, data mining, information retrieval, and artificial intelligence. It contains 28,569 papers, 28,702 authors, 20 conferences, and 11,771 unique terms. This dataset is more challenging to the network-based topic extraction method because it has a significant overlap between topics. The areas in this dataset are closely related, and terms that fall into one partition, could very well be consistent with other partitions as well.

As with the NSF dataset, we construct the term network by connecting two terms that appear in the same document, and the weight is the number of documents in which both terms appear in. The final network contains 11,771 nodes and 3,810,446 edges. The network defined by the top 5% of the edges is shown in Figure 3.9. The figure indicates that this network is more connected at the core than the NSF dataset. This is due to the nature of the dataset; the topics are not as distinct as in the NSF dataset.

For more insight about the term network, we start by looking at some basic network properties. In Figure 3.9(a) we show the degree distribution of the nodes in the network. As with the plot for the NSF dataset, each bin is of size 5. Unlike the degree distribution for the NSF dataset, that for the DBLP
Figure 3.8: Term network for the DBLP dataset

term network shows a slightly bimodal distribution, indicating the presence of a small fraction of words with high degree.

To investigate the words in the second peak – those with degree greater than 1,100 – we calculate their centrality values, and their clustering coefficients. As discussed in the case of the NSF dataset, there are four different centrality measures that we look at. For each measure we create a word cloud where the size of the word in the word cloud reflects its centrality value. We also create a word cloud where the size of the word reflects its clustering coefficient value. The word clouds can be found in Figure 3.10. The words which have a degree value in the second peak do not stand out as significant topic terms based on any of the centrality measures or the clustering coefficients. They seem to be words that are generally used across different documents regardless of the topic, i.e., functional words. This makes sense
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

Figure 3.9: DBLP dataset term network properties

because such words are likely to be connected to terms from all topics, and therefore acquire higher degree. It is interesting to note that the eigenvector centrality and clustering coefficient identify very different functional words than the similar sets identified by the other centralities. This is potentially an interesting observation from the viewpoint of text filtering, and will be investigated in future research.

Similar to the term network for the NSF dataset, we calculate the four different centrality measures for the DBLP term network. In Figure 3.12 we show words with the highest 1% of centrality values in the DBLP term network. As in the NSF dataset, degree, betweenness, and closeness centralities are positively correlated in general. From the figures we can see that the
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

(a) Degree Centrality Distribution  
(b) Betweenness Centrality

(c) Closeness Centrality  
(d) Eigenvector Centrality

Figure 3.10: DBLP terms with degrees in the second peak of the degree distribution.

The top word is always \textit{data}, which can be regarded as a central word for all the topics in this dataset. It is also apparent that all centrality measures pick up some of the words that are of common importance across the whole dataset, such as \textit{system}, \textit{learning}, \textit{model}, \textit{information}, \textit{search}, etc., but not the words that could be topic-specific.

The step is to apply the community extraction method on the term network to extract communities of words as shown in figure 3.13. As before, node colors indicate community membership.

For each community of terms (topic) we generate a word cloud as shown in Figure 3.14. Similar to the NSF dataset results, the size of a word in the word cloud reflects its hub value within its topic network.

As mentioned previously, this dataset has a more overlapping structure than the NSF dataset, and the number of topics is not clear \textit{a priori}. The
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

Figure 3.11: DBLP dataset term network centrality measures
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

(a) Degree Centrality Distribution  
(b) Betweenness Centrality  
(c) Closeness Centrality  
(d) Eigenvector Centrality

Figure 3.12: DBLP Terms with High Centrality Values

Figure 3.13: Term network of the DBLP dataset with community labeling
network-based topic extraction method was able to determine that the optimal number of topics is 5, and that partitioning into these 5 communities ensures the highest modularity value. This is an advantage over other topic extraction methods like LDA, where a priori knowledge of the number of topics is assumed.

Table 3.2 has a list of the top terms in every topic, though given the highly interwoven nature of the topics, it reasonable to expect that most of the high ranked terms may also be significant in other topics – as will become clear when we obtain overlapping topics. However, the topics extracted by the method show clear cohesion and they are meaningful. In Table 3.3 we map each word community to a meaningful research area.

The DBLP dataset is different than the NSF dataset in the sense that all five extracted topics are in the computer science area, where many terms
can be significant in many topics. For example, the term *data* is the node with the highest value for degree and for all centrality measures in the term network. It is an example of a term that can have a significant value in more than one topic. In Section 3.2, we describe the process which allows terms like *data* to be assigned to multiple topics.

In the next section we describe another dataset which is even more narrowly focused than the DBLP dataset in the sense that the documents in it discuss research in a single field: *Neural networks*.

### 3.1.4 IJCNN Dataset Partitions

The IJCNN dataset is based on papers published in the proceedings of the annual *International Joint Conference on Neural Networks* organized by the IEEE and the International Neural Network Society. The dataset consists of 2,012 abstracts of papers from several years between 2000 and 2012, and has more than three thousand unique terms.

IJCNN is a more challenging dataset than the NSF and the DBLP be-
cause it is collected from documents that all focus on a similar topic: Neural Networks. Nevertheless, there are sub-areas that appropriate analysis can detect.

Another difference between this and the DBLP dataset is that in the latter, each document was given as a bag of words, whereas in the IJCNN dataset, each document is an actual text with sentences, paragraphs, etc. However, before it is used by TExPLAN, it is preprocessed to remove stop words such as prepositions, pronouns, conjunctions, etc. Still, the resulting set of words represent a much broader sample than those in DBLP, which comprise mainly technical words.

We start by constructing the term network in a similar fashion to the previous two datasets: The nodes in the network are the words in the text corpora, and two words are connected if they appear in at least one document. The weight of the connection between two words reflects the number of times they appear together in the same document. In Figure 3.15 we show the term network for the IJCNN dataset.

Similar to the previous datasets, we first look at the term network properties of the dataset and plot the results in figures 3.16 and 3.17. The degree distribution for the IJCNN dataset is slightly bimodal as for the DBLP dataset. However, unlike the DBLP dataset but consistent with the NSF dataset, the clustering coefficient distribution is strongly bimodal. Both datasets are collections of scientific publications where there is an overall theme that can describe them. In the IJCNN dataset, the publications are about neural networks, and in the DBLP, the publications are about machine learning and knowledge extraction.

In Figure 3.18 we show the top 1% of the terms for each centrality mea-
sure with the aim of understanding the nature of high centrality terms. The figure indicates that the term \textit{classifier} has the highest degree, closeness, and eigenvector centralities, and the term \textit{prediction} has the highest betweenness centrality. In general the terms with high centrality values all play significant role roles in this dataset.

Once the term network is constructed, we extract communities of terms. Figure 3.19 shows the term network where nodes from the same community have the same color. From the figure we can see that the communities are not clearly segregated. The reason behind this is the nature of the dataset which, like the DBLP case, does not have clear boundaries.

To display the topics we generate a word cloud for each one of the term communities in Figure 3.20, where the size of the word is its hub value in the topic term network. Although the articles in this dataset are all related
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

(a) Degree Distribution

(b) Degree Distribution on a log-log Scale

(c) Clustering Coefficient Distribution

(d) Clustering Coefficient Rank

Figure 3.16: IJCNN dataset term network properties

<table>
<thead>
<tr>
<th>Term Rank</th>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
<th>Topic 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank 1</td>
<td>classifier</td>
<td>neurons</td>
<td>control</td>
<td>image</td>
<td>forecasting</td>
<td>model</td>
</tr>
<tr>
<td>Rank 2</td>
<td>data</td>
<td>spiking</td>
<td>optimal</td>
<td>recognition</td>
<td>series</td>
<td>visual</td>
</tr>
<tr>
<td>Rank 3</td>
<td>classification</td>
<td>implementation</td>
<td>nonlinear</td>
<td>face</td>
<td>prediction</td>
<td>brain</td>
</tr>
<tr>
<td>Rank 4</td>
<td>clustering</td>
<td>synapses</td>
<td>dynamic</td>
<td>extraction</td>
<td>time</td>
<td>object</td>
</tr>
<tr>
<td>Rank 5</td>
<td>features</td>
<td>hardware</td>
<td>estimation</td>
<td>facial</td>
<td>electricity</td>
<td>robot</td>
</tr>
</tbody>
</table>

Table 3.4: Top 5 terms in each topic for the IJCNN dataset.

to neural networks, we can see that the network-based approach for topic extraction was able to find topics in the IJCNN dataset that discuss different aspects of research in the neural network world. Table 3.4 shows the five terms with the highest degree centrality in every topic.

So far we have looked at three different datasets, one of which had disjoint
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

Figure 3.17: IJCNN dataset term network centrality measures
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

(a) Degree Centrality Distribution  (b) Betweenness Centrality

(c) Closeness Centrality  (d) Eigenvector Centrality

Figure 3.18: IJCNN Terms with High Centrality Values

Figure 3.19: Term network of the IJCNN dataset with community labeling
topics with very distinctive terms, and the other two with topics that would be expected to share terms. However, the use of partitioning forced terms to be assigned to unique topics. This problem is overcome in Stage II of the TExPLAN algorithm, where the disjoint partitions are extended to allow overlaps.

3.1.5 University of South Florida Word Association Dataset Partitions

All three datasets investigated so far were based on documents created by researchers: In the NSF dataset case, the documents were proposals for grants, and in the DBLP and IJCNN cases, scientific publications. The goal of the authors in each case was to describe scientific methods and results.
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

The term networks for the datasets were created by the terms that appear in the documents, and two terms were connected if they appeared together in at least one document. In this section, we consider a very different kind of dataset which does not come from documents at all, but from experiments with human subjects who are probed to elicit word associations in their minds.

The TExPLAN approach is based primarily on the idea that co-occurrence of words in text reflects the association of those words in the minds of authors. In this framework, the knowledge related to each topic is organized in the mind in the form of associations between significant terms, and these mental associative networks representing knowledge are used to generate writing about those topics. The term networks being constructed from documents are, therefore, seen as an approximate “reverse engineering” of these mental epistemic networks [77, 55, 54]. However, psychologists have also conducted explicit experiments to directly access associations in the minds of individuals, leading to widely-used association norms datasets [148, 98, 100]. Of these, the most well-known is the University of South Florida (USF) association norms dataset (available from http://web.usf.edu/FreeAssociation/). This dataset was collected over a period of almost 20 years by having more than 6,000 individuals respond to cue word stimuli by producing the first associated word that came to their mind. Over the course of the project, 5,019 cue words were used to elicit more than three-quarters of a million responses. This dataset is regarded as the gold standard of association data in the English language for single word associations.

For this type of dataset, methods like LDA can not be applied, since the notion of a document is not defined. However, TExPLAN is directly applicable. Essentially, the analysis would result in identifying distinct epistemic
topics across the minds of the individuals who participated in the test. If they were a representative sample (as claimed), these topics could be seen as normative for the English-speaking U.S. population. Other studies have also applied network clustering methods to this dataset to identify such topics [161, 5].

To construct the term network for this dataset we connect terms that were at least associated by one participant. The edge weight between terms reflects how many participants associated them together. The final network is shown in Figure 3.21. This network has a total of 10,619 words that come from a wide variety of topics. In the experiment, the participants were given a total of 5019 cue words, and they responded with words that may or may not be cues. The nature of the data will result in a directed network. However,
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

For the purposes of this analysis, we only consider the undirected version of the network, and we represent any word that appeared, cue or not cue, as a node in the network. We look at the same network properties that were analyzed for the previous datasets. In Figure 3.22 we plot the degree distribution and the clustering coefficient distribution for the term network. From the degree distribution we can see that it is significantly more bimodal than the DBLP and IJCNN cases, and is also different in the sense that the first peak is with nodes with a very low degree in the range of 1-5. This peak was much broader for the DBLP and IJCNN datasets. When plotting the node degree ranks on a log-log scale, we observe a broad range with a linear shape, indicating that the degree distribution is at least partially power law. This has also been observed in other studies [203, 128]. This suggests a power-law like behavior in that region, which was not seen in the previous three term networks.

To complete the analysis of the term network, we calculate the node centrality values discussed in Section ???. In Figure 3.23 we show the distribution and the rank plot for each of the four measures. One thing that stands out is the closeness centrality distribution. It is similar to the DBLP, and IJCNN in the sense that there are two peaks, but it is different because the two peak are quite far from each other.

Figure 3.24 shows word clouds for words with the highest 1% of centrality values in the term network. The centrality measures for this dataset produced different top ranked terms which was not the case in the previous three datasets. The top ranked word in degree centrality is Food, the top ranked in betweenness centrality is Money, and the the top terms for closeness and eigenvector centralities are Good and Money respectively. Looking
at a wider set of high centrality words, it is clear that there is a fairly consistent list across the four metrics, but each one weighs the words differently. The deeper ramifications of this will be studied in future investigations.

Figure 3.25 shows the communities extracted from the term network, where the nodes in the same community have been given the same color. From the figure, it is apparent that there are many more communities in this network than the other three networks. The reason is the fact that this dataset is not really for a specific area of interest with a limited number of topics. The terms are general and diverse, and are drawn from a large number of potential topics.
3.1. TEXPLAN STAGE I: TERM NETWORK PARTITIONING

(a) Degree Centrality Distribution
(b) Degree Centrality Rank
(c) Betweenness Centrality Distribution
(d) Betweenness Centrality Rank
(e) Closeness Centrality Distribution
(f) Closeness Centrality Rank
(g) Eigenvector Centrality Distribution
(g) Eigenvector Centrality Rank

Figure 3.23: University of South Florida word association norms term network centrality measures
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

(a) Degree Centrality Distribution  (b) Betweenness Centrality
(c) Closeness Centrality  (d) Eigenvector Centrality

Figure 3.24: Word clouds for high centrality words in the University of South Florida word association norms term network

Figure 3.25: Term network of the University of Florida word association dataset with community labeling
### TEXPLAN STAGE I: TERM NETWORK PARTITIONING

