A Framework for Unsupervised Extraction of Social Networks from Textual Materials

Thomas Way
Department of Computing Sciences, Villanova University, Villanova, PA, USA

Abstract - *Digital communication is increasingly widespread and contains a wealth of information including social context. Understanding the social context within data can provide meaningful insights that contribute valuable knowledge for many purposes. This paper presents the design of a domain-independent framework for extracting social networks from textual material using an unsupervised machine learning approach. The design is discussed with motivation given for various design choices, social graph terminology is introduced, and the evaluation of an initial implementation of a social network extraction and graphing application is presented which points to future directions for this research.*

Keywords: Social network extraction, social graphs, data mining, co-reference resolution.

1 Introduction

As communication has become more facile and our world has become more technologically interconnected, so too have our networks of social relationships become increasingly interconnected and complex. Indeed, social networks have been an active area for sociologists, anthropologists and others who study human interactions. The need to understand complex relationships extends beyond these human interaction to data interactions, and in a world flooded by massive amounts of digital information, the challenges are enormous. The ability to detect and measure relationships in any sort of data can enable better use of that data to identify and solve problems, improve security, reduce misunderstanding, and enhance both scientific discovery and day-to-day life in many ways.

Extraction and analysis of social networks has been explored extensively in a variety of contexts, including expected areas like online social media networks such as Facebook, Twitter, and Pinterest [7, 10] and literature [5] including film and television [1] (see Figure 1), and also less obvious applications such as the detection of money laundering [4] and potential terrorism planning [11]. A vast array of algorithms [2, 13] and system-based [9] and web-based [8] software tools have been developed for extracting and analyzing social networks in a variety of ways covering a diverse range of specialized needs. While the considerable advances in social network extraction and analysis covering many years of research efforts provide excellent techniques and tools, most are limited to specific domains or require a significant amount of technical expertise to manage their use and interpret their results.

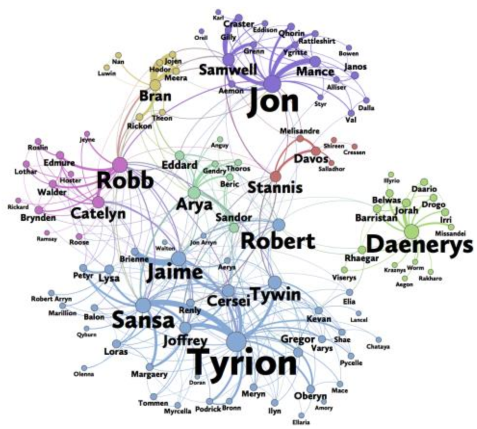

Figure 1. Social network for Game of Thrones. [1]

In this paper we present a framework describing a general purpose design and initial implementation of an approach for general-purpose social network extraction from textual material from just about any source or domain. This work attempts first to synthesize the more effective and efficient approaches and concepts from the large body of literature in this area into a framework and second to build upon this previous work to improve effectiveness and efficiency, and do so in a way that is accessible to the non-technical user. Ideally, the results of this work will include producing software tools that researchers in many disciplines can easily use to supplement their activities.

The framework is described as a sequence of phases, each accomplishing the next step in social network extraction and graphing of textual material that serves as its input, inspired by the design style of a programming language compiler which isolates implementation details from the design of each phase. Taken as a machine translation task, as when a compiler transforms a human-readable language into a machine-readable language, our framework has the goal of transforming a human-readable language into a visually-readable one in the form of a social graph that can be viewed, analyzed, and explored.

The paper first presents background and necessary terminology on social network extraction and social

graphing. Next, it presents the design and motivation of a domain-independent framework for extracting social networks from textual material. It then explains the incorporation of unsupervised machine learning that is essential for making the approach portable to new domains and making an implementation more readily used by those without a technical background, followed by an evaluation of the design and initial implementation of a social network extraction and graphing application is next presented. Finally, potential future directions for this research that would enhance its efficacy and efficiency are provided.

2 Background

A social network is a graph where each vertex or node represents an entity (person, location, organization, etc.) with additional information associated with edges or connections between vertices that define the nature of the relationship between each pair of connected entities [9]. Once a social network is in hand, it can be displayed graphically, compared with other social networks, and analyzed to reveal deeper information about the relationships it represents.

