DETECT TEXT TOPICS BY SEMANTICS GRAPHS

Alex Romanova

Melenar, LLC, McLean, VA, US, 22101

ABSTRACT

It is beneficial for document topic analysis to build a bridge between word embedding process and graph capacity to connect the dots and represent complex correlations between entities. In this study we examine processes of building a semantic graph model, finding document topics and validating topic discovery. We introduce a novel Word2Vec2Graph model that is built on top of Word2Vec word embedding model. We demonstrate how this model can be used to analyze long documents and uncover document topics as graph clusters. To validate topic discovery method we transfer words to vectors and vectors to images and use deep learning image classification.

KEYWORDS

Graph Mining, Semantics, NLP, Deep Learning, CNN Image Classification.

1. INTRODUCTION

Nowadays data volumes are growing exponentially. For organizations that are daily getting huge amounts of unstructured text data, analyzing this data is too difficult and time consuming task to do manually. Topic analysis can solve document analysis problems as well as support other NLP problems such as search, text mining, and documents summarization.

Most common traditional approaches for topic analysis are topic modelings and topic classifications. Topic classifications as supervised machine learning techniques require topic knowledge before starting the analysis. Topic modelings as unsupervised machine learning techniques such as K-means clustering, Latent Semantic Indexing, Latent Dirichlet Allocation can infer patterns without defining topic tags on training data beforehand [1]. In this study we will introduce method of finding document topics through semantic graph clusters.

Word embedding methods such as Word2Vec [2], are conceptually based on sequential, logical thinking. These methods are capable of capturing context of a word in a document, semantic and syntactic similarity, and therefore solving many complicated NLP problems. However word embedding methods are missing capabilities to 'connect the dots', i.e. determine connections between entities. Understanding word relationships within documents is very important for topic discovery process and graph techniques can help to feel this gap.

In this article we will introduce a semantic graph model Word2Vec2Graph. This model combines word embedding and graph approaches to gain the benefits of both. Based on this model we will analyze long documents and uncover document topics as graph clusters. Document topics defined as semantic graph clusters will not only uncover sets of keywords, but will show relationships between words in topics.

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA - 2021
pp. 65-72, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110806

Our novel Word2Vec2Graph model, a semantic graph built on top of Word2Vec model is created on Spark - a powerful open source analytic engine [3] with libraries for SQL (DataFrames), graphs (GraphFrames), machine learning, and NLP [1]. Until recently there were no single processing framework that was able to solve several very different analytical problems in one place. Spark is the first framework for data mining and graph mining right out of the box.

Finding text document topics within semantic graph can be done using various community detection algorithms. In this paper we will use a simple community detection method - graph connected components - subgraphs where any two nodes are connected to each other by paths, and which are not connected to any additional nodes.

To validate topic correctness through method independent on semantic graph topic discovery method, we will transform word vectors to images and use Convolutional Neural Network image classification technique. Please see Figure 1 that shows the data flow diagram for the process of finding and validating document topics.

In this paper we propose a new, graph-based methodology, which has the following original contributions:

- Introduced a novel Word2Vec2Graph model that combines analytic thinking and holistic thinking functionalities in semantic graph.
- Established an ability of the Word2Vec2Graph model to analyze long documents and discover document topics as graph clusters with relationships between words in topics.
- Proposed CNN image classification method for topic validation.

In the pages that follow, we will show:

- Studies related to semantic graph building methods and algorithms of finding text topics based on semantics graphs.
- Process of building Word2Vec2Graph model by getting document pairs of words, training Word2Vec model and building a graph for pairs with high cosine similarities.
- Topic discovery method through calculating connected components and top PageRank words within components.
- Topic correctness validation method by deep learning CNN image classification.

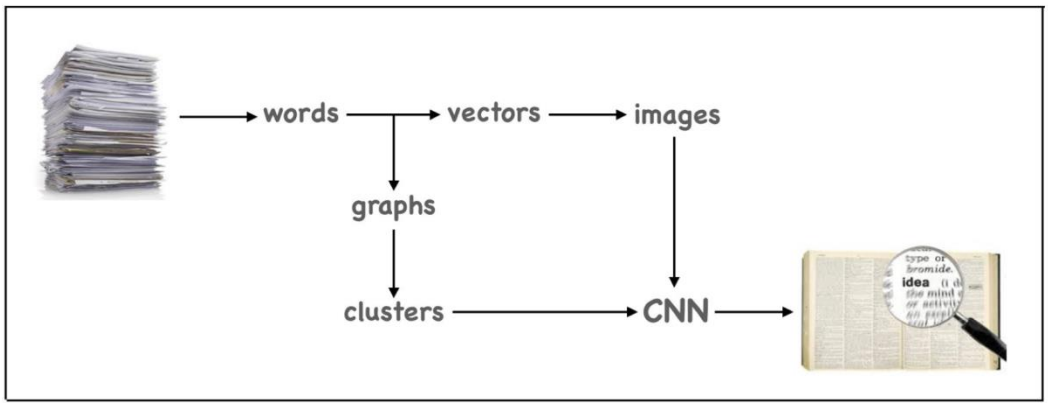

Figure 1. Finding text topics through a Word2Vec2Graph model and validating topics via