#### Table 3.5: Top 5 terms in every cluster for the University of South Florida word association norms dataset

<table>
<thead>
<tr>
<th>Topic</th>
<th>Term Rank 1</th>
<th>Term Rank 2</th>
<th>Term Rank 3</th>
<th>Term Rank 4</th>
<th>Term Rank 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic 1</td>
<td>BOOK</td>
<td>READ</td>
<td>TEXT</td>
<td>NOVEL</td>
<td>LIBRARY</td>
</tr>
<tr>
<td>Topic 2</td>
<td>SLEEP</td>
<td>TIRED</td>
<td>BED</td>
<td>REST</td>
<td>NAP</td>
</tr>
<tr>
<td>Topic 3</td>
<td>TREE</td>
<td>LEAF</td>
<td>OAK</td>
<td>FOREST</td>
<td>FLOWER</td>
</tr>
<tr>
<td>Topic 4</td>
<td>LIGHT</td>
<td>DARK</td>
<td>NIGHT</td>
<td>BULB</td>
<td>DAY</td>
</tr>
<tr>
<td>Topic 5</td>
<td>SMART</td>
<td>DUMB</td>
<td>STUPID</td>
<td>INTELLIGENT</td>
<td>INTELLIGENCE</td>
</tr>
<tr>
<td>Topic 6</td>
<td>LOVE</td>
<td>HATE</td>
<td>LIKE</td>
<td>FRIEND</td>
<td>AFFECTION</td>
</tr>
<tr>
<td>Topic 7</td>
<td>SMALL</td>
<td>BIG</td>
<td>LARGE</td>
<td>SHORT</td>
<td>LITTLE</td>
</tr>
<tr>
<td>Topic 8</td>
<td>MONEY</td>
<td>CASH</td>
<td>BANK</td>
<td>SPEND</td>
<td>FUND</td>
</tr>
<tr>
<td>Topic 9</td>
<td>POLICE</td>
<td>COP</td>
<td>SHERIFF</td>
<td>OFFICER</td>
<td>DEPUTY</td>
</tr>
<tr>
<td>Topic 10</td>
<td>ANSWER</td>
<td>QUESTION</td>
<td>REPLY</td>
<td>RESPONSE</td>
<td>ASK</td>
</tr>
<tr>
<td>Topic 11</td>
<td>MATH</td>
<td>CALCULUS</td>
<td>ARITHMETIC</td>
<td>ALGEBRA</td>
<td>EQUATION</td>
</tr>
<tr>
<td>Topic 12</td>
<td>SAME</td>
<td>DIFFERENT</td>
<td>ALIKE</td>
<td>SIMILAR</td>
<td>OPPOSITE</td>
</tr>
<tr>
<td>Topic 13</td>
<td>FOOD</td>
<td>EAT</td>
<td>COOK</td>
<td>CAFETERIA</td>
<td>RESTAURANT</td>
</tr>
<tr>
<td>Topic 14</td>
<td>PAIN</td>
<td>HURT</td>
<td>OUCH</td>
<td>HEADACHE</td>
<td>INJURY</td>
</tr>
<tr>
<td>Topic 15</td>
<td>CAT</td>
<td>DOG</td>
<td>KITTEN</td>
<td>ANIMAL</td>
<td>PUPPY</td>
</tr>
<tr>
<td>Topic 16</td>
<td>NEEDLE</td>
<td>THREA</td>
<td>CLOTHES</td>
<td>SEW</td>
<td>SPOOL</td>
</tr>
<tr>
<td>Topic 17</td>
<td>CLOSE</td>
<td>OPEN</td>
<td>DOOR</td>
<td>NEAR</td>
<td>FAR</td>
</tr>
<tr>
<td>Topic 18</td>
<td>MUSIC</td>
<td>SONG</td>
<td>SING</td>
<td>TALK</td>
<td>INSTRUMENT</td>
</tr>
<tr>
<td>Topic 19</td>
<td>WATER</td>
<td>BOAT</td>
<td>OCEAN</td>
<td>POOL</td>
<td>SWIM</td>
</tr>
<tr>
<td>Topic 20</td>
<td>OLD</td>
<td>NEW</td>
<td>YOUNG</td>
<td>ANCIENT</td>
<td>ELDERLY</td>
</tr>
<tr>
<td>Topic 21</td>
<td>GOOD</td>
<td>BAD</td>
<td>EVIL</td>
<td>WRONG</td>
<td>RIGHT</td>
</tr>
<tr>
<td>Topic 22</td>
<td>STOP</td>
<td>GO</td>
<td>START</td>
<td>HALT</td>
<td>END</td>
</tr>
<tr>
<td>Topic 23</td>
<td>GUN</td>
<td>PISTOL</td>
<td>SHOOT</td>
<td>KILL</td>
<td>BULLET</td>
</tr>
<tr>
<td>Topic 24</td>
<td>COLD</td>
<td>HOT</td>
<td>ICE</td>
<td>WARM</td>
<td>CHILL</td>
</tr>
<tr>
<td>Topic 25</td>
<td>CAR</td>
<td>AUTO</td>
<td>VEHICLE</td>
<td>AUTOMOBILE</td>
<td>DASHBOARD</td>
</tr>
<tr>
<td>Topic 26</td>
<td>FALL</td>
<td>UP</td>
<td>DOWN</td>
<td>STUMBLE</td>
<td>AUTUMN</td>
</tr>
<tr>
<td>Topic 27</td>
<td>BALL</td>
<td>THROW</td>
<td>GAME</td>
<td>BASEBALL</td>
<td>RACQUET</td>
</tr>
<tr>
<td>Topic 28</td>
<td>BIRD</td>
<td>CHIRP</td>
<td>BLUEJAY</td>
<td>NEST</td>
<td>FEATHERS</td>
</tr>
<tr>
<td>Topic 29</td>
<td>PUSH</td>
<td>SHOVE</td>
<td>PULL</td>
<td>TUO</td>
<td>FIGHT</td>
</tr>
<tr>
<td>Topic 30</td>
<td>KING</td>
<td>QUEEN</td>
<td>THRONE</td>
<td>ROYALTY</td>
<td>CROWN</td>
</tr>
<tr>
<td>Topic 31</td>
<td>HAPPY</td>
<td>SAD</td>
<td>CRY</td>
<td>LAUGH</td>
<td>UNHAPPY</td>
</tr>
<tr>
<td>Topic 32</td>
<td>DRINK</td>
<td>BEER</td>
<td>DRUNK</td>
<td>ALCOHOL</td>
<td>KEG</td>
</tr>
<tr>
<td>Topic 33</td>
<td>CHURCH</td>
<td>GOD</td>
<td>RELIGION</td>
<td>PRIEST</td>
<td>CATHEDRAL</td>
</tr>
</tbody>
</table>

The total number of topics obtained by the algorithm is 33. Figures 3.263.273.28 show the word clouds for these topics. Again, the size of the term reflects its hub value in the topic term network. In table 3.5 we show the top 5 words in every topic. From the results we can see that the resulting topics are all meaningful and coherent. It is also clear that they are diverse, which is why there are 33 of them. Some are clearly more general than others, but that too is a function of the cue words used to collect the data, and the nature of the dataset.

It is clear from the metrics and the results that this dataset is different. The
Figure 3.26: Wordclouds for topics 1-16 extracted from the University of South Florida word association norms dataset.
Figure 3.27: Wordclouds for topics 17-32 extracted from the University of South Florida word association norms dataset.
network is constructed differently and the semantics of the edges are different as well. That being said, the network-based approach for topic extraction was able to extract clear and significant topics. The number of topics is larger and the topics are more diverse as expected. This dataset has demonstrated how a network-based approach could apply to networks that are not based on documents. This can be seen as an advantage for this approach over other topic extraction methods like LDA.

In this section we described a network-based approach for extracting topics from text. We also showed how this approach is applicable to word association networks, for which methods like LDA are not viable. The method described so far is a partitioning method, and each word belongs to one and only one topic. However, in languages as well as intellectual disciplines, words are often associated with multiple topics. Also, other topic extraction methods like LDA for example are able to extract overlapping topics. In the next section we describe Stage II of the TExPLAN approach that builds on the partitions obtained in Stage I to obtain topics such that each word has a membership value in each topic. The extension is also network-based and is applicable to datasets where methods like LDA cannot be applied.
3.2 Stage II: Obtaining Overlapping Topics

The method described in the previous section generates topics with non-overlapping word assignments. This is achieved by applying a modularity-maximizing graph partitioning method on the term network. While this can be useful in some situations, in general it is expected that words would be shared across topics. It is important, therefore, to develop a method that obtains meaningful topics with overlapping word assignments, and occurs, for example, with LDA. The resulting topics can also be compared more fairly with LDA to assess the strengths and weaknesses of the TExPLAN approach.

In this section, we describe Stage II of the TExPLAN framework that allows each word in the vocabulary to be assigned to every topic with a weight that indicates its relevance for that topic.

3.2.1 Algorithm Description

In the network-based approach for topic extraction, we look at text from a network perspective, where words are treated as nodes and are connected based on their co-occurrence in documents. The partitioning process starts by defining a term network $G = (V, E)$ for the text corpora, where words are nodes $V$, and edges $E$ are links between nodes. Depending on the dataset in question, the edges in the term network will have a different meaning. For instance, in the publication datasets, two words are connected if they appear together at least in one document. The weight of the edge between the two words reflects the number of documents in which they both appear. On the other hand, words in datasets like USFWAN two terms are connected if at least one participant associated them together. The edge weight in this case
is the number of participants who associated the two terms together. Regardless of the method used to create the term network, the result is a network in which the nodes are terms and the edge weights represent strength of association between the terms. A topic $k$ is conceptualized as a subnetwork $G_k = (V_k, E_k)$ where $V_k \subset V$ and $E_k \subset E$, where the subnetwork – called a topic network – captures the epistemic content of the topic in associative form. The edge weights in the topic network $G_k$ represent the strength of association between words in the context of that topic, and the importance of word $w$ in topic $k$ is indicated by its hub value $\Gamma[w][k]$ in network $G_k$. Stage I has specified a process which takes a term network $G = (V, E)$, and generates $N_k$ subnetworks $G_k = (V_k, E_k)$ such that:

- $\mathcal{L}(G = (V, E)) \rightarrow S_v = \{V_1, V_2, ..., V_{N_k}\}$
- $V_1 \cup V_2 \cup ... \cup V_{N_k} = V$
- $V_1 \cap V_2 \cap ... \cap V_{N_k} = \emptyset$

The main goal of the Stage II process is to define a membership value for every word $w$ in all topics, and not just topic $k$ for which it belongs. To determine a word's value in topics other than the one to which it was initially assigned by Stage I processing, we quantify the value of its connections to important words in all other topics. The intuition behind this is that if word $w$, which was initially assigned to topic $k$, appears frequently with the word $z$, which belongs to topic $l$, then the value for $w$ in $l$ should take into account the importance of $z$ in $l$, and the edge weight between $w$ and $z$ in $G$. The more $w$ connects to important words in topic $l$, the higher its membership value will be in the topic. To define this mathematically, we construct two matrices: 1)
3.2. STAGE II: OBTAINING OVERLAPPING TOPICS

The co-occurrence matrix $W = N_W \times N_W$ where $W[i][j]$ is the edge weight between node $i$ and node $j$ in $G$, and 2) Matrix $\Gamma = N_W \times N_K$ that has the hub values for nodes in their corresponding topic networks, defined as follows:

$$\Gamma[i][k] = \begin{cases} 
\text{hub}(i, G_k) & \text{if } i \in V_k \\
0 & \text{otherwise}
\end{cases}$$

where $\text{hub}(i, G_k)$ is the hub value for node $i$ in network $G_k$. Note that these hub values are based on the disjoint topic networks produced by Stage I.

From matrices $W$ and $\Gamma$, matrix $B$ representing the value of each word in each topic is constructed as:

$$B = W\Gamma$$

$B$ is a $N_W \times N_K$ matrix where $N_W$ is the number of words in the dataset, $N_K$ is the number of topics, and $B[w][k]$ defines the value of word $w$ in topic $k$.

**Algorithm 2** Node-Topic Participation

1: **procedure** Topic Participation($N, T, H$)
2: \hspace{1cm} $B \leftarrow W \times K$ matrix
3: \hspace{1cm} $V \leftarrow$ nodes in $N$
4: \hspace{1cm} **for** each $v \in V$ do
5: \hspace{2cm} $G^N_v \leftarrow \text{neighbors}(v, N)$ \hspace{1cm} $\triangleright G^N_v$ is the neighbors of $v$ in $N$
6: \hspace{2cm} **for** each $g \in G^N_v$ do
7: \hspace{3cm} $s_g \leftarrow \text{seed topic of node } g$
8: \hspace{3cm} $B[v][s_g] = B[v][s_g] + (e_N(v, g) \times \Gamma[g][s_g]) \triangleright e(v, g)$ is the weight of the edge between $v$ and $g$ in network $G$
9: \hspace{2cm} **end for**
10: **end for**
11: **end procedure**

In Algorithm 2 we describe the process by which we construct matrix $B$, which we use then derive two matrices as follows:
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

• $H^W$: This matrix is obtained by normalizing every row in $B$ to sum to 1. This matrix indicates the degree to which each word belongs to each topic.

• $H^K$: This matrix is obtained by normalizing every column in $B$ to sum to 1. This matrix shows how much of each topic is represented by each word.

Matrices $H^W$ and $H^K$ tell us different things about the words and topics, and both are useful as we will see in the remainder of this chapter.

As indicated above, matrix $H^W$ describes how words are distributed across the different topics, and $H^K$ describes the relative importance for the different words in each particular topic. We applied this method to the disjoint topics obtained for each of the four datasets as described in the previous section to now obtain overlapping topics for the same datasets. From these topics, new word-clouds were generated, but unlike the ones shown earlier, these could share words. The size of the words in the disjoint word-clouds had been based on their hub values within their unique topic topic networks, but in the new word-clouds, the size is based on the values captured in $H^K$. Also, to get a better idea about how each word is being distributed across topics, we calculate the entropy of the word’s vector in matrix $H^W$ using Equation 3.4. A word that is concentrated in a few topics has a low entropy value, while one that is distributed across many topics has higher entropy. Thus, the entropy value becomes a measure for the generality or specificity of each word within the dataset, or more broadly, a measure of its epistemic diversity.

Entropy is mathematically defined as follows:
3.2. STAGE II: OBTAINING OVERLAPPING TOPICS

The next four subsections summarize the main results of Stage II processing for each of the four datasets.

3.2.2 Overlapping Topics in the NSF Dataset

The first dataset we applied this extension to is the NSF dataset. This dataset has topics that are not expected to overlap significantly. The word-clouds in Figure 3.29 display the overlapping topics extracted. As expected, the results do not look very different from those for disjoint topics because of the nature of the dataset. One notable difference is in Topic 3 where the most important word changed from *Policy* to *Model*.

![Figure 3.29: Wordclouds for overlapping topics extracted from the NSF dataset](image)

Figure 3.29: Wordclouds for overlapping topics extracted from the NSF dataset

\[
E(q) = - \sum q_i \log q_i \tag{3.4}
\]

The next four subsections summarize the main results of Stage II processing for each of the four datasets.
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

Figure 3.30: Word entropies for the NSF dataset

Figure 3.31: Hub value vs. entropy for terms in the NSF dataset term network
To determine how words are being distributed across topics, we plot the entropy distribution for words in the NSF dataset. Figure 3.30 shows that there is a high number of words that are diverse across topics. To tie the entropy value of a word to its importance in the topic, we plot the entropy value versus the hub value for every word in Figure 3.31.

From Figure 3.31 we can see that important words come from a wide range of entropy values. This suggests that some topics tend to be more diverse than others, but there are topics that are very narrowly focused. It also appears that words with very low entropy are terms that are not very important in their main topic. The high entropy high hub-value words are terms like Data, Model, etc., that are significant in many topics even when the topics are quite distinct. In Stage I, these terms were forced into a particular topic, but now their general significance can become apparent. Of particular interest are words that have relatively lower entropy but very high hub values like Algebra and Geometry: These are words that are, in fact, specific to one or two topics, and especially important in at least one of them. In Figure 3.32 we show the relationship between the entropy value and other network metrics for each word in the term network.

From the plots in Figure 3.32, it is noticeable that higher entropy usually leads to higher centrality values. This is because when the word is more diverse, it plays a part in more than one topic. However, this is not the case for clustering coefficient. The more diverse word \( w \) is, the more it is connected to words from different topics. Nodes from different topics are less likely to be connected to each other, which results in a lower clustering coefficient for \( w \).

