# Automated Phrase and Sentence Mining to Develop Graph Stories

Kevin Carman
Department of Computer Science
Elizabethtown College
carmank@etown.edu

*Abstract*-- **Graphs are everywhere and are growing increasingly compact with data. Understanding the data found within requires a summarization process to dwindle the information down to a human digestible amount. When summarizing text, translating how humans build connections and realize the semantic meaning of phrases to a computer is far from trivial. Some methods involve computing scores for each word based on how often they occur in a given text and computing word vectors to determine similarity between words. However, none of these processes by themselves work well to accurately summarize text. The result of my process shows that combining techniques can produce promising summarized representations of information. These results demonstrate the potential that finding the perfect combination of techniques has. I anticipate my process to be a starting point for more sophisticated combinations of methods to develop.**

***Keywords*-- Automation, summarization, phrase mining, sentence mining, graph sensemaking**

## I. Introduction

Graphs are everywhere, growing increasingly complex, yet still lack scalable, interactive tools to support sensemaking [1]. While there are several approaches to drawing graphs, recent edge decomposition algorithms, based on fixed-points of degree peeling, show strong potential in helping users explore graph data without being overwhelming. These edge decomposition algorithms discover peculiar subgraph patterns, quantify possible "roles" a vertex can play in the overall network topology, and scale to large graphs, making them perfect for extracting overall descriptive information. In particular, ATLAS enables a new paradigm for large graph exploration by generating explorable multi-layered representations of the data [1]. However, ATLAS still lacks tools to support efficient information extraction. The vertices and edges are represented in an easy to understand and interactive manner, but only visually. Through my automated phrase and sentence mining process, 'graph stories' can be developed to summarize the information within the visualization.

Graph stories in this context are defined as summaries generated from the data associated with each vertex. Each vertex contains metadata such as a link to a website or an ID of an object. This metadata is used to build a corpus of information about a particular subgraph which can then be summarized by my process. Even though the user may be able to visually understand the graph, when graphs have upwards of millions, if not billions, of edges and vertices, the decomposed graphs may still contain more information than a user is able to digest, leading to a continued lack of sensemaking.

## II. Related Works

Summarising text is an area of research that has encouraged the development of many tools and techniques. Efficient and automated sentence and phrase mining tools are useful for a myriad of tasks such as search engines returning relevant documents from a query or our beloved autocorrect learning how to fix our mistakes and predict our future phrases [2].

One method that inspired much of my process focused on summarizing news documents. A supervised model was introduced to predict word importance in a document and to better extract summaries [3]. The method showed promising results when comparing their process to summaries generated by other processes, but sadly, the code was

*2019 DIMACS REU funded by NSF Grant CCF-182215*

unable to be obtained within the timeline of this project to compare my process to.

Another project called gensim was created to realize unsupervised semantic modeling from plain text [4]. The project primarily uses scalable statistical semantics to determine semantically similar documents, but it also has a built in summarization function. This summarization function summarizes based on ranks of text sentences using a variation of the TextRank algorithm and was used as a comparison to evaluate the progress of this paper's process [5].

## III. Methods

### Dataset

The graph utilized for evaluation of my process consists of 3,774,768 vertices and 16,518,948 edges and was extracted from the National Bureau of Economic Research's U.S. patent data set [6][7]. The patents span 37 years (January 1, 1963 to December 30, 1999) and each vertex is associated with a patent ID while each edge signifies that a patent cites another patent.

### ATLAS and Data Acquisition

In the initial version of ATLAS, graphs were decomposed into layers of fixed points. Since the fixed points could still be massive, a second similar decomposition was used to generate graph waves which were more manageable. Still, graph waves could be massive, so the waves are also further broken down into fragments and represented at the fragment level [8]. The data is acquired via a script that accepts a fixed point number, wave number, and fragment number as parameters and returns a list of the patent IDs that can be found within a given fragment. Each patent ID is then queried to the Google Patents database and the title and abstract are returned and stored in separate files.

### Model Training

Two models need to be trained with relevant data before the process can run effectively. Using the same process as mentioned in the Data Acquisition section, but to a much larger scale, patent titles and abstracts are obtained, concatenated, and stored in a single file. The data required to train these models should be at least 500MB or greater to obtain reasonable results.