2.1 Social Network Extraction

Social network extraction refers to the manual or automatic identification and graphing of the social relationships contained in some textual input [6]. Social networks can be extracted from social networking web sites such as the "friend" lists from Facebook [3] and similar web sites, from textual material from a novels, screenplays [1, 5, 7, 12] or other character-based sources [2], as well as from financial documents and finance-related data [4], and from intercepted communications between known terrorists that may be indicative of planned terroristic activities [11].

There are perhaps as many specific approaches to social network extraction as there are domains that produce textual materials, which means that just about every area of human endeavor contains one or more social networks that may be discovered and better understood. The phases describe here are gleaned and synthesized from the literature and then extended, and sometimes renamed or refined, to suit the needs of our work.

2.1.1 Phases of Social Network Extraction

In order to extract a social network from textual material, a number of sequential phases or steps must be performed.

Data normalization is an optional, though frequently beneficial, initial step. The textual material to be analyzed and extracted from is prepare for processing by removing extraneous information that is unrelated to the material of interest. For example, in our initial work we analyzed public domain versions of well-known novels downloaded from the Project Gutenberg web site (gutenberg.org). Most of the content downloaded from Project Gutenberg contains additional front and back material, such as information about the author, the transcription process, permission to use, and more. Preprocessing such material involved removing this irrelevant material, leaving a clean text file containing only the contents of the novel to be analyzed. In some cases, preprocessing is either impractical or impossible, such as when automatically gathering material directly from the Internet or automatically processing significantly large collections of material. In such cases, any inaccuracies due to irrelevant material may be detected and ignored or otherwise accounted for in subsequent steps. Often, any inaccuracies is outweighed by the gains from processing large amounts of materials automatically.

Named entity identification, or **named entity recognition** (NER), is the process of parsing textual material and identifying entities that fall into a desired category, such as the names of persons, locations, organizations, dates, times, and quantities. The underlying mechanism of NER typically relies on computational linguistics and machine learning classification techniques, with a number of implementations and libraries available to researchers who need NER as part of their work. Some NER techniques restrict identification to names within spoken material recognized within quoted speech, which can be a strong indicator of relationship but may under account for relationships revealed in narrative content. [5, 9]

Co-reference resolution of named entities, also called **alias association**, **entity linking** or **entity disambiguation**, involves grouping together named entities that are textual different yet are synonymous with, or refer to, the same entity [9]. Among the common techniques are using unsupervised machine learning clustering techniques based on morphological similarity, string comparison distance measures, and subsequence similarity where two names being compared are either similar enough or resemble each other enough to be considered names for the same entity [8]. An obvious and rudimentary technique that only resolves identical named entities is fast, though it can lead to significant co-reference resolution inaccuracies in textual material where numerous variations for an entity are present. One technique that overcomes inaccuracies is a supervised classification technique that uses a custom list of named entities for a given textual input which is the result of an initial, manual co-reference step performed by a person [5]. While performing this step manually is very accurate, it is time-consuming and impractical when extracting social networks from large quantities of input sources.

Entity occurrence and position calculation is performed during the named entity recognition phase and continually updated during the co-reference resolution phase. Each occurrence of a recognized entity is tagged with its **position**, commonly its sentence number or other

specific location within the textual material. This tagging is coalesced as entities are resolved, with the result being a list of **entity groups**, each containing resolved co-references for the same entity and all positions within the material. The count of positions associated with each entity group is its occurrence count, a weight that expresses the relative importance of the entity it represents, with a higher occurrence count representing a higher importance. It is not unusual to exclude entity groups with a low occurrence count indicating relatively low importance in the social network, particularly when there are large numbers of entity groups which lead to very large social graphs.

Adjacency identification, or **relation extraction**, indicates a social relationship between two entities. It is defined as the maximum distance between entities (or entity groups) that are deemed to have a true social relationship. For example, a maximum distance of three sentences means that any entities within three sentences of each other are considered to have a social relationship.

Relationship importance is a measure of how important, or unimportant, a relationship is as measured by the **adjacency count**, the number of adjacencies between the two entities. The more often two entities are adjacent, meaning there is an identified relationship, the more strongly associated they are to each other. Some approaches also assign relative strength or weakness to a relationship between entities depending on shorter or longer distances, respectively, between them. [9]

Social graph generation is the final phase of social network extraction. It consists of the display of a graphical network of nodes or vertices that represent entity groups and connecting lines or edges between vertices that represent relationships between connected pairs of vertices. Each entity groups is represented by a single vertex, with the diameter of the vertex determined by the occurrence count of its entity group. Thus, more important entities have larger nodes in the groups. The thickness of edges between vertices can be determined by the adjacency count of the entities represented by vertices being connected. A stronger relationship between two entities is represented by a thicker line between their two vertices in the graph. [2, 3, 5, 6, 9, 12]

3 Framework Design

The framework for social network extraction and exploration presented here relies on the previously described phases. An algorithm is given to provide a more formal specification of the framework and approach, with motivation and details about our specific approaches noted as needed to augment each of the phases. Next, a description of planned extensions to the framework that will provide more functionality and flexibility is given.