The next dataset we look at is the DBLP dataset, for which we expect to
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

Betweenness Centrality vs. Entropy

Closeness Centrality vs. Entropy

Eigenvector Centrality vs. Entropy

Clustering Coefficient vs. Entropy

Figure 3.32: Node property values vs. entropy for terms in the NSF dataset term network

see more overlap between topics.

3.2.3 Overlapping Topics in the DBLP Dataset

In the case of the DBLP dataset, the change from the case of disjoint topics is more significant. As shown in Figure 3.33 the term Data appears in most of the topics once overlap is allowed. One major change is that Data also becomes the most important term in the in the data mining topic in addition to the databases topic, which is understandable. In the partitioning case, Data had to appear in one topic, and it was more connected to terms in the database topic and appeared there. With overlap allowed, Data turns out to
be strongly connected to many important words in the data mining topic as well. This is important because the meaning of Data in databases is a little different than the meaning in data mining, so what the algorithm has done can also be seen as a step in a word sense disambiguation process.

To determine whether indeed this dataset has a more overlapping structure, we look at the entropy value distribution. We notice that this data set has a higher percentage of its words with high entropies. As more entropy means more participation in more topics, this dataset has a higher overlap rate than the NSF data set. Figure 3.35 compares the entropy distribution for the DBLP and NSF datasets. As we can see from the figure, the DBLP dataset has more words with higher entropy.

We also plot the relationship between the different centrality values, clus-
Figure 3.34: DBLP dataset word entropies

Figure 3.35: Entropy Distribution for NSF and DBLP datasets
3.2. **STAGE II: OBTAINING OVERLAPPING TOPICS**

Figure 3.36: Cumulative percentage of words with entropy values up to specific levels in the NSF and DBLP datasets

Figure 3.37: Entropy value vs. hub for terms in the DBLP dataset term network
Betweenness Centrality vs. Entropy  
Closeness Centrality vs. Entropy  
Eigenvector Centrality vs. Entropy  
Clustering Coefficient vs. Entropy

Figure 3.38: Node property values vs. entropy for terms in the DBLP dataset term network

By comparing betweenness centrality and entropy, we can see that words with high centrality values tend to have higher entropy. This aligns well with our explanation of how important words in this dataset are more diverse. For the clustering coefficient measure, we note again that high entropy words have lower clustering coefficient values. This is similar to the NSF case as it follows from the nature of the clustering coefficient. The higher the entropy the more diverse the word is, but a diverse word will have neighbors in many topics which will, therefore, not be connected to each other.
3.2.4 Overlapping Topics in the IJCNN Dataset

The third dataset we consider is the IJCNN publication dataset. Similar to the DBLP dataset, we expect this dataset to have more overlap in topics than the NSF dataset. Similar to the previous datasets, once the term network is constructed, and the seed topics are identified, we apply Algorithm 2 to extract the overlapping topics. In Figure 3.39 we show the word clouds for each topic, with the size of the word reflecting its importance according to matrix $H^k$.

When comparing the word clouds in Figure 3.20 and those in Figure 3.39 we notice that the important words have not changed much between the two cases. However, there are some changes in the ranking of the words in some topics. In topic 4, the term feature became more important than the term detection, and image continues to be the term with the most importance. In topic 5 electricity became more important than it was in the disjoint topics case. The term prediction has significantly dropped in importance however. In topic 6, the most important word is now neurons followed by learning instead of model.

To check for overlap between topics, we look at the entropy distribution in Figure 3.34. The figure shows that, as with the other datasets, the number of words with high entropy is significantly higher than those with lower entropies.

In terms of the relationship between term entropy and significance in the original seed cluster, the IJCNN behaves similarly to the DBLP dataset. In Figure 3.37 we see that important terms tend to have a higher entropy value. However, there are two words that stand out; forecasting and series. These two words have relatively low entropy values, but still managed to have high
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

Figure 3.39: Wordclouds for overlapping topics extracted from the IJCNN dataset

Figure 3.40: IJCNN dataset word entropies
3.2. STAGE II: OBTAINING OVERLAPPING TOPICS

Figure 3.41: Hub value vs. entropy for terms in the IJCNN dataset term network

importance values. This suggests that they are more specific to their topic in the disjoint case.

As for the previous two datasets, we also plot the relationship between term entropy and other network metrics to determine the relationship between them. From Figure 3.42 we notice behavior very similar to the DBLP dataset. High centrality words have high entropy, and high entropy words have low clustering coefficient. As noted in the DBLP case, nodes that are diverse will tend be connected to more nodes that are not themselves interconnected.

The results for the IJCNN dataset suggest that it has more similarity with the DBLP dataset than with the NSF dataset, probably reflecting the fact that the IJCNN and DBLP datasets are both based on conference papers makes them more similar.
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

Betweenness Centrality vs. Entropy Closeness Centrality vs. Entropy

Eigenvector Centrality vs. Entropy Clustering Coefficient vs. Entropy

Figure 3.42: Node property values vs. entropy for terms in the IJCNN dataset term network

3.2.5 Overlapping Topics in the University of South Florida Word Association Dataset

The USFWAN dataset is quite different from the previous three datasets in several ways. It is not based on scientific publications, and it is not based on any documents. As mentioned previously, topic extraction methods like LDA cannot be applied to it because it is not based on documents. In this section we show how the topics TExPLAN is able to extract from this dataset.

As before, we start by constructing the term network and apply the partitioning method to it to extract seed topics. The seed topics for this dataset
3.2. STAGE II: OBTAINING OVERLAPPING TOPICS

Once the partitions are defined, we apply the process described in Section 3.2 and extract membership values for all the words in all the topics. In Figure 3.43 we show the distribution of entropy values in the network, and it is very clear that most of the nodes have extremely low entropies suggesting they are not diverse at all. In fact, the word clouds between the partitioning case and the overlapping case are almost identical.

This dataset also behaves very differently from the previous three datasets in terms of how word entropies relate to their hub values in their primary (disjoint) topics. In Figure 3.44 we see that important terms tend to have lower entropy values. One word that has a relatively higher entropy, and is still considered important is Money!

Figure 3.45 shows the relationship between entropy and other network metrics. Again, significant differences can be observed between this dataset and all other datasets. In the previous datasets, high centrality nodes had tended to be more diverse, i.e., their centrality in their primary topics was not specific to that topic. For the USFWAN dataset, the relationship between centrality and entropy is the reverse, except for closeness centrality. The pattern...
of relationship between entropy and clustering coefficient is also somewhat different, though it remains the case that words with higher clustering coefficients tend to have lower entropy.

Overall, it can be observed that the datasets based on documents tend to behave similarly and share similar characteristics. The NSF dataset is closer to the DBLP and IJCNN datasets than to the USF dataset, but the DBLP and IJCNN are closer to each other than the NSF dataset. Stage II of TExPLAN was able to extract overlapping topics that are very meaningful. The ability for terms to belong to multiple topics has allowed terms like *data* to play a more significant role in multiple topics in the DBLP dataset. It has also allowed the emergence of more important words to surface like *classifier* in topic 1 in the IJCNN dataset.
3.2. STAGE II: OBTAINING OVERLAPPING TOPICS

Betweenness Centrality vs. Entropy

Closeness Centrality vs. Entropy

Eigenvector Centrality vs. Entropy

Clustering Coefficient vs. Entropy

Figure 3.45: Node property values vs. entropy for terms in the USF dataset term network

3.2.6 Comparison of Topic Overlap in All Datasets

One of the interesting aspects of the analysis developed above is that it allows a quantitative evaluation of topic overlap in different datasets. As indicated by the word entropy distributions for the four datasets analyzed above, most words in the NSF, DBLP and IJCNN datasets tend to have fairly high entropy, indicating that they participate in several topics. In contrast, words in the USFWAN association norms data tend to have low entropy because the dataset comprises a large number of distinct topics, each characterized by a relatively limited number of words. However, the question might be asked
whether there is also some difference in the topic overlap in the first three datasets, and how that can be evaluated.

Figure 3.46 plots the cumulative percentage of words with entropies below a certain level for all four datasets. The plot for each dataset begins at 0 on the left and rises to 1 on the right, but the pattern of this rise indicates how the entropy is distributed across different levels. For the USFWAN dataset, low entropy words account for most of the rise, and the curve flattens out thereafter. The DBLP and IJCNN datasets show a fast rise towards the right edge, indicating that almost all of it is accounted for by the high entropy words. This, in turn, means that the topics overlap a lot. The curve for the NSF dataset lies between these two extremes, so while higher entropy words account for most of the rise, the rise begins at a lower entropy level. This indicates that the topics have significantly less overlap than for the DBLP and IJCNN datasets, but much more than for the USFWAN data.
3.3 Comparison with LDA

The motivation behind the network-based TExPLAN method of topic extraction is to have an approach that is applicable to any dataset that can be described in terms of word associations, and therefore provide both breadth of applicability and cognitive insight. So far we have shown that the network-based approach has done very well on two different types of term networks. One type constructed by associating words based on their co-occurrence in documents, and the other based on experimentally obtained word associations. While this is a novel approach for extracting topics from corpora, there is an extensive body of research that tackles the problem of topic extraction using statistical methods. These were described in detail in Section ??, and their advantages and disadvantages were discussed. Among these approaches, topic models stand out as the most widely used. Topic modeling uses probabilistic techniques to identify hidden structure in documents. They are mainly used to analyze large datasets, and have proved to be successful at doing that. The most well known example of topic models is Latent Dirichlet Allocation (LDA). In LDA, the main assumption is that every document is a mixture of topics, and each topic is a distribution over the vocabulary. This is similar to other techniques like Probabilistic Latent Semantic Analysis (pLSA), but LDA assumes that the multinomial distributions of topics and words have Dirichlet priors. LDA considers documents to be bags of words, where the order of the word in the document does not matter. All that matters is the number of times the word appear in the document. Topics are then inferred as word distributions based on their co-occurrence in documents. Thus, the basic information used by LDA is similar to that used by TExPLAN,
but LDA infers the parameters of multinomial word and topic distributions, while TExPLAN infers topics and word membership in topics based on properties of word association networks. A more detailed description of LDA and its extensions can be found in Section ??

One challenging aspect of using a technique like TExPLAN or LDA is evaluating the quality of the results. These are unsupervised techniques and the ground truth is simply unknown, precluding the use of standard validation methods. Both of the techniques – TExPLAN and LDA – represent topics in terms of the weight of words in those topics, so one way to do a relative evaluation is to look at word clouds for the topics the algorithms produce on the same datasets.

One of the difficulties when applying LDA is to determine the number of topics. Clearly, the quality of topics extracted depends on the number of topics, but there is no formal or automatic way to determine the optimal number formally. To chose a number for purposes of comparison, we applied the process proposed by Arun et al [12] for finding the best number of topics for LDA. In general the method calculates KL-Divergence for the LDA topics, and higher values indicate poorer topics. Thus we choose a topic number where this value is low.

The LDA implementation we use for this task is a Python library developed by Nakatani Shuyo at Cybozu Labs [91]. In this implementation, terms are lemmatized using a set of WordNet morphology functions implemented in the Natural Language Toolkit (NLTK) library [22]. The lemmatization process removes inflectional endings from individual words to produce a form that is in WordNet. More about this process can be found in [136]. It is important to mention that TExPLAN does not lemmatize words, and every form
3.3. COMPARISON WITH LDA

Figure 3.47: Determining the number of topics of LDA for the NSF Dataset

of the word, which appears in the corpus, is considered a unique word. The motivation behind this decision is to design a process that requires minimal preprocessing of the documents. However, experience has shown that this diversity in fact improves the process of topic inference.

To compare LDA to TExPLAN, we apply it to three of the four datasets we used so far. LDA cannot be applied to the USFWAN dataset because that dataset does not comprise documents.

3.3.1 LDA Results for the NSF Dataset

Figure 3.47 shows the result of applying the approach of Arun et al to the NSF dataset. The goal is to identify the dips in the KL-Divergence value, and use these as a heuristic for the number of topics in the dataset. For the NSF case, we see that the best number of topics is 5. The word clouds for the 5 topics are shown in Figure 3.48. LDA was also able to extract the same 5 topics that were extracted using the network-based approach.
3.3.2 LDA Results for the DBLP Dataset

The second dataset we apply LDA to is the DBLP dataset. This dataset has more of an overlapping structure, and the documents used discuss topics that are similar. This overlap might be expected to make this dataset a more challenging one for the network-based approach method. To apply LDA we first start by determining the number of topics to extract. Figure 3.49 shows the results of the matrix factorization method. According to the figure, the best three topic numbers are 3, 5, and 9. We run LDA for these values on the DBLP dataset and show the results in figures 3.50-3.52.

In the case where the number of LDA topics is set to 3 we can see general topics. The topics do not map to an area of interest in the DBLP area directly. Topic 1 is about data, databases and mining, which can be seen as more
3.3. COMPARISON WITH LDA

Figure 3.49: Determining the Number of Topics of LDA for the DBLP

Figure 3.50: DBLP LDA Topics K=3
than one topic. The case is similar in topic 2 where learning and classification can be one topic, however image and logic should probably fall into another topic. From the word clouds, and the knowledge we have about the dataset, we can conclude that the number of topics should probably be higher than 3. In the case where the number of LDA topics is set to 5, we can see some rough mapping between the areas of research in the DBLP dataset and the LDA topics. Topic 1 is about databases, topic two is about data mining, topic 3 is about learning, topic 4 is about information retrieval from web pages, and topic 5 is about knowledge based systems. One thing that stands out is the missing topic about images. The network-based approach was able to automatically extract a topic that deals with image analysis. From the figure in 3.49 we know that for LDA the number of topic being 9 is also a good solution.
3.3. COMPARISON WITH LDA

Figure 3.52: DBLP LDA Topics K=9
In Figure 3.52 we show the word clouds for the 9 LDA topics. In this case, LDA was able to extract the topic about images. However, the topic about databases was split into two; topic 3 and topic 9. Also topic 6 does not map to any area of interest. It is clear that LDA needed more resolution to extract the image analysis topic, which the network-based method was able to extract automatically.

As we described earlier, the LDA implantation we use lemmatizes words and tries to produce a form that is available in WordNet. To demonstrate how LDA would behave without the lemmatization step, we show the word-clouds of the topics extracted from all the words in the documents in figure 3.53. As we can see from the figure, the quality of the topics extracted without lemmatizing words has dropped significantly. The word data became the most significant word in two topics, and the word based became the dominant word in System Knowledge topic. In the case where words were lemmatized, the word data is significant in two topics, Data Mining and Databases, but it is not the most significant in either topic. These results should be compared with the results reported for TExPLAN, which required no lemmatization.

From the results of applying LDA to the DBLP dataset, we can see that the network-based approach is a good method to extract topics, and it was able to extract topics that LDA was not able to do while keeping the overall number of topics small.