AutoPhrase is the first model that needs to be trained and is another automated phrase mining tool [9][10]. After it is trained, the phrasal segmentation function can be run on the training corpus to tag semantic phrases. There are two thresholds for single and multi-word phrases that can be tweaked before running this process. My testing found that setting the single word threshold to 10.0 and the multi-word threshold to 0.7 produced the best results. Once this is complete, we run a novel process to remove the phrase tags from the segmented corpus and reinsert the tagged phrases with spaces replaced by underscores.

The resulting file can then be used to train a word2vec model. Word2vec represents words in a vector space and can be used to determine similarities between phrases as will be explained in the phrase mining section [11].

### Phrase Mining

After the data has been acquired and the models have been trained, the phrases are now ready to be extracted. The titles of the patents are run through AutoPhrase to tag semantic phrases. The phrases are then parsed out of the segmented file and stemmed using the PorterStemmer algorithm to make sure that similar phrases with different endings are treated equally [12]. The phrases are then checked against a list of stopwords, provided by AutoPhrase, to be filtered. The TF-IDF (term frequency vs. inverse document frequency) scores are then calculated for each of the remaining phrases. The top-k phrases, where k = 10 in this particular case, are then selected from the list. The remaining top-k phrases are finally fed into the word2vec model to generate the top similar phrases for use later on.

### Sentence Mining

The final part of the graph story generating process is the sentence mining. The process starts off by loading in all of the unique sentences parsed from the abstracts. A set cover algorithm is applied to the sentence pool to determine how many of the top-k selected phrases can be found within each sentence. Lastly, the sentences are checked against a threshold where only the sentences with a phrase coverage score higher than the determined threshold are output to the graph story.

## IV. RESULTS

The results analyzed in this section pertain to a single, small fixed point that contains 22 vertices and 33 edges. The process was tested on several fixed points of varying size anywhere from 5 to 100 vertices. After manually reading through all of the associated patents in this fixed point, a short human-generated summary would describe them as all relating to various golf bag cover patents.

My phrase mining process generated the following list of phrases from the fixed point's data: [“golf bag(s)”, “high-resolution”, “high-speed”, “self-retaining”, “golf bag rain cover”, “protective cover”, “golf club(s)”, “slip cover”]. These phrases are highly representative of the data in the fixed point, except for the phrase “high-resolution”. This phrase comes from patent data that was incorrectly included in the data file. When querying to Google Patents, it very rarely returns data for a different patent due to it not being able to find the patent ID in question. This is a known bug with my process and this dataset when querying to Google Patents, but it was the only patent database with an API that allowed me to obtain both the titles and abstracts of patents in an efficient manner. Other databases that were tested either did not have an API that allowed this type of data to be returned, did not allow many sequential queries, or did not contain enough patent information to reliably return data. That being said, the rest of the phrases, especially ones such as “golf bag rain cover”, “golf bag(s)”, “slip cover”, and “protective cover”, are borderline perfect extractions from the provided data.

The similar phrases obtained from the word embedding of word2vec produced helpful results as well. Some high quality examples include “flexible cover”, “protective shield”, “fabric cover” and “removable cover” for the phrase “protective cover”. One thing that should be noted is that the number of terms in a phrase is inversely proportional to the quality of the similar phrases. As the number of terms in a phrase rises, the quality of the similar phrases decreases due to the frequency that they appear in the corpus. One example of this occurring is in the phrase “golf bag rain cover”, while it is a fantastic representation of the data by itself, the similar phrases include garbage such as “foil-tip”, “vertical joints”, and “73*”.

When comparing my phrase mining process to the gensim process, it can be seen that gensim rarely looks at phrases that consist of more than one word. In fact, in this scenario, the list of phrases generated from gensim have a maximum length of two, with about 90% of the list being single words. Not only that, but gensim also does not generate a top-k list of phrases, which results in having a massive list that may or may not be relevant.

Lastly, the following sentences were generated about this particular fixed point: ["A **flexible cover** has a hood with a first opening for receiving an open end of a **golf bag** to protect the **golf clubs** retained in the **golf bag**.", "A **golf bag protective cover** composed of plastic sheet material having slit openings therein which are covered by a plastic skirt which circumvents the entire cover, thus preventing dislodgement by wind while protecting the interior of the **golf bag** against ingress of rain.", "A **golf bag rain cover** composed of waterproof plastic material having a flap to which a golf scorecard pocket is sealed; the flap extends across the top opening and is sealed on both ends.", ”A **slip cover** for a **golf bag** and **clubs** to protect them from moisture during a sudden shower or rain storm."]. The phrases tagged in bold signify selected or similar phrases from the previous results. Without even seeing the entire corpus, any reader, or user in this case, can understand what the data pertains to with ease. The sentences selected by our process matched similarly to sentences selected by human evaluation.