3.1 Algorithm

Performing social network extraction in a flexible and retargetable way requires a clearly defined sequence of steps. It is assumed that the input used is textual material that contains occurrences of the type of named entities being graphed. It is also assumed that the textual material has been normalized and preprocessed to the extent desired for the analysis being performed.

The algorithm makes use of a two key parameters:

- **Minimum occurrence count** to include, which eliminates less important named entity groups to reduce complexity of the graph. An initial value of 3 is used, which for our experiments using literary textual material enables a social graph to focus on more important character entities.
- **Maximum adjacency distance** is used to determine if two entities are adjacent. An initial value of 1 is used, meaning that entities must be mentioned within the same sentence to be considered adjacent.

1. Tokenize textual material into sentences.
2. Process each sentence in turn, creating a list of any **recognized named entities** annotated with their sentence number.
3. Sort entity list in descending order by length of its name.
4. Process each entity in turn, identifying and grouping together all remaining, coreferent entities in the list, using subsequence and morphological similarity with each group, looking for a best match. When an entity is added to a group, it is removed from the available entity list and its position and occurrence count are merged with those of the group.
5. Process any still ungrouped entities, using string comparison distance (i.e., edit distance) and gender identification to attempt resolution.
6. Sort the entity group list by cumulative occurrence count. Eliminate any entity groups with an occurrence count below the **minimum occurrence count** parameter.
7. Calculate an all-pairs **adjacency count** of all remaining entity groups.
8. Generate a **social graph** by creating:

   a. One named vertex for each entity group using its most frequently occurring entity name and cumulative occurrence count to determine its diameter.

   b. One edge for each non-zero adjacency count between vertex pairs, using the adjacency count to determine thickness of the edge.

9. Export the generated graph in a persistent format, such as a spreadsheet that is interpretable by graph viewing software, or as a complete web page or graphical image ready for future viewing.

3.2 Extensions

The algorithm describe provides a foundation for general purpose social network extraction. Several extensions are planned that will provide additional power to the resulting analysis.

3.2.1 Graphing window

The use of parameters for a graphing window width in the form of a start and finish position, could be used to indicate the starting and ending position (in this case, sentence number) of the data to be graphed. This provides a sliding window that can display change in social relationships over time as indicated by earlier and later positions in the textual material.

In our initial design and implementation, we use a graphing window that encompasses all of the input. By offering an interface that enables a user to determine the desired sliding window as part of a social graph viewer, evolution of relationships through the material can be explored.

3.2.2 Social graph animation

Extending the idea of a sliding window further, the animation of a social graph can be accomplished. By selecting a reasonably starting point and sliding window width, an interface could automatically slide the window from start to end of the material, displaying an animated sequence of social graphs that illustrate the evolution of the material. Animation could also fix a starting position and gradually extend the ending point, illustrating the growth of the social network throughout the material.

3.2.3 Sentiment analysis

By applying sentiment analysis to the text containing each named entity, a positive sentiment can indicate that entity is a positive entity or protagonist, while a negative sentiment may indicate a negative entity or antagonist. Sentiment analysis applied to the text containing an adjacency of two entities can indicate whether the relationship is friend or foe. By merging the sentiments for each of the entity occurrences and each of the adjacency relationships, and overall impression can be gleaned of the positive or negative nature of each entity and relationship.

3.2.4 Social graph comparison

Graphs of social networks can be compared to identify similarities or to uncover differences. Graphs of literary works could identify similarities in works by the same author or changes to an author's style over the course of a lifetime. Similarly, social graphs can compare genres, time periods, gender or age of authors, and various other desired comparisons. The same graph comparison techniques could be applied to other domains: newspaper articles retrieved about a given topic, a series of books about the same character, applications to homeland security such as automatically or analyzing a series of intelligence reports from a period of time to coax out changes in relationships that only are apparent over time. Other types of named entities rather than people, such as cities or countries, as part of security uses, or for literature or news story analysis where organization, location, date and time, or facility, among others.