### 3.3.3 LDA Results for the IJCNN Dataset

The last dataset we apply LDA to is the IJCNN dataset. This dataset has more of an overlapping structure, and the documents used discuss topics that are even more similar than those of the DBLP dataset. To determining
3.3. COMPARISON WITH LDA

Figure 3.53: DBLP LDA Topics K=5 for all the words in the corpus

the number of topics for LDA, we apply the KL divergence heuristic and the results are shown in Figure 3.54. According to the figure, the best three topic numbers are 6, 13, and 17. We run LDA for these values on the IJCNN dataset and show the results in figures 3.55-3.57.

In the case where the number of LDA topics is set to 6 we can some similarity to the topics extracted using the network-based approach. One major difference is that LDA was not able to extract the topic about time-series forecasting, which was topic 5 in Figure 3.39. Another difference is the missing topic about the neurons, which was topic 2 in Figure 3.39.

Overall, the comparison between TExPLAN and LDA topic extraction demonstrates three clear strengths for TExPLAN:

1. In TExPLAN, the topics emerge naturally from the structure of the word networks, and therefore reflect the actual semantic structure of the
CHAPTER 3. TOPIC EXTRACTION - A NETWORK BASED APPROACH

Figure 3.54: Determining the Number of Topics of LDA for the IJCNN

Figure 3.55: IJCNN LDA Topics K=6
3.3. COMPARISON WITH LDA

Figure 3.56: IJCNN LDA Topics K=13
Figure 3.57: IJCNN LDA Topics K=17
3.3. COMPARISON WITH LDA

data. The quality of topics determined by LDA depends strongly on the number of topics, but LDA offers no clear guidance on this. As a result, the user is forced to make a heuristic choice and live with its consequences.

2. The TExPLAN algorithm runs much faster than LDA on the same machine because it is inherently a lightweight algorithm. The most computationally intensive step in TExPLAN is the partitioning process using the Louvain algorithm, which is believed to be \(O(N\log N)\), where \(N\) is the size of the vocabulary. LDA uses Gibbs sampling for its inference step, and typically many cycles of this have to be run before the algorithm can be assumed to have converged.

3. While not been reported in this dissertation, informal observation of multiple runs with TExPLAN and LDA on the same datasets shows that TExPLAN always recovers virtually identical topics, while the topics extracted by LDA can vary significantly. Thus, TExPLAN is also more robust than LDA as a topic extraction algorithm. However, this issue requires further systematic investigation.

In terms of subjective quality determined from the word clouds, it is clear that TExPLAN extracted topics that were at least as good – and often better – than LDA. Thus, the TExPLAN approach, in addition to being more consistent with the workings of human cognition, also seems to provide clear computational advantages.
3.4 Conclusion

The main goal of this thesis is to define an epistemic space where documents, authors, and venues can be embedded and analyzed. The first step towards achieving this goal is to determine the dimensions of such a space, then define the process by which entities can be embedded. As the goal is to build an epistemic space, the dimensions need to capture the semantic structure of the entities. Topics are natural candidates for such dimensions, and this chapter has presented a network-based approach called TExPLAN for extracting topics from text corpora and other word association datasets. A comparison of TExPLAN and LDA has shown that TExPLAN extracts more meaningful topics without requiring prior specification of the number of topics, and is more robust than LDA.

The idea of applying community extraction algorithms to extract groups of words from term networks is not unique. Pella et al [161], and Ahn et al [5] are examples of such approaches. However, these algorithms were not used for topic extraction, but only for identifying groups of related words. In the TExPLAN approach described in this chapter, we go further and determine a membership value of each node in each community, thus generating topics with specific word distributions. In the next chapter we describe the second component of this research project: Embedding documents, authors and venues in epistemic space, and analyzing them.
Chapter 4

Epistemic Space Analysis

The central idea for the research in this thesis is that of an epistemic space, i.e., a metric space in which epistemic entities such as words, texts, documents, etc. can be embedded and compared. The axes of this space represent epistemically independent dimensions, which can be identified with topics. The previous chapter presented a graph-based method for identifying topics from texts, but the topics used to define an epistemic space are independent of this and can be obtained in principle by any suitable algorithm, including LDA. However, the work in this thesis uses topics obtained using the approach presented in Chapter 3. An overview of epistemic spaces is given in Section 4.1. Section 4.2 describes how entities are embedded in the epistemic space, and Section 4.3 poses some relevant questions that can be answered using the idea of epistemic space. The goal of this chapter is to demonstrate the advantage of having a unified space where multiple type of epistemic entities can be embedded.
CHAPTER 4. EPISTEMIC SPACE ANALYSIS

4.1 Overview

One of the main motivations behind this research is to make sense of a world created by large corpora. This world contains different types of entities such as documents, authors, and (Venues) – the places where these documents are published. In order for us to understand the world of these entities, we need to create a unified epistemic space where all the different entities can be embedded. This unified space will not only allow us to answer questions about single types of entities, but also allow us to explore the relationship between different types.

In this chapter we use the DBLP dataset because, as described in Section 1.2, it provides document, author, and venue information. However, this is only one example of how the approach presented in this chapter can be used in general.

For the purposes of this research we define the epistemic space to be an $n$-dimensional space where $n$ can be any natural number. A point $x$ in this space is defined by $n$ coordinates $x_0, x_1, x_2, \ldots, x_n$, where $0 \leq x_m \leq 1$ and $\sum_{x \in x} x_i = 1$. The last condition indicates that the entities are actually embedded on a simplex within the semantic space rather than in the whole space. One consequence of doing this is that the coordinates can be interpreted naturally as probabilities, and particularly as the parameters of a multinomial distribution, though this interpretation may not always be meaningful. In general, however, the topical composition of an entity can be seen as a conserved quantity, so that any bias towards some topics much come at
the expense of others. This allows more meaningful epistemic comparisons between entities without worrying about their “amplitude” in epistemic space.

The epistemic world considered here has the following types of entities:

- **Words**: Single terms
- **Publications**: a bag of words that make up the publication
- **Authors**: the creators of the publications
- **Venues**: the places where the publications were published.

The dimensions of the epistemic space are defined to be the topics extracted from the term network using the graph-based method defined in Chapter 3. The result of the topic extraction method is a membership value for each word in every topic. We use this membership value to be the coordinate of the word in our $n$-dimensional space. This process ensures that every word is placed in the space. Embedding the other entities becomes a weighted aggregation process that is described in subsequent sections.

### 4.2 Embedding in Epistemic Space

The result of the topic extraction method described in Chapter 3 is to give the coordinates of the words in the topical space. The main matrix that captures this is $B$, where $B_{ij}$ reflects how relevant topic $j$ is to word $i$. From $B$ we derive two matrices:
• **H^W**: This matrix is obtained by normalizing every row in B to add up to 1. Thus, the entry \( H^W_{ij} = B_{ij} / \sum_k B_{jk} \) reflects the *relative relevance* of topic \( j \) for word \( i \).

• **H^K**: This matrix is obtained by normalizing every column in B. Thus, the entry \( H^K_{ij} = B_{ij} / \sum_k B_{kj} \) reflects the *relative significance* of word \( i \) for topic \( j \).

As documents are seen as bags of words, a natural extension would be to embed the documents in the space using the terms that appear in them. Once the documents are embedded in the space, we can then embed authors by aggregating their documents or terms, and we can embed venues by aggregating the publications that were published in the venues.

### 4.2.1 Embedding Documents

To embed a document in the epistemic space requires determination of its coordinates. The matrix \( D^K = \mathcal{N}_D \times \mathcal{N}_K \) represents the coordinates for all the documents and is calculated as follows:

\[
D^K = P(D^W (H^W \circ H^K))
\] (4.1)

where \( H^W \circ H^K \) is the Hadamard product, also known as the pair-wise product, and \( P \) is a function that does row normalization on the matrix. \( D^W \) is a \( \mathcal{N}_D \times \mathcal{N}_W \) matrix where \( D^W_{ij} \) is the number of times the word \( j \) appeared in document \( i \). \( \mathcal{N}_D \) is the number of documents, \( \mathcal{N}_W \) is the number of words, and \( \mathcal{N}_K \) is the number of topics. Each row in the matrix \( D^K \) represents the coordinates for that document in the space and is termed the *document topic profile* (DTP).
4.2. EMBEDDING IN EPISTEMIC SPACE

4.2.2 Embedding Authors

As with documents, embedding authors in the epistemic space requires determination of their coordinates in the space. The authors can be seen as an aggregate of their documents, or they can be seen as a bag of words for all the words they used in their documents. To capture both aspects, authors can be embedded in two ways; Document-Based or Term-Based.

Document-Based Embedding of Authors

To embed the authors in the epistemic space based on their documents, the epistemic coordinates of all the author’s documents are added and the resulting vector normalized. In mathematical terms:

\[ A^K_D = P(A^D D^K) \] (4.2)

\( P \) is a function that does row normalization on the matrix. \( A^D \) is a \( N_A \times N_D \) matrix where \( A^D_{ij} = 1 \) if author \( i \) authored document \( j \). \( N_A \) is the number of Authors and \( N_K \) is the number of topics.

Matrix \( A^K_D \) represents the coordinates for every author in the space based on documents. We also refer to every row in \( A^K_D \) as the author topic profile - document-based (ATP-DB) because it basically creates a profile for the author in the topic space.

Word-Based Embedding of Authors

The authors can also be embedded using the individual words they use. For an author, a bag-of-words can be obtained by getting all the words the author
CHAPTER 4. EPISTEMIC SPACE ANALYSIS

Figure 4.1: Distances distribution of the cosine distance between the ATP-DB and ATP-TB for all the authors

used while authoring their publications. The coordinates of the author in the space based on terms are calculated mathematically as:

\[ A^K_W = P(A^W|H^W \circ H^K) \] (4.3)

where \( A^W \) is a \( N_A \times N_W \) matrix where \( A^W_{ij} \) is the number of times author \( i \) used term \( j \). \( N_A \) is the number of authors

Matrix \( A^K_W \) represents the coordinates for every author in the space based on words. Each row in \( A^K_W \) is termed the author topic profile- word-based (ATP-WB) for a specific author.

To determine the relationship between the authors’ ATP-DB and ATP-WB, plot a distribution of distances between ATP-DB and ATP-WB for all the authors in Figure 4.1.

From Figure 4.1 it is clear that for the vast majority of the authors the
4.2. EMBEDDING IN EPISTEMIC SPACE

Figure 4.2: Number of documents vs the distance between the two author topic profiles

distance between ATP-DB and ATP-TB is very small. Figure 4.2 displays the relationship between the number of publications, and the distance between the two profiles for every author. It is clear that for authors with a large number of publications the distance is close to 0.

4.2.3 Embedding Venues

Venues, i.e., the site (e.g., journal, conference proceedings, website, etc.) where the documents in the dataset are published can be embedded in the epistemic space using methods similar to those used for authors. Every venue is seen as a collection of documents and the Venue Topical Profile (VTP) is defined as follows:

\[ V^K = P(V^D D^K) \]  

(4.4)

where \( V^D \) is a \( N_V \times N_D \) matrix where \( V^D_{ij} \) is 1 if document \( j \) was published in \( i \). \( N_V \) is the number of venues in the dataset.
4.2.4 Epistemic Diversity

One of the most interesting features of a semantic entity – document, author’s corpus or venue – is the diversity of topics at ranges over. This is significant for various reasons. For example, an author whose work covers many topics can be seen as working in an interdisciplinary manner, whereas an author whose work is dominated by one topic has a narrower focus. Such authors may serve different roles within the larger community of authors in their field, and may be seen as contributing to different aspects of it. There is an intuitive assumption that, in research domains, more interdisciplinary researchers are likelier to produce creative ideas through cross-fertilization between topics, while more focused researchers may have more depth in their expertise. However, validating such an assumption requires a way to quantify epistemic diversity in the work of individual authors. Similarly, a conference or a journal may be deemed more or less interdisciplinary based on an analysis of its epistemic diversity.

The epistemic space approach provides a natural way to quantify diversity by looking at where documents, authors and venues fall in this space. One can define a region of diversity as a region around the centroid of the epistemic simplex where the semantic entities are represented. Epistemic entities falling within this region can be regarded as more diverse. The most diverse entity is one that has equal values in all topics. Since the values of all topics for an entity add up to 1, we can treat this topic profile vector as a probability distribution and use the entropy defined in Equation 3.4 to quantify diversity.

Entropy is maximum when all \( q_i \) are equal, and minimum when one of the
4.2. EMBEDDING IN EPISTEMIC SPACE

$q_i$ is 1 and the rest 0. Thus, entropy captures exactly the notion of epistemic diversity proposed above. In the case of authors, there are two diversity values that can be measured, \textit{average publication entropy} (APE), $E_P$ and \textit{global author entropy} (GAE) $E_G$. The difference between the two measures is the level of granularity for which the entropy is calculated. The two levels of granularity are:

- \textbf{Individual Publication Level}: For this level, we calculate the entropy for each document by applying equation 3.4 to the \textit{document topic profile} (DTP), and then we take the average of that. This gives the APE for the author.

- \textbf{Global Level}: For this level, we treat all the documents created by the author as one document, and the entropy function is applied to the \textit{author topic profile - term based} ATP-TB of the author to calculate the GAE.

Using the levels of granularity defined above, the average publication entropy, $E_P(a)$, for author $a$ is calculated as follows:

$$E_P(a) = \frac{\sum_{p \in P_a} E(DTP_{p_i})}{N_D(a)}$$  \hspace{1cm} (4.5)

where $P_a$ is the set of publications which $a$ authored, and $DTP_{p_i}$ is the $DTP$ for the $i$th publication in $P_a$. $N_D(a)$ is the number of documents authored by author $a$. The global author entropy for author $a$ is calculated as:

$$E_G(a) = E(\textit{ATP} - \textit{TB}^a)$$  \hspace{1cm} (4.6)

where $\textit{ATP} - \textit{TB}^a$ is author’s $a$ ATP-TB. and $E(x)$ is defined in Equation 3.4
In the next section we will analyze different entities and ask questions that can be answered by looking at a single or multiple types of entities.

### 4.3 Epistemic Space Analysis

Embedding entities in epistemic space allows us to ask question about the similarity between entities, and identify densely occupied and empty regions. Asking questions about a single type of entities is useful, but what makes this analysis even more interesting is the fact that different types of entities can be handled uniformly, which can be used to answer questions about the relationship between different types of entities. In this section we look at a set of such questions.

#### 4.3.1 Analysis of Venues in Epistemic Space

In Section 4.2.3 we described how to place venues in the epistemic space. Here, we describe how this embedding can be used to infer useful information about venues. The goal is to understand what the placement of venues means, and to analyze how they are related. Figure 4.3 shows how each conference in the DBLP dataset is associated with different topics based on the 5-topic analysis in Chapter 3. From the figure it is clear that most conferences have one peak, representing the core topic of the conference, but every conference includes papers from all five topics.
4.3. EPISTEMIC SPACE ANALYSIS

To visualize the proximity of the venues, we use a technique called multidimensional scaling (MDS). The idea behind this technique is to place data points with multiple dimensions into a lower-dimensional space so it becomes possible to visualize it. The technique achieves this by remapping to visualization space while maintaining the relative distance between points in the full representational space, i.e., the epistemic space. Figure 4.4 displays the conferences in a two dimensional space.