When comparing my sentence mining process to the gensim process, a few things can be seen. Five different comparisons were made by adjusting the word count parameter from 0 to 200 in increments of 50. The 0 word count summary was roughly 25 sentences long and had many repeated sentences. While this was the only case where sentences were repeated, a common occurrence was that the sentences chosen were not as representative of the information as my process chose. The gensim results focused heavily on how the covers functioned rather than what the patents were.

## V. Discussion and Future Work

The results generated thus far have shown promise, but are far from perfect. As mentioned in the last section, a major weak point is in the acquisition of the data in this set. Not having a 100% reliable way to access the patent data adds uncertainty to the results. Luckily, this will likely only affect the results of small fixed points. Since larger fixed points have more data and therefore more robust calculations, the outlying data should not show up in the results. Another weak point is in the sentence mining process. It currently works well on datasets that are at least bigger than 20 vertices, but anything smaller than that has variable results. Sometimes it will not produce any results for a very small fixed point, which should not happen. This can easily be addressed, but it was not feasible within this research's timeline.

There are many ways I can think of to improve the process to get more accurate and consistent results. The easiest ones would be tweaking the single and multi-word phrase thresholds in AutoPhrase to get better results across a larger testing set, training our models with bigger corpora, and replacing the hyphens in hyphenated words with spaces to more accurately calculate the TF-IDF of phrases with respect to various writing styles. Some of the more involved ways include clustering phrases based on their semantic distance to provide a smaller, more diverse phrase set to the sentence mining process, testing out ranking functions such as BM25 ranking function for the sentence mining process [13], and testing the process over a wider variety of data sets both in terms of quantity and content and to evaluate the results anonymously and quantitatively.

## VI. Conclusion

Automated phrase and sentence mining are areas that will continue to develop as new tools and techniques are discovered, and as processing power increases, to further enhance the functionality and reliability of many of the things we use everyday. Although the goal of this project was to eventually implement my process into ATLAS as an interactive sensemaking tool, sadly, that did not occur within my given timeline as more research is needed to develop more robust and reliable mining methods.

## Acknowledgements

Thank you to NSF Grant CCF-182215 for supporting this work and to Dr. James Abello, Jingbo Shang, and Haodong Zheng for all their help and contributions to the project.

## References

[1] James Abello, Fred Hohman, Varun Bezzam, Duen Horng Chau, "Atlas: Local Graph Exploration in a Global Context," in *Proceedings of the International Conference on Intelligent User Interfaces*, ACM 2019.

[2] Jialu Liu, Jingbo Shang, Jiawei Han, "Phrase Mining from Massive Text and Its Applications," Morgan & Claypool, 2017.

[3] Kai Hong, Ani Nenkova, "Improving the Estimation of Word Importance for News Multi-Document Summarization," *Proc. of the 14th Conf. of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden, April 2014.

[4] Radim Rehurek, Petr Sojka, "Software Framework for Topic Modelling with Large Corpora," *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks,* ELRA 2010, Valletta, Malta.

[5] Federico Barrios, Federico Lopez, Luis Argerich, Rosita Wachenchauzer, "Variations of the Similarity Function of TextRank for Automated Summarization," 2016.

[6] J. Leskovec, J. Kleinberg, C. Faloutsos, "Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2005.

[7] "The National Bureau of Economic Research," *The National Bureau of Economic Research*. [Online]. Available: https://www.nber.org/.

[8] "Graph Waves, Part I, J. Abello, Dan, Sean, Qi", submitted for publication to IEEE *Information Visualization Conference*, *Infovis* 2019.

[9] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, Jiawei Han, "Automated Phrase Mining from Massive Text Corpora," accepted by IEEE Transactions on Knowledge and Data Engineering, Feb. 2018.

[10] Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren and Jiawei Han, "Mining Quality Phrases from

Massive Text Corpora," Proc. of 2015 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'15), Melbourne, Australia, May 2015.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of Word Representation in Vector Space," 2013.

[12] Martin Porter, "An algorithm for suffix stripping," *Program 14* no. 3, pp 130-137, July 1980.

[13] Stephen Robertson, Hugo Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval* 3, no. 4, 2009.

*2019 DIMACS REU funded by NSF Grant CCF-182215*