4 Implementation

The described framework has been implemented using Python 3 and the Natural Language Toolkit (NLTK). Sentence tokenization is done using the NLTK Punkt tokenizer and named entity recognition is performed with NLTK POS tagging and NE chunking support. Co-reference resolution is done using a variety of custom comparison techniques as described above. Adjacency counts are through straightforward techniques. Social graph generation relies on the open source graphing library Plotly (plot.ly) which provides support for graphing with both JavaScript and Python.

5 Evaluation

An initial implementation was used during initial development to extract and visualize the social graph from a text file containing the classic work by Charles Dickens, "A Christmas Carol." This work was selected because it is of a manageable size (20000 bytes) and number of named entities (113 identified names), is very familiar, and contains many examples of difficult entity resolutions.

Figure 2 shows a subset of a social network graph generated by analyzing Dicken's A Christmas Carol. Note that Ebenezer Scrooge is in the largest node, indicating he is the main character in the work. The other characters relative importance to the story, as quantified by their occurrence counts is represented by the sizes of their nodes. Connections between nodes indicate a social relationship exists and the thickness of those connections illustrates how important the relationship is as quantified by adjacency counts for pairs of entities.

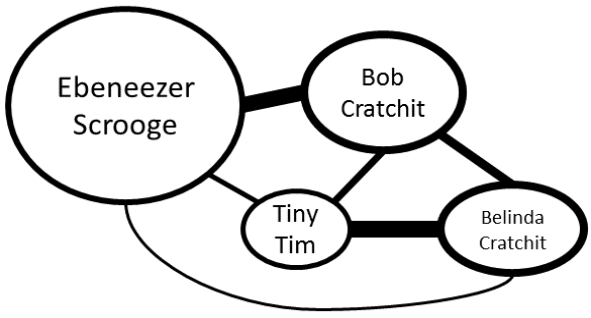

Figure 2. Subset of a social graph showing four entities.

Named entities contained in the A Christmas Carol for the character Scrooge are: Scrooge, Mr. Scrooge, Master Scrooge, Ebenezer, Mr. Ebenezer Scrooge, Uncle Scrooge, and Ebenezer Scrooge. These were all resolved using straightforward comparisons.

In the case of the character Bob Cratchit, similar names include: Bob Cratchit, Mr. Cratchit, Cratchit, Belinda Cratchit, Miss Belinda, Belinda, and Bob. Through significant trial and error, resolution that included gender identification produced the correct grouping, creating one group containing Belinda Cratchit, Miss Belinda and Belinda, and another group with Bob Cratchit, Mr. Cratchit, Cratchit and Bob.

The following three illustrations (Figures 3-5) show the evolution of a subset of the social relationships in A Christmas Carol. The graphs have been colored using a preliminary application of sentiment analysis as described above, though it was manually applied to the graphs after automatic extraction and graph generation.

Nodes and connections in red indicate a negative sentiment while green indications a positive one and black indicates neutral. The thickness of the node borders indicates the relative strength or weakness of the sentiment associated with the contained entity.

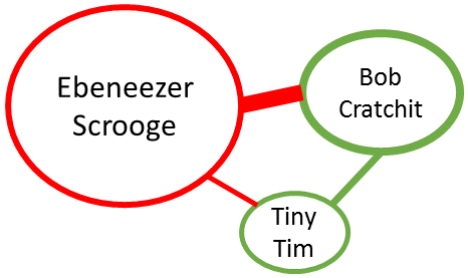

Figure 3. Subset of social graph early in the novel.

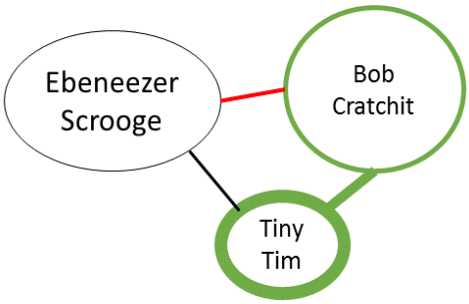

Figure 4. Subset of a social graph later in the novel.

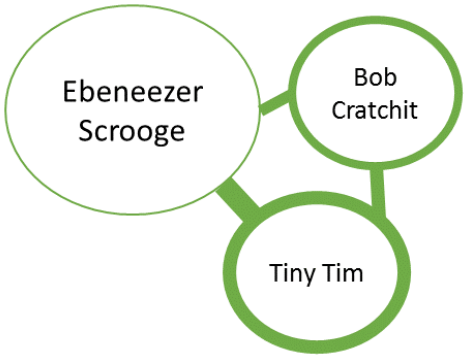

Figure 5. Subset of a social graph near the end.