By looking at Figure 4.4 we see some points (conference) that are placed near each other suggesting that the conferences are epistemically similar. Such regions, where several conferences are clustered together, can be seen as representing meaningful areas of interest (AoI). To investigate this further we applied a clustering algorithm called DBSCAN to extract clusters of conferences in the reduced space. DBSCAN is a density-based clustering algorithm used to extract data points that are located close together in a space,
and is one of the most popular in scientific studies. DBSCAN requires two parameters, $\varepsilon$ and $MinPts$ where:

- $\varepsilon$ is the maximum distance between two points that can still be seen as belonging to the same cluster
- $MinPts$ is the minimum number of points required to form a cluster

For the purpose of clustering the conference, we chose $\varepsilon$ to be the small-
**4.3. EPISTEMIC SPACE ANALYSIS**

Table 4.1: The resulted clusters from applying DBSCAN to the conferences in the DBLP dataset with $\varepsilon = 0.025$

<table>
<thead>
<tr>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Cluster 5</th>
<th>Cluster 6</th>
<th>Cluster 7</th>
<th>Cluster 8</th>
<th>Cluster 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDBT</td>
<td>ECML</td>
<td>ICDM</td>
<td>ECIR</td>
<td>AAAI</td>
<td>WWW</td>
<td>CVPR</td>
<td>CIKM</td>
<td></td>
</tr>
<tr>
<td>ICDE</td>
<td>ICML</td>
<td>KDD</td>
<td>SIGIR</td>
<td>IJCAI</td>
<td>WSDM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PODS</td>
<td>PAKDD</td>
<td>SDM</td>
<td></td>
<td></td>
<td></td>
<td>CVPR</td>
<td>CIKM</td>
<td></td>
</tr>
<tr>
<td>SIGMOD1</td>
<td>KDD</td>
<td>ICDM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLDB</td>
<td>PKDD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\varepsilon = 0.025$

Table 4.2: The resulted clusters from applying DBSCAN to the conferences in the DBLP dataset with $\varepsilon = 0.029$

<table>
<thead>
<tr>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Cluster 5</th>
<th>Cluster 6</th>
<th>Cluster 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDBT</td>
<td>ECML</td>
<td>ECIR</td>
<td>AAAI</td>
<td>WWW</td>
<td>CVPR</td>
<td></td>
</tr>
<tr>
<td>ICDE</td>
<td>ICML</td>
<td>SIGIR</td>
<td>IJCAI</td>
<td>WSDM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PODS</td>
<td>PAKDD</td>
<td>SDM</td>
<td></td>
<td></td>
<td>CVPR</td>
<td></td>
</tr>
<tr>
<td>SIGMOD</td>
<td>KDD</td>
<td>ICDM</td>
<td></td>
<td></td>
<td>CIKM</td>
<td></td>
</tr>
<tr>
<td>VLDB</td>
<td>PKDD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\varepsilon = 0.029$

est value that allows the formation of largest number of clusters of size 2 or more. Ideally we would like to minimize the number of singleton clusters. In order to determine the value of $\varepsilon$ we ran DBSCAN with a range of values and plotted the relationship between $\varepsilon$ and the number of clusters. Figure 4.5 shows the change in number of clusters of size 2 or more with the change of $\varepsilon$. There are two values of $\varepsilon$ that ensure we get the maximum number of clusters of size 2 or more; 0.025 and 0.029. The value 0.025 gives more condensed clusters, but leads to a larger number of singleton clusters. The value of 0.029 leads to a smaller number of singleton clusters, but results in less cohesive clusters. Tables 4.1 and 4.2 shows the conference-cluster membership for the two values.

By looking closer at the results in Tables 4.1 and 4.2 we see that when using $\varepsilon=0.029$ we get one large cluster that has the following conferences:
• ECML
• ICDM
• ICML
• KDD
• PAKDD
• PKDD
• SDM

In this list ECML and ICML are *Machine Learning* conferences and the rest are *Data Mining* conferences. This is indeed an acceptable cluster as a lot of Data Mining papers get published in Machine Learning conferences. In the $\varepsilon = 0.025$ case, this list is split into two clusters, one for Data Mining, and one for Machine Learning, which is clearly also meaningful. The $\varepsilon = 0.029$ case combines WWW and WSDM, which are the two conferences most closely focused on the World-Wide Web. In the $\varepsilon = 0.025$ case, however, the two conferences are split into separate singleton clusters. Since WWW is a conference that deals with the future direction of the World Wide Web and WSDM is a conference is about web search and data mining, it is understandable that these two conferences would be differentiated under stricter similarity constraints. Also looking at Figure 4.4 we can see that WWW and WSDM are not very close to each other. The other two singleton clusters contain CVPR and CIKM respectively. CVPR is a conference that deals with Image Analysis and CIKM deals with *Knowledge Management*. Although CIKM is close to the *Artificial Intelligence* cluster, it can still be seen as a
4.3. EPISTEMIC SPACE ANALYSIS

<table>
<thead>
<tr>
<th>Area of Interest</th>
<th>Conferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases</td>
<td>EDBT, ICDE, PODS, SIGMOD, VLDB</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>ECML, ICML</td>
</tr>
<tr>
<td>Data Mining</td>
<td>ICDM, KDD, PAKDD, PKDD, SDM</td>
</tr>
<tr>
<td>Information Retrieval</td>
<td>ECIR, SIGIR</td>
</tr>
<tr>
<td>Artificial Intelligence</td>
<td>AAAI, IJCAI</td>
</tr>
<tr>
<td>World Wide Web</td>
<td>WWW</td>
</tr>
<tr>
<td>Web Mining</td>
<td>WSDM</td>
</tr>
<tr>
<td>Image Analysis</td>
<td>CVPR</td>
</tr>
<tr>
<td>Knowledge Management</td>
<td>CIKM</td>
</tr>
</tbody>
</table>

Table 4.3: The interpretation of clusters resulted from applying DBSCAN on the DBLP dataset

separate area. CVPR is the only conference that deals with Image Analysis in the data set, therefore, it has its own cluster. The interpretation of the clusters obtained by applying DBSCAN on the reduced conference space can be found in Table 4.3. The fact that the conference clusters produced are so clearly meaningful demonstrates that the topics extracted by the graph-based approach described in Chapter 3 are of sufficient quality to define a useful epistemic space.

The areas (clusters) resulting from applying DBSCAN to the conference can also be embedded in the space. Just as conferences comprise collections of papers, areas can be seen as collections of conferences (or journals). Therefore, each area can also be assigned a topic profile, as shown in Figure 4.6. This also indicates why the Data Mining and Machine Learning clusters, and the WWW and WSDM conferences are merged when $\epsilon = 0.029$ is used. Figure 4.7 shows the topic profiles of the areas of interest in 2-D space after application of MDS.

In this section we showed how the epistemic space can be useful in understanding more about the venues. In the upcoming sections we show how
Figure 4.6: Topic profiles for areas of interest obtained from the DBLP dataset using DBSCAN with $\epsilon = 0.025$. 
4.3. EPISTEMIC SPACE ANALYSIS

Figure 4.7: MDS-based 2-dimensional view of AoI topic profiles for the DBLP dataset

the space can be used to answer questions about multiple types of entities like documents and venues, and authors and venues.

4.3.2 Document Analysis in the Epistemic Space

In section 4.2.1 we described how to embed documents in the epistemic space, and in Section 4.3.1 we demonstrated how such a space can be used to answer questions about entities that can be placed in it. In this section, we aim to show how this space can help us answer questions about documents. Having also embedded conferences in the same epistemic space, we also address joint questions about documents and conferences.

The first question we consider is how well papers fit in the conferences where they were presented. This is important to understand for two reasons: It indicates whether authors are publishing in conferences which are a good fit for their publications, and it helps us understand the scope and diversity of the conferences. To do this we need to know how papers are distributed
Figure 4.8: For every conference the figure displays the distribution of distances between the conference’s location, and the documents’ locations for the documents that were published in the conference.

relative to the coordinates of their conferences in the space.

Figure 4.8 shows the distribution of distances between publications and their conferences. From the figure, we can see that, for most conferences, the majority of their publications are within a small distance of the profile of the conference. This indicates that publications are indeed being published in conferences closely related to their topic of interest, and therefore have a cohesive core of publications. In Section 4.3.1, we grouped conferences into areas of interest. To get a better idea about how well documents fit the areas of interest for their respective conferences, we calculate the distance between each document and the area of interest profile of the conference in which it was published. Figure 4.9 displays the distribution of these distances in every area of interest. Similar to the document-conference distance distribution in Figure 4.8, the majority of the areas of interest have documents that are positioned close to them except two areas, Knowledge Management (KM) and Artificial Intelligence (AI). This indicates two things: 1) The conference clus-
Figure 4.9: For every area of interest, the figure displays the distribution of distances between the area’s location in epistemic space and the documents’ locations for the documents that were published in that area.

ters corresponding to the areas other than KM and AI are well-represented by their areas of interest; and 2) 1) Conferences in the KM and AI fields have a greater diversity of publications and the area of interest profiles for these clusters are less representative.
This raises the question of what is special about KM and AI? Are these areas inherently more diverse or is there something special about the authors who send papers to conferences in these areas? To answer this question, we start by looking at the diversity of these areas by plotting the diversity distribution for all the publications in these areas and in all the other areas. In Figure 4.10 the area of AI has the highest percentage of diverse publications, indicating by the strong skew of the distribution’s peak to the right. Relatively speaking, more diverse papers are being published in the area of AI than in other areas, which explains why publications in AI tend to have a larger distance from the AoI profile. However, this is not the case for KM and we have to look elsewhere to explain why KM papers fall far from the AoI profile.

If we now look at the topic values for AI and KM in Figure 4.6 we see that AI has an almost equal participation in all the topics, except topic 1. Topic 1 is about images and it only peeks in one area of interest: Image Analysis. However, when looking at KM, we see that KM peeks in two topics, 3 and 5. Topic 3 focuses on retrieving information from the web, and Topic 5 focuses on databases and queries. This indicates that the spread in KM does not come from general diversity of topics, but due to the papers coming from two distinct areas, i.e., KM is a mixture of two topics.

To summarize, although the areas of AI and KM both have a distance distribution that is flatter than other areas, this is due to different reasons. In AI, it is because of diversity across different topics, but in KM it is because the area is a hybrid between web information retrieval and databases. This insight is made possible by the epistemic space framework.

Having looked at documents ind the implications of their diversity, it is
Figure 4.10: Document diversity distribution for areas of interest.
also interesting to understand more about the authors who publish papers far from the core areas of conferences. One question is, are these papers far from the primary area of interest because they bridge other areas of interest and are published by more mature authors, or because they were published by authors with relatively low number of publications who are outliers at the conferences. To get better insight about documents that fall far from the position of their conferences’ areas of interest, we identified the furthest 500 documents and looked at the authors who published these documents. Figure 4.11 shows about 50% of the authors responsible for publications that are far from their area of interest have 5 or fewer publications, indicating that they are outsiders or beginning researchers in the general field covered by the DBLP dataset. In Figure 4.12, we look at the relationship between the documents’ distance from the area of interest versus their authors’ number of publication. The figure shows that documents that are the furthest are being authored by authors who have a relatively small number of publications. Authors with high number of publications also tend to author papers that are somewhat distant from the area of interest, but they tend not to go too far. Of course, the converse that authors with fewer publications necessary go farther from the AoI is not true, and most such authors also remain close to the conferences’ core AoI.
We should mention that a publication’s being far from the area of interest for the conference does not mean that the publication itself is diverse. An author can publish a paper that is solely focused on mining web text, but publish it in an artificial intelligence conference. Such a paper would probably fit better in a Web Mining conference, but some conferences allow for some papers that are not in its mainstream to be submitted and published. To understand this better, we measure the diversity of individual publications that are far from the area of the conference in which they were published. Figure 4.13 shows the diversity distribution of documents that are far from their areas of interest and the diversity distribution of the documents that are close to their areas of interest. This indicates that, in fact, a lot of publications that are atypical for their conferences are also more diverse, and are probably being published where they are because either their diversity allows them to connect with some aspect of the conference, or there is no conference that truly matches their content due to its diversity. It is worth noting that, in all cases, we are measuring diversity relative to the originally inferred topics, so that the dominant topical structure defines what is a pure topic and what is a mixture.

In this section we answered questions about documents in the space. We also shed some light on why conferences in different areas of interest tend to have publications that are not close to their mainstream. Understanding document diversity and tying this to the diversity of venues is one advantage of have a unified epistemic space where documents and venues can be embedded. In the next section we look at authors in the space, and address questions about them.
CHAPTER 4. EPISTEMIC SPACE ANALYSIS

Figure 4.12: Distance of far-from-AoI documents vs. Authors’ number of publications

Figure 4.13: Diversity distribution for documents that are far from their areas of interest, and documents that are close to their areas of interest
4.3.3 Author Analysis in Epistemic Space

Authors are another type of entity that can be embedded in the epistemic space, as described in section 4.2.2. As shown there, authors can be embedded in two different ways but they are quite similar in practice. In this section the Author Topic Profile (ATP) will refer to the Author Topic Profile Document Based (ATP-DB). We start by looking at some descriptive statistics about the authors, and then get into analyzing their locations in epistemic space.

Figure 4.14 shows that most of the authors have a small number of publications, and therefore only participate in small number of conferences. A few authors, however, have a large number of publications across many conferences, allowing them potentially to participate in several areas of interest. For such authors, it is interesting to know if they still tend to publish more in a focused area of their expertise or if they are epistemically more diverse? To answer this question, we first look at all authors’ diversity, which we defined in Section 4.2.4. For a quick recap, the two measures for author’s diversity are:

- **Global Author Entropy** measured by calculating entropy on the author’s ATP-TB
- **Average Publication Entropy** measured by calculating the entropy of all the DTPs for the author and taking their average.

The first question we consider is how the two entropy measures relate to each other, and if they are related to the number of publications for the author. Figure 4.15 shows the relationship between the global author entropy and the average publication entropy for every author, where the size of the
Authors Statistics

Figure 4.14: Author statistics for the DBLP dataset. Top: Rank plot of the number of papers published by individual authors. Middle: Rank plot of the number of conferences attended by individual authors. Bottom: Rank plot of the number of areas of interest for individual authors.
4.3. EPISTEMIC SPACE ANALYSIS

Figure 4.15: Global Author Entropy vs. Average Publication Entropy for authors

Figure 4.16: The difference between GAE and APE vs. number of publications for authors
Figure 4.17: Global Author Entropy and Average Publication Entropy vs. publication statistics for authors

point reflects the number of publications the author has. The strong positive correlation is expected as authors with more diverse publications, will be epistemically more diverse. The figure indicates that authors with relatively large number of publications fall in a narrower range for the entropies. Figure 4.16 shows that the range of values for the difference between global entropy and average entropy for authors gets smaller when the number of publications increases.

Figure 4.17 show the relationship for each of the entropy measures to the number of publications, number of conferences, and number of areas of interest for each author. From this figure, it is clear that as each of these measures of author participation increase, the range spanned by authors for both entropies narrows monotonically, and converges to between 2.1 and 2.2 at the high end. To get a better understanding of this, we look in more detail at a subset of authors considered to be prominent. To define this subset, we assign a status to every author based on their number of publications in a conference. We define an author $a$ to be prominent in a conference $c$ if they publish considerably more papers in the conference than the rest of the
4.3. EPISTEMIC SPACE ANALYSIS

Figure 4.18: Global Author Entropy and Average Publication Entropy vs. publication statistics for prominent authors

In mathematical terms:

\[
D(a, c) = \begin{cases} 
1 & \text{if } P(a, c) \geq \theta_c^D \\
0 & \text{otherwise}
\end{cases}
\]

where \(P(a, c)\) is the number of publications author \(a\) published in conference \(c\), and \(\theta_c^D\) is the minimum number of papers needed in the conference for the author to be considered prominent. For \(\theta_c^D\) we standardized the number of publications for each author who participated in the conference and set \(\theta_c^D\) to be three standard deviations above the mean. An author who is prominent in any conference is included in the subset of prominent authors.

Figure 4.18 shows how global author entropy and average publication entropy is related to number of publications, number of conferences in which the author has a prominent status, and the number of areas where they have prominent status. Similar to Figure 4.17, it is clear that the average publication entropy converges to a value around 2.1 and the global entropy to near
2.2. The question is what type of a behavior leads to this value of entropy.

To reach a prominent status in multiple areas of research, an author can follow two possible strategies:

- **Strategy I**: In this strategy, the author would write a large number of individually interdisciplinary papers, each of which is potentially suitable for several conferences, and thus reach many conferences.

- **Strategy II**: In this strategy, the author write a large number of papers where each is narrowly focused on a topic and thus suitable for only one or a few conferences, but writes such papers on several different topics, thus ending up with many conferences.

In the second strategy, that author needs to have greater depth of knowledge in multiple Aols, whereas in the first strategy, having enough knowledge to tie different Aols can be enough. One other possibility is for the author follows a hybrid approach. To look deeper into this, we select 10 authors with the largest number of publications and look at the diversity distribution of their publications.

To determine what path to prominence the authors chose, we look at their document entropy distributions. That distribution will determine how diverse the publications are in general, and what is the reason behind the authors diversity. Figure 4.19 shows the document entropy distribution for the top 10 authors. Some of the authors, e.g., authors 2346 and 10019, have a larger portion of publications with high diversity than other. This indicates that they tend to publish more diverse publications that can be applicable to a wide variate of conferences (Strategy I). On the other hand, authors with less diverse publications, e.g., authors 700 and 4930, appear to be following
Figure 4.19: Entropy distributions for the top 10 authors in the DBLP dataset
Strategy II, targeting multiple conferences with papers suited to those conferences. Figure 4.21 shows the topic profile for each of the top 10 authors. Among other things, this indicates that most of the top 10 authors achieve their prominence by focusing on Topic 5 (Database) and, to a lesser degree, on Topic 4 (Machine Learning) and 3 (Information Retrieval).

To get a closer look at each author’s publications, we plot all their publications in a reduced dimensional space using MDS. This allows us to visualize where the publications are placed in the epistemic space, and get another view of the authors diversity. Figures 4.22-4.31 show the publications of all the top 10 authors, with each pure topic also indicated to determine how the
Figure 4.21: Top 10 authors’ topic profiles
publications fall with respect to these topics.

In Figure 4.24 we see that although most of the author's publications are around the DM area, we see some documents that reach far towards the areas of IR and Web Mining. On the other hand, for author 4930 in Figure 4.30, the reach to areas far from the Database area is not significant. Author 9300 is purely focused on a section of the space that deals with Databases.
4.3. EPISTEMIC SPACE ANALYSIS

Figure 4.24: MDS epistemic space views of publications for Author 2346

Figure 4.25: MDS epistemic space views of publications for Author 700
Figure 4.26: MDS epistemic space views of publications for Author 3137

Figure 4.27: MDS epistemic space views of publications for Author 22593
4.3. EPISTEMIC SPACE ANALYSIS

Figure 4.28: MDS epistemic space views of publications for Author 10019

Figure 4.29: MDS epistemic space views of publications for Author 9300
CHAPTER 4. EPISTEMIC SPACE ANALYSIS

Figure 4.30: MDS epistemic space views of publications for Author 4930

Figure 4.31: MDS epistemic space views of publications for Author 1395
4.3. EPISTEMIC SPACE ANALYSIS

and Information Retrieval in relation to the Web. From Figure 4.21 we see that author 9300 peaks in the topics that discuss Databases and Information Retrieval.

Authors and Conferences

One of the advantage of having a unified space where authors, documents, and venues can be embedded is the ability to ask questions about different types of entities. In this part we analyze authors and conferences and ask questions that can be answered by looking at both of these together in the epistemic space. The first question consider is how far the authors are from the conferences they publish in. Then we look at their distance from the areas of interest we defined in Section 4.3.1. To do this, we plot the distance distribution of authors to conferences. The results in Figure 4.32 show that conferences tend to attract authors that are placed relatively close to their location in the space. That seems to be the case for most of the conferences except AAAI, IJCAI and CIKM. The first two conferences are in the AI area and the last conference is in the KM area. Figure 4.33 shows that the two areas, AI and KM, are the areas with the most distant authors. This is also reflected in the document analysis of conferences.

We also look at how conferences and their areas of interest attract diverse authors. From Figure 4.35 we see that in the ares of ML and AI, most of the authors have high diversity. WWW and IR however, have more authors that have lower diversity values.
CHAPTER 4. EPISTEMIC SPACE ANALYSIS

Authors and Documents

In Section 4.3.2 we looked at documents that are significantly distant from their conferences mainstream. In this section we look at prominent authors who tend to publish distant publications and analyze their ATPs. For this example we select prominent authors who authored the furthest 5% of the publications in their conferences. Figure 4.36 shows that these authors tend to have publications with low entropy, therefore they are not very diverse. The authors also do not have a high number of publications, therefore they are prominent in a low number of conference. In fact, for this example, most of the authors are prominent only in one conference and a few in two. These results suggest that the authors considered here work in an area that is not really suitable for the conference they publish in, but they publish multiple papers in that conference – perhaps for extraneous reasons.

When looking at the authors’ topic values shown in Figure 4.37 we see that these authors have a topic in which they peak, but also other topics with relatively high values. This leads us to look into how their publications
4.3. EPISTEMIC SPACE ANALYSIS

Figure 4.33: Author-Conference AoI distance distribution

are distributed in the space. We applied MDS to their publications and the Figures 4.38 - 4.43 show the results.

From these authors, the publications for author 11198 and author 8725 are located closer to the Image Analysis AoI. The number of publications available to us in the area of Image Analysis is limited, so it is difficult to make any strong conclusions about these authors. For the other authors in this example, all of them are close to the AI and Database region. As we
see the number of publications being low, we believe that those authors have not yet established themselves as prominent authors in many areas, but they tend to publish more than the average author which makes us believe they are progressing towards a more prominent status in the future.

4.4 Conclusion

In this chapter, we have described a unified epistemic space, where multiple types of entities can be embedded. We defined the space dimensions using the topics extracted in Chapter 3, and explained how different types of entities can be embedded. We then analyzed the different types of entities in the epistemic space, and asked questions related to their location. For the purposes of this thesis, we used a scientific publication dataset known as DBLP 1.2 that contains information about authors, documents and venues. A part of the analysis focused on venues and their location in the space. We were able to group venues together based on their locations in the space and derive ar-
4.4. CONCLUSION

Figure 4.35: Author entropy in areas of interest

eas of interest. We were also able to analyze the diversity of venues and determine which venues tend to attract more diverse authors and publications. Another part of this chapter focused on authors and their participation in venues. We defined prominent status and answered questions about authors that have this status. We also looked at the different ways an author can become prominent, and tied that to their epistemic diversity. Finally, we looked at prominent authors that tend to publish documents that are relatively far from the mainstream of the conference in which they were published. With these authors having a low number of publications, we believe that most of
them will reach a prominent status in different conferences in the future. One way to verify this is by looking at the temporal aspect of their publications, which is not in the scope of this research.

This is only one example of how having a unified epistemic space for different types of entities allows us to ask and answer different types of questions. The scope of this research was focused on scientific publications, but this type of a space can be extended to other applications such as social

Figure 4.36: Publication entropies for prominent authors with atypical publications
Figure 4.37: Topic profiles for prominent authors with atypical publications
CHAPTER 4. EPISTEMIC SPACE ANALYSIS

Figure 4.38: Publications in the Space for Author 11198

Figure 4.39: Publications in the Space for Author 20456
4.4. CONCLUSION

Figure 4.40: Publications in the Space for Author 8725

Figure 4.41: Publications in the Space for Author 3114
CHAPTER 4. EPISTEMIC SPACE ANALYSIS

Figure 4.42: Publications in the Space for Author 7365

Figure 4.43: Publications in the Space for Author 3283
media analysis and search engine optimization.
Chapter 5

Conclusion and Future Work

The research described in this dissertation was motivated by the desire to study texts, their authors and the communities where they are generated and communicated within a single epistemic framework. This is essential in order to understand how ideas emerge and evolve within communities of thinking individuals, and how they shape these communities. The focus in this dissertation was on three specific tasks:

1. Identifying latent epistemic variables, or topics, from text corpora.

2. Using the topics to define an epistemic space in which epistemic entities such as documents, authors and collections of documents can be embedded.

3. Discovering interesting patterns by the separate and joint analysis of different types of epistemic entities.

The first part of this dissertation described a network-based approach called *Topic Extraction through Partitioning of Lexical Associative Networks* (TExPLAN) for extracting topics from text corpora. The motivation for this
method was to develop a more cognitive approach than the statistical one used by probabilistic topic models. TExPLAN starts by constructing an associative network of words from the given text corpus, where word associations represent the frequency of co-occurrence between pairs of words. Once the network is constructed, seed topics are extracted by identifying communities in the word network using a community extraction algorithm. The method is based on optimizing the modularity of node partitions, and generates a hierarchical structure of communities. This method was chosen because it is fast, hierarchical, and it does not require a priori knowledge about that number of communities in the network. Once the set of seed topics is defined, TExPLAN uses the connections between seed topics to assign a membership value to every word in each topic, thus obtaining a final set of topics, each defined as a distribution over the entire vocabulary.

The TExPLAN algorithm was applied to four datasets, three of which are scientific publication datasets and the fourth is the University of South Florida Word Association Norms dataset. For the three scientific publication datasets, TExPLAN was compared with the most widely used PTM known as Latent Dirichlet Allocation (LDA). The results led to several important conclusions:

1. TExPLAN was able to discover coherent, meaningful topics in all cases. Across all four datasets, there was no instance where it produced a topic that could not be interpreted clearly. In contrast, LDA occasionally produced topics that were hard to interpret or were mixtures of two topics. This depended on the number of topics LDA was asked to produce.
2. TExPLAN did not require external specification of the number of topics, but found the appropriate number based on the inherent structure of the underlying lexical association networks. This is the likely reason why the topics were always meaningful. LDA required explicit specification of the number of topics but provided no guidance on how to choose this number. A heuristic suggested by Arun et al [12] was used to test LDA with several values, which showed that the results were extremely sensitive to this choice. Setting the number of topics too low produced mixtures while setting it too high yielded split or meaningless topics.

3. In at least one instance (the DBLP dataset), LDA required more topics to discover a particular significant topic that TExPLAN discovered naturally with a smaller total number. When LDA did discover the topic, the total number of topics was high enough to also generate other meaningless topics.

4. The topics generated by TExPLAN were very robust, so that running the algorithm again on the same dataset produced virtually the same topics. This contrasted sharply with LDA, which showed a great deal of variation in its output even for the same dataset.

5. At least in the LDA implementation used for comparison in this dissertation, the dataset needed to be stemmed (i.e., versions of the same word with different endings unified) in order to produce results even somewhat comparable with TExPLAN. Without such stemming, LDA’s results were even worse. Paradoxically, TExPLAN performed more poorly after stemming, probably because stemming causes significant loss of network structure information. In general, the order of performance was:
1) TExPLAN without stemming; 2) LDA with stemming; 3) LDA without stemming; 4) TExPLAN with stemming.

6. Since TExPLAN only requires a lexical associative network and not actual documents, it can be used with datasets such as the University of South Florida Word Association Norms. LDA and other algorithms of its type require documents, and cannot be used on such datasets. Also, though this was not tested in the work reported here, LDA does notoriously poorly with corpora of short documents, but this should not pose a problem for TExPLAN.

7. TExPLAN ran faster than the non-compiled version of LDA on the same dataset and the same machine. A significant part of this was the time taken by the lemmatizing process LDA used that required access to the WordNet ontology [136]. However, when LDA was run without using lemmatizing, its comparative performance on extracting topics deteriorated in some instances.

In summary, the experiments in this dissertation indicated that TExPLAN is generally better, more robust, more broadly applicable and much faster than LDA. Of course, it is possible that this may not hold for some datasets. Future work can focus on making TExPLAN even more scalable and dynamically expandable as new documents added to the collection.

The second part of the dissertation described the construction of an epistemic space using the topics discovered by TExPLAN as its dimensions. The papers, authors and conferences in the DBLP dataset were embedded in this space and analyzed in several ways. For this dataset, it was shown that conferences can be grouped into meaningful areas of interest which were
not given *a priori*. A measure of epistemic diversity was defined for entities based on the entropy of their normalized coordinates. Using this diversity metric, the diversity of conferences and determine which ones tended to attract more diverse authors and publications. Diversity was also calculated for authors and used to analyze how diversity is related to the number of publications. Another part of the analysis focused on authors and their participation in conferences. Prominent status for authors was defined by their degree of participation in conferences, and several issues were addressed about authors that have this status, including the different ways an author could become prominent based on their epistemic diversity.

The examples listed above are a small subset of queries that this unified space allows us to answer. The applications of this type of unified epistemic space can range from specific tasks like understanding author behavior in a single conference, to more general tasks like clustering web pages on the World Wide Web. The direct contribution of this part is new approach to analyzing multiple interrelated epistemic entities which can be embedded in a unified geometrical space.

The next section lists some future work that can utilize and build upon the contributions made in this dissertation.

## 5.1 Future Work

The future work of this research can take two main approaches; one that focuses on improving the topic extraction method, and the other focuses on applying the notion of a unified space to different domains. Some ideas of two approaches are:
• **Dynamic Topics:** Most approaches for extracting topics from text corpora, including LDA and TExPLAN, need to work with the whole dataset. This is problematic when dealing with dynamic datasets because the number of documents is continuously growing. Future work can focus on developing *incremental methods* that can handle new documents as they come in without having to reapply the algorithm on the whole dataset. The network-based nature of TExPLAN already makes it more suitable for such an incremental version.

• **Temporal Analysis:** The process of topic extraction and epistemic space configuration can be applied to longitudinal text corpora, i.e., corpora collected over an extended period where documents are tagged by date. This has many interesting applications. For example, just looking at the topics extracted at different times can show how topics in a research field have evolved over time. Or, if an epistemic space is defined over the entire period, one can look at how documents coalesce and move apart in this space over time, indicating where most of the epistemic activity is focused at various times. Similar tracking can be done for ideas extracted from the documents, which can elucidate the evolution of important ideas in the field and pinpoint the authors who contributed to them.