The three figures above show the evolution of the social network as the story progresses. Earlier in the story, Ebenezer is an unpleasant (negative sentiment) character who is prominent if despised, especially by Bob Cratchit. As Ebenezer discovers his own flaws, the nature of his relationships with Cratchit and Tiny Tim evolve until, by the end, he is a much more positive person than he was, as are his relationships.

Social network graphs can be quite large, so the examples shown in this paper are kept to a legibly small size. We have found that to fully appreciate social graphs, an interactive graphing tool of the sort we have implemented is necessary. Being able to optional reduce the number of entities included in a social graph is an especially desirable feature.

6 Conclusions & Future Work

The framework presented describes an overall approach for social network extraction from textual material. Initial experiments with an implement of the framework has demonstrated good results that reflect a manual analysis of the same material. Further development is needed to improve efficiency of the implementation and accuracy of the social network extraction. Experiments first with the built-in NLTK Gutenberg Corpus of classic works of literature is planned, followed by an exhaustive study using a large set (N > 1000) of Project Gutenberg materials.

Social network extraction is compute-intensive, so we are exploring ways to reduce compute overhead. One approach that shows promise is to perform named entity recognition and co-reference resolution as separate phase that reads input and writes output. In this way, these time-consuming steps need only be performed a single time for each textual material analyzed, making repeat analysis less costly. Once these two costly steps are completed the remainder of social graph generation is relative fast and efficient.

7 References

[1] A. Beveridge and J. Shan, "Network of Thrones," Math Horizons Magazine, Vol. 23, No. 4, pp. 18-22, 2016

[2] Anthony Bonato, David Ryan D'Angelo, Ethan R. Elenberg, David F. Gleich, and Yangyang Hou. Mining and Modeling Character Networks. In Algorithms and Models for the Web Graph: 13th International Workshop, Montreal, Canada. Springer International Publishing, pages 100-114, November 11, 2016.

[3] J. Bonneau, J. Anderson and G. Danezis, "Prying Data out of a Social Network," 2009 International Conference on Advances in Social Network Analysis and Mining, Athens, 2009, pp. 249-254.

[4] Andrea Fronzetti Colladon and Elisa Remondi. Using social network analysis to prevent money laundering. Expert Syst. Appl. 67, 49-58, 2017.

[5] David K. Elson, Nicholas Dames, and Kathleen R. McKeown. "Extracting social networks from literary fiction." In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10). Association for Computational Linguistics, Stroudsburg, PA, USA, pages 138-147, 2010.

[6] Shalin Hai-Jew. Social Media Data Extraction and Content Analysis (1st ed.). IGI Global, Hershey, PA, USA. 2016.

[7] Leung C.K., Jiang F., Poon T.W., Crevier PÉ. Big Data Analytics of Social Network Data: Who Cares Most About You on Facebook?. In: Moshirpour M., Far B., Alhajj R. (eds) Highlighting the Importance of Big Data Management and Analysis for Various Applications. Studies in Big Data, vol 27. Springer, Cham, 2018.

[8] Yutaka Matsuo, Junichiro Mori, Masahiro Hamasaki, Keisuke Ishida, Takuichi Nishimura, Hideaki Takeda, Koiti Hasida, and Mitsuru Ishizuka. "POLYPHONET: an advanced social network extraction system from the web." In Proceedings of the 15th international conference on World Wide Web (WWW '06). ACM, New York, NY, USA, pages 397-406, 2006.

[9] Wouter De Nooy, Andrej Mrvar, and Vladimir Batagelj. Exploratory Social Network Analysis with Pajek. Cambridge University Press, 2018.

[10] Derek Ruths and Jürgen Pfeffer. Social media for large studies of behavior. Science. Vol. 346, Issue. 6213, pages 1063-1064, November 28, 2014.

[11] Todd Waskiewicz. Friend of a Friend Influence in Terrorist Social Networks. Air Force Research Laboratory, technical report, January 2012.

[12] M.C. Waumans, T. Nicodème, and H. Bersini. Topology Analysis of Social Networks Extracted from Literature. PLoS ONE 10(6), 2015.

[13] Yunqing Xia, Weifeng Su, Raymond Y. K. Lau, and Yi Liu. "Discovering latent commercial networks from online financial news articles." Enterprise Information Systems, Vol. 7, No. 3, pages 303-331, August 2013.