• **Recommendation Systems:** A straightforward application of the unified epistemic space discussed in this dissertation is to recommendation systems. The proposed space provides a geometrical representation of entities where the epistemic similarity between entities can be measured. Using other definitions of dimensions, one can obtain
spaces where other types of similarity can be measured. For instance, if a unified space can be defined where readers are embedded based on reading patterns as well as hobbies and interests, and books based on their content, then book recommendations can be improved considerably over those based purely on reading patterns.

- **Search Engine Optimization**: Search engines like Google try to show users the most relevant web pages in response to their queries, but different users could have very different notions of relevance even with the same query. A unified space where users and web pages can be embedded can potentially be used by a search engine to customize content for each user.

To summarize, the research reported in this dissertation represents an important first step in the larger project of using text corpora to understand how ideas emerge and evolve in communities, and change these communities in the process. The epistemic space framework described here is, in a real sense, the key “enabling technology” for such research, and promises to open up many interesting directions of investigation for future researchers.
Bibliography


[cited at p. 53]

[cited at p. 33]


[cited at p. 38, 70]


[cited at p. 6]


[106] G. Kumaran and J. Allan. Text classification and named entities for new event detection. *Proceedings of the 27th annual international confer-
ence on Research and development in information retrieval - SIGIR ’04, page 297, 2004. [cited at p. 55]


Appendix A

A Multi-Agent Model of Evolving Communities and Ideas

In this part of the research, we explore the emergent of ideas and community structures through a computational model of the co-evolution of social networks and ideas in a large group of individual cognitive agents. We explore how ideas emerge in such a setting, and how these emergent ideas affect the structure of the social network. We then try to explain the relationship between social dynamics and innovation in large groups.

Capturing and identifying embedded and hidden ideas in social networks is a fundamental issue for creativity, innovation, and security. Building artificial systems that are capable of extracting such ideas is especially attractive because of the exponentially increasing participation by millions of individuals all over the world. Not only does this make these social networks a rich source of ideas, it also increases their value as a medium of information, influence and organization for both good and ill. This is the potential exploited by search engines like Google, and by marketers, opinion-makers and pro-
pagandists everywhere. This part is motivated by two fundamental insights:

- **Latent Ideas**: In addition to explicitly stated or apprehended ideas, every mind and every community also has a vast store of latent ideas, i.e., ideas that are not obvious to anyone, but can emerge easily in response to environmental stimuli.

- **Emergent Communities**: Ideas play a fundamental role in forming and shaping emergent communities within large social networks, with individuals organizing into coherent groups with magnified influence.

Both these insights are especially relevant in this age of cyberspace social media and instant global communication. Our ultimate goal is to understand the formation of latent ideas, to develop ways to extract them, and to model the dynamics of the emergent communicates they help shape.

### A.1 Goals and Methods

We begin with the postulate that the ideas latent in a population will eventually emerge based on patterns of interaction among agents. Sometimes, they emerge easily to form conventional wisdom, public opinion or prejudices. In other cases, they emerge with more difficulty, and are often seen as radical or innovative. However, before these ideas emerge, they are presaged in the discourse of the population. This means, that in principle, by mining the discourse of the population in the appropriate way, and taking the social network into account, one can "predict" ideas that have not yet emerged. To do this, we must be able to do the following:
1. Identify salient idea components (concepts / phrases / terms) in the discourse (e.g., by mining texts or conversations).

2. Have a mechanism to determine the affinities between these in specific contexts.

3. Have a mechanism to generate a combination of these components productively and predictively.

Almost by definition, any model trying to predict what people will think or say will be extremely hard to validate. Something can be "in the air" and yet go unsaid for years. Thus, the test of such a model should be whether any ideas are predicted in a reasonably efficient, non-exhaustive way, i.e., whether the model, in some sense, does better than chance. Even this will be extremely difficult to prove. However, the payoff for even limited success is great. Consider, for example, if one out of a thousand ideas mined from intelligence intercepts turned out to reveal a planned terrorist plot. Furthermore, if the system shows promise, one can consider how the emergence of ideas can be promoted, retarded, diverted or shaped by interventions, i.e., propaganda or advertising. Ideas can be "seeded" into some systems, and not into others. It would be interesting to discover which ones are more seedable, and why. Another important assumption in our work is that the dynamics of both the formation and emergence of ideas in a population is affected strongly by the underlying social network and the ideas, in turn, affect the dynamics of the social network. As agents communicate over the social network, their individual knowledge undergoes change, leading to an "aha!" moment in some agents, and new ideas are born. These ideas then become part of the discourse, causing the strengthening and weakening of social ties, and
the reorganization of the social network. Clearly, the structure of the social network and the preferences and propensities of the agents will play a crucial role here. In this paper we develop a simple multi-agent approach to study these issues. The model has several important features:

1. Ideas as conceptual combinations: Ideas are defined as combinations of concepts or words[38, 193], and represented by cliques in an agent-specific semantic network whose nodes are the words. The ideas in an agent's semantic network can be active or latent, depending on whether the agent is aware of them.

2. Heterogeneous dynamic agents: Agents are defined cognitively by ideas in their semantic networks, so every agent is different. New ideas form as concepts and links between concepts are added to the agents' semantic networks.

3. Self-organized social network: Social networks form in the agent population based on similarity of ideas among agents who meet each other and exchange ideas. Once formed, the social bonds are maintained by subsequent follow-ups, and directly influence the knowledge of agents.

4. Idea extraction by clique discovery: The process by which agents become aware of new ideas in their own minds is modeled using a simple backtracking algorithm [34] to discover cliques in their semantic nets.

5. Community detection: Communities are detected using a backtracking algorithm for mining cliques in the social network.
A.2. SEMANTIC NETWORKS AND IDEAS

6. Ideas in the community: Each community is mined for ideas latent in the community using clique detection in the joint semantic network of the community.

These features allow us to explore an idea exchange system where social networks, idea generation in agents, and community formation co-evolve. The last item is especially significant because it allows us to associate each community with a general set of ideas that are of interest for its members. We consider the effect of social preference for each agent in creating social links and then evaluate the resulting communities with their innovativeness. This admittedly very model obviously ignores several major issues. However, it represents a first step in terms of integrating social networks, idea generation, idea exchange, and community formation in a virtual population.

A.2 Semantic Networks and Ideas

An idea, $I$, comprises of a set of words $I = w_0, w_1, ..., w_n$ connected by links to form a clique. A semantic network is a graph, $G = (V, E)$, of words formed by the superposition of many ideas. Here, $V$ denotes the set of nodes (words), and $E$ the set of connections between them. If nodes $w_i$ and $w_j$ have co-occurred in any known idea, they have a link between their nodes. Ideas can be added to a semantic network by adding any nodes that are not already in it, and/or by adding links between nodes in the idea. This results in the formation of latent ideas as emergent cliques in the semantic network A.1.
Figure A.1: Three embedded ideas, ABC, BDE, and CEF generate a latent idea BCE (shown in solid blue)

A.2.1 Agent Model

Each agent in the system has a semantic network of words, linked together as described above. Each agent also has a number of social links to other agents its immediate contacts which embeds it within the populations social network.

A.2.2 Communication and Idea Formation

For this model, each agents semantic networks seeded with an initial idea $I_i^0$. At every step in the simulation, each agent interacts with one of its immediate contacts, and possibly shares an idea from its semantic network. An agent receiving an idea from another agent incorporates it within its own semantic network with a probability proportional to the strength of its social linkage with the other agent.

Each agent also discovers latent ideas within its own mind by searching for emergent cliques in its semantic network. We use a backtracking algorithm[34], to find cliques of sizes 3, 4, and 5. If a clique $Q_i$ is a subset
of clique $Q_j$, we ignore $Q_i$ and keep $Q_j$. A new idea is defined to be an idea that is newly formed in the agents semantic network but already existed somewhere in the population; a novel idea is a new idea that did not exist previously anywhere in the population.

A.2.3 Social Network Formation Mode

Initially, the agents are connected in a small-world social network, as shown in A.2. A small-world network is a one where nodes are connected mainly within a cluster or neighborhood, with a small number of more distant connections in the network[222]. The existence of such short-cuts reduces the shortest path between the nodes without decreasing clustering, resulting in the small-world effect. At each step of the simulation, agent $i$ picks an agent $j$ from its social network to communicate with. This choice depends on two normalized factors: 1) The semantic similarity, $S_{i,j}(t)$, between agent $i$ and agent $j$, and 2) the social connection from $i$ to $j$, denoted by $C_{i,j}(t)$. Agent $i$ picks agent $j$ with a probability

$$P(i, j) \approx \mu_s S^j_i(t) + \mu_c C^j_i(t)$$  \hspace{1cm} (A.1)

where $\mu_s$ is the weight of the semantic similarity, $\mu_c$ is the weight of the social connection strength, and $\mu_s + \mu_c = 1$. Agent $i$ also tries to communicate with a random agent $k$ connected to agent $j$, hoping to expand its social network. A social connection between agent $i$ and agent $k$ will be formed only if $S_{ik}(t) \geq \theta$, where $\theta$ is a similarity threshold. Thus, new social connections are established based only on semantic similarity (i.e., a meeting of the minds), but the agents choice to interact with other agents in its network and thus to maintain its links with them may depend on both social connectivity
APPENDIX A. A MULTI-AGENT MODEL OF EVOLVING COMMUNITIES AND IDEAS

Figure A.2: The initial social network, with agents organized in a small-world formation

and semantic similarity.

The semantic similarity, $S_{ij}(t)$, is calculated as follows:

- Every word $w$ in $i$’s semantic network $G_i$, i.e. $w \in V_i$, has a similarity value $D_{ij}(w)$ to the same word in $j$’s semantic network $G_j$, $w \in V_j$.
- If $w \in V_i$ and $w \in V_j$ then $D_{ij}(w) = 0$
- If $w \in V_i$ and $w \in V_j$ then $D_{ij}(w) = |K_i^{w} \cap K_j^{w}| / |K_i^{w}|$ where $K_i^{w}$ is the set of all words that are connected to $w$ in $G_i$ and $|X|$ is the cardinality of set $X$
- $S_{ij}(t) = \sum_w D_{ij}(w)/ |V_i|

After the social interaction between agent $i$ and agent $j$ takes place at time $t$, the strength of the social linkage is updated as follows:

$$L_{ij}(t) = L_{ij}(t - 1) + 1 \quad (A.2)$$
If the social interaction does not take place between socially connected agent i and agent j the strength of the social connection decays as:

$$ L_{ij}(t) = L_{ij}(t - 1) - 1 $$ (A.3)

Finally,

$$ C_{ij}(t) = \min(L_{ij}(t)/|N_i|, 1) $$ (A.4)

where $N_i$ is the set of all the agents in the social network of agent i. When the strength of the social connection falls below 0, the connection is lost. The initial value of $L_{ij}(t) = |N_i|$.

### A.3 Community Identification Model

As the last step of the simulation, we extract the different communities in the social network using a backtracking clique extraction algorithm \cite{34}. After determining the initial communities a merge step is performed to minimize the number of communities and come up with a set of communities where the members are very well connected. For a set of communities to merge they have to share at least a fraction $\theta_c$ of their members. An agent could appear in more than one community but it will be associated with the largest community in which it appears. Some agents may not meet the criterion for membership and are designated as singletons. These can be divided further as those who are truly isolated (e.g., have very few links to any community) and those who bridge several communities. The emergence of bridge singletons is a potentially very interesting social phenomenon, and will be addressed in a separate report in the future.
A.3.1 Idea Extraction

After merging the communities and identifying the final set of communities to be considered for idea extraction, we generate a community semantic network (CSN) for each community. The semantic network of the community is the superposition of the semantic networks of all its individual members. Once the CSN is created, we treat it as a network of an individual agent and use the clique extraction algorithm to extract the ideas in the community. We then look for the most frequent words in the ideas extracted by the algorithm and create a set of words that represent the theme of ideas in the community.

A.3.2 Results

The model was run to test different scenarios under which agents make their interaction choices. This part of the research focuses on one of the many possible parameter variations that are possible in this model. Three different experiments were conducted for different values of $\mu_s$ and $\mu_c$. All experiments had the same number of agents $n = 500$, $\theta_s = 0.4$, and $\theta_c = 0.4$. The results of the simulations are shown in the following section. The agents were initially seeded with ideas from posts in two different on-line news groups. The posts discussed topics in Middle East politics and religion. The maximum size of a post was 300 words, and some of the posts were replies to previously discussed ideas. The ideas were the topics captured from the subject lines, after removing stop words, and used to seed agents with initial ideas. The total number of distinct ideas was 460 made of 890 different words. The data set was obtained from the Machine Learning Repository[1].
A.3. COMMUNITY IDENTIFICATION MODEL

Figure A.3: The social network produced by Experiment 1. Different communities have different colors. Agents shown in black are not part of any community.

A.3.3 Experiment 1: Social Dominant

For this experiment we chose $\mu_s = 0$ and $c = 1$ to show the extreme case in which the choice to interact is based on social preference only, i.e., agents prefer to communicate with the agents they already know best. Figure A.3 shows the final social network with communities labeled in different colors. Clearly, the agents have organized into several distinct, strongly connected communities, with very weak linkage between groups. In fact, several communities have become detached from the larger population. Also, remnants of the initial pattern of connectivity among agents can still be seen in the final communities (compare A.2 and A.3), indicating that the communities are socially quite stable. The agents in this experiment were able to generate 226 new ideas, and 8 novel ideas.
Appendix A. A Multi-Agent Model of Evolving Communities and Ideas

A.3.4 Experiment 2: Mixed

In this experiment we tested a mixed scenario where the social dynamics was based on both social and semantic preferences: $\mu_s = 0.5$ and $\mu_c = 0.5$. The resulting social network in this experiment is shown in figure A.4. Here again, we see the emergence of distinct communities, but the communities are much more connected to each other than in Experiment 1, giving the whole system a more web-like structure. The agents in this experiment were able to generate 480 new ideas but only 4 novel ideas.

A.3.5 Experiment 3: Semantic Dominant

In this experiment we considered the other extreme scenario with the social dynamics based only on semantic preference; $\mu_s = 1$ and $\mu_c = 0$. Thus, agents choose to associate mainly with like-minded agents. The resulting network is shown in figure A.5. In this case, the communities that form are much more inter penetrated, presumably because agents keep evolving se-
Figure A.5: The social network produced by Experiment 3. Different communities have different colors. Agents shown in black are not part of any community.

mentally and quickly establishing new relationships outside their current communities based on similarity of ideas. In contrast to the communities based on historical connections (Experiment 1), these idea-based communities are much more dynamic, with a continuous turnover of members. This is apparent from the fact that the original social structure in figure A.2 has been almost completely destroyed in the final distribution shown in figure A.5. In this experiment the number of new ideas generated by the agents is 318 and the number of novel ideas is just 2.

These results suggest that injecting intellectual diversity (and, thus, possibly intellectual conflict) into communities held together by social rather than intellectual bonds is more likely to generate innovation than making intellectual harmony the basis of social interaction. Of course, the situation is much more complicated in actual human populations, but the insight is potentially quite interesting.


A.3.6 Degree Distribution

Figure A.6 shows the degree distributions for the networks produced by the three experiments. In spite of the significant differences in network structure, the degree distributions are quite similar, and closely resemble those found in real social networks [156].

![Degree Distribution Diagram]

Figure A.6: Degree distributions for the networks generated by the three experiments

A.3.7 Community Size Distribution

Figure A.7 shows the final distributions of community sizes obtained in the three experiments. The graphs show only communities of size 2 or larger, with the number of singleton agents for each case shown in the inset. It is immediately clear that the experiments produce very different distributions, confirming the qualitative impressions from the network plots. In Experiment 1, the dynamics results in the formation of a large number of very small communities and a few communities of various larger sizes. At the other extreme, Experiment 3 leads to the formation of small communities across a broader spectrum of sizes, and a small number of much larger communities. The dis-
A.3. COMMUNITY IDENTIFICATION MODEL

tribution produced by Experiment 2 falls between these two extremes. The number of singleton agents in Experiments 2 and 3 is lower than in Experiment 1, but the explanation of the pattern seen requires deeper analysis, and is subject for future research.

![Figure A.7: Community size distribution generated by the experiments](image)

A.3.8 Random Interactions

The results of the three experiments described above indicate that the ideational productivity of social systems depends quite strongly on the basis of social interactions, and that more diverse communities produce more novel ideas. To check this further, we considered a case where agents interacted with each other purely randomly rather than on the basis of prior social linkage or similarity of ideas. This, of course, exposes every agent to a wide diversity of ideas, which would presumably lead to many novel ideas. In fact, starting from the same initial network as in the other three experiments, the agents in the random case were quite productive, generating 5 novel ideas, which is similar to what is seen in Experiment 2. However, as shown in A.8, the network ended up losing almost all community structure.
A.4 Discussion

By looking carefully at the results we can discern some interesting patterns. First, it is clear from figure A.7 that the pattern of community sizes depends significantly on the social communication preferences of the agents. Communicating based only on historical social connectivity produces a large number of small communities and isolated agents. All communities produced in this case also tend to be much more stable. In contrast, communicating based on commonality of ideas leads to communities of many sizes some of them quite large and high mobility. A second interesting observation is that the number of novel ideas decreases when the weight of semantic similarity in communication increases A.9. We believe this is due to the fact that while communicating mainly with like-minded agents increases social mobility, it diminishes the possibility of synthesizing anything truly novel because the communities the agents move in largely share a common semantic substrate. On the other hand, communities that are based on historical social connections, while less
dynamic (i.e., more stable), can be quite diverse in terms of ideas, since the initial (i.e., historical) connectivity is not based on similarity of ideas. Nor is similarity of ideas as important in maintaining social links in these communities, so the communities remain intellectually diverse even as they become structurally more stable. These "old and diverse" communities, therefore, become incubators for truly novel ideas. We have not verified this explicitly, but we speculate that this benefit holds only so long as the communities retain some degree of connectivity with other communities. Once they become isolated, they will eventually cease to generate novel ideas as well (unless new concepts are allowed to enter the system). The opposite situation occurs when agents communicate preferentially with others who share their ideas. This results in structurally dynamic communities with high turnover, but agents move mainly within a familiar intellectual milieu and do not generate much in the way of novel ideas. To check these hypotheses further, we looked at the internal semantic homogeneity of the communities in which novel ideas were generated at the time of the generation of these ideas. Figure A.10 shows the results. As hypothesized, the communities generating novel ideas were significantly less homogeneous than the rest. Interestingly, however, this depended quite strongly on the basis of social interaction. When social interaction was based on similarity of ideas, only exceptionally diverse communities produced novel ideas. Basically, agents in this situation generated novel ideas only when they found themselves stranded in diverse communities and were forced to interact with agents with (somewhat) different ideas. Since this only happened rarely, novel ideas as a whole were rare. Most agents, even if they were in a fairly diverse community, were able to find like-minded agents to interact with and avoid exposure to the diversity in
their community. In contrast, when social interaction was based on prior social linkage, communities needed to be only slightly more diverse to generate novel ideas. This is because agents in these communities were not looking for similar agents to interact with, and were thus exposed to the full diversity of the community. These hypotheses are also supported by the results of the situation where agents encountered each other purely randomly. The increased exposure of agents to new concepts in this case does generate a fairly high number of novel ideas, but with an unnatural loss of almost all community structure.

![Figure A.9: Number of novel ideas with respect to the social weight for choosing target agents.](image)

We also note that recent work by Palla et al. [163, 161] has suggested that there can be two types of long-lived communities: Those that are small and stable (i.e., have low turnover), and those that are large and dynamic (i.e., have high turnover). These are exactly the two cases we see in Experiments 1 and 3.

Another significant feature we observed is that, in each experiment, these are a few combinations of words that become dominant for several small,
isolated communities that, if they could interact socially, might form a larger community. This also suggests that some ideas have an inherent attraction for members of a community with common interests, and that such ideas can persist as "hot topics" in small detached pockets within the population. The phenomena produced by the simulations we describe suggest that the approach is a promising one for the study of larger-scale social semantic networks. The very idea of semantic networks for populations and communities is potentially very useful, and will be pursued in greater detail in future work.

Figure A.10: Community homogeneity with respect to the social weight for choosing target agents.

A.5 Conclusion

The co-evolution of social networks and collective knowledge is a very rich area of research that has not been explored much. In this section of the research, we have presented a very simple agent-based model that is used to integrate social network dynamics, idea exchange, idea generation, and community formation in idealized agent populations.
Appendix B

Behavioral Classification of Agents on Facebook

In this part of the research, we use several anonymized data sets of Facebook interactions among large groups of agents over extended periods to identify classes of agent behavior. In particular, we consider the relative social attention agents devote to interacting with other agents, and identify four distinct types of behavior in this regard. These four classes are found to be robust across multiple data sets, and thus, presumably, reflect real differences. We then train a neural network classifier to recognize these behaviors and show that it can successfully classify agents according to their interaction patterns in a novel data set. We also provide provisional interpretations of the four behavior patterns identified in this study.
B.1 Background

In the area of data mining, a lot of the research has been done to understand the structure and evolution of online social communities[105][174][14][113]. Some of these studies have classified members of social networks into one of three groups[14][113] based on their structural role in the network: The Singletons are agents with zero friends; The giant component comprises users who are connected directly or indirectly to a large fraction of the entire network; and the The middle region covers small groups whose agents only interact with others within the group and with the network at large. In this study we aim to classify users into groups based on their interactional behavior rather than structural position. We believe that such classification will reveal valuable insights on the way users divide their social time between their friends on the network.

B.2 Data sets:

This study used two data sets - labeled A and B - representing agents on Facebook. Each data set provides a social graph indicating “friend” links between agents, and several interaction graphs, indicating contacts between linked agents over a period of time. The social graph for Dataset A comprised 3,097,166 users and 28,377,481 edges, and that for Dataset B 2,937,614 users and 24,236,701 edges. Interaction graphs were available for each data set for durations of 1 month, 6 months and 12 months, with the following number of interactions in each:

- A1 - Dataset A (1 month): 1,412,252 interactions
B.3 COMMUNITY EXTRACTION

- A6 - Dataset A (6 months): 7,483,904 interactions
- A12 - Dataset A (12 months): 16,889,111 interactions
- B1 - Dataset B (1 month): 1,974,590 interactions
- B6 - Dataset B (6 months): 8,442,451 interactions
- B12 - Dataset B (12 months): 13,650,113 interactions

The data was obtained from [227] and used with permission.

B.3 Community Extraction

Given the large number of agents in the data sets, we decided to focus on the subsets of agents that could be considered to have significant participation in a functionally useful sense. To do this, we extracted communities of agents from the social networks using the Clique Percolation Method (CPM) [163, 52]. In CPM a $k$-community is defined as the maximal chain of adjacent $k$-cliques. Two $k$-cliques are considered to be adjacent if they share $k-1$ nodes. We decided to use $k = 4$ as it appears to be the most reasonable value to use. A value of $k > 4$ produces very few communities, and a value of $k = 3$ produces too many small ones. Only agents belonging to at least one community were classified in the analysis below, though their interactions with all agents were taken into account. With this restriction, the number of agents and links analyzed were:

- Dataset A1: 1051 nodes and 3859 edges
- Dataset A6: 21690 nodes and 202611 edges
• Dataset A12: 64879 nodes and 959927 edges
• Dataset B1: 2229 nodes and 8128 edges
• Dataset B6: 30532 nodes and 202611 edges
• Dataset B12: 53633 nodes and 652776 edges

B.4 The Devotion Measure

The interaction pattern between two connected agents $i$ and $j$ was measured through a quantity termed relative devotion, $\Delta_{ij}$, defined as:

$$\Delta_{ij} = \left(\frac{I_{ij}}{I_i}\right) - \left(\frac{I_{ji}}{I_j}\right) \equiv D_{ij} - D_{ji} \quad (B.1)$$

where $I_{ij}$ is the number of interactions that $i$ has with $j$, $I_i$ is the total number of interactions for $i$, $I_{ji}$ is the number of interactions that $j$ has with $i$, $I_j$ is the total number of interactions for $j$ (note that $I_{ij} = I_{ji}$), $D_{ij}$ is the fraction of $i$'s interactions that are with $j$ and $D_{ji}$ the fraction of $j$'s interactions that are with $i$. Thus, $-1 < \Delta_{ij} < 1$, where a negative value of $\Delta_{ij}$ means that $i$ allocates a lower fraction of his/her social time to interact with $j$ than $j$ is allocating to interact with $i$, and vice-versa for a positive value. A value of 0 means that $i$ and $j$ allocate the same portion of their social time to each other.

Based on this definition, each agent $i$ has a devotion vector, $\Delta_i = [\Delta_{i1} \Delta_{i2} \ldots \Delta_{in}]$, where $n_i$ denotes the number of agents to which $i$ is directly connected. Since, agents often have high-dimensional devotion vectors with variable lengths across agents, it is convenient to look at the histogram of their relative devotion values. This is obtained by distributing the relative devotion
values for \( i \) into 5 bins: \([-1.0, -0.6]\), \([-0.6, -0.2]\), \([-0.2, 0.2]\), \([0.2, 0.6]\) and \([0.6, 1.0]\). The total is normalized to 1 and the resulting length 5 vector, \( q_i = [q_i^1, q_i^2, q_i^3, q_i^4, q_i^5] \) is called the feature vector for agent \( i \). This feature vector constitutes a quantitative representation of an agent, and is the basis of classification.

\section*{B.5 Agent Clustering}

The feature vectors obtained for individual agents in Dataset B6 were clustered using an agglomerative hierarchical clustering algorithm with the \textit{earth mover's distance} (EMD) as the distance measure. The weighted average distance (WPGMA) was used as a the distance measure between the clusters. We decided to use EMD as the distance measure because it is appropriate for capturing the distance between two distributions. It is defined to be the minimum cost required to transform one distribution into another \cite{167}. For the purpose of this work we use a fast EMD algorithm developed by Pele et al.\cite{167}.

The clustering identified four types of agents in Dataset B6. Some representative feature vectors for each class are shown in Figure B.1.

The dendrogram for the clustering process is shown in Figure B.2a. Figure B.2b shows the number of agents assigned to each class, indicating that the vast majority of agents fall into classes 2 and 4. Figures B.2c and B.2d show the distribution of the node clustering coefficient and node degree for each class.
(d) Distribution of node degree for each class.

Figure B.2: (a) Dendrogram resulting from the clustering process; (b) Number of agents placed in each cluster; (c) Distribution of network clustering coefficient for each class; (d) Distribution of node degree for each class.
B.6 Neural Network Classifier

Using the classes identified by clustering, a neural network classifier was trained to recognize the class of any agent based on its feature vector. The classifier had two hidden layers and was trained using the back-propagation algorithm [226]. The classifier was trained using a portion of the B6 data set with the class labels (as given by the clustering), and was validated and tested on two other subsets of the data. Figure B.3a shows the confusion matrices for the training, validation and testing case as well as over the entire B6 data set. It is clear the neural network was extremely successful in learning the classes.

One advantage in training a classifier is that it can be used to label agents in networks that are too large for clustering, such as Dataset B12. The neural classifier was applied to this data set and provided class labels for all 53,633 agents. However, since B12 had not been processed through clustering, the accuracy of these labels could not be verified. For this, we used three methods:

Method 1: In this method, we plotted the feature vectors for 20 representative agents from each class as given by the neural network classifier B.4. A comparison of these with the feature vectors given in Figure B.1 shows excellent agreement, indicating that the classes assigned in B12 by the classifier were qualitatively the same as those found by clustering in B6. This is strong evidence that these classes are, in fact, robust across different networks, and that the classifier is able to generalize.

Method 2: Here, we plotted the degree distributions B.5 and clustering coefficient distributions B.6 for agents in all four classes as identified in B6 by
clustering (Figures B.5a and B.6a) and by the neural classifier in B12 (Figures B.5b and B.6b). Comparing this, it is apparent the corresponding distributions are similar, providing further evidence that the clustering and the classifier are finding the same classes in the two networks.

Method 3: Finally, we also did a partial direct comparison by taking a subset of agents from B12, subjecting them to the clustering algorithm, and comparing the labels obtained with those given by the neural classifier. The confusion matrix for this comparison is shown in Figure B.7, indicating that the two methods agreed on the classification of almost 97% of the agents.

Taken together, these results indicate two things: 1) The interaction behavior of agents in multiple social networks studied falls into four distinct and consistent classes; and 2) The neural network classifier is able to assign classes accurately to agents across different networks based on their feature vectors.

### B.7 Interpretation of the Classes

A natural question that arises is whether the classes found by the above analysis are meaningful. While we are still investigating this issue in detail, some provisional suggestions can be made as follows.

(Class 1 - Invisibles: In this class, the number of users is generally small and comprises users with low node degree and high clustering coefficient. These users are usually involved in a small number of interactions, which suggests that they belong to a small lightly-connected group of friends. The users in this class tend to have positive relative devotion values, which means that they are either less active than their friends, or that their friends ignore
B.7. INTERPRETATION OF THE CLASSES

Figure B.3: Confusion matrices for the neural network classifier: (a) Data used to train the network; (b) Data used to check for generalization during training but not used for training directly; (c) Data not used during training at all; (d) All data in B6.

Figure B.4: Classes of members based on neural network classifier
Class 2 - Normals: The majority of the users fall into this class. They have relatively high degree (30-50) and a wide range of interactions. The majority of the users in this group tend to have low to moderate clustering coefficient values, indicating that they have a broad and loosely knit group of friends with significant connections between them. The users in this class have directed devotion values that are close to 0. This means that they tend to interact with friends who interact equally with them.
Class 3 - Celebs: This class has a small number of users who tend to have a large number of friends. The members of this class have very low clustering coefficients, indicating that they tend to interact with a large and disparate set of people. The members of this class generally have negative directed devotion values, which means that they allocate a smaller portion of their time to friends, who tend to allocate a larger portion of their time to them.

Class 4 - Casuals: Members in this class are significant in number, with most of them having degree in the range of (40-70). Most of the members in this class have low clustering coefficient values, which means that they are connected to disparate groups of members. Members in this class have a wide range of interactions and directed devotion skewed slightly to the negative end of the spectrum. Members who belong to this class tend to allocate less time to their friends than what their friends allocate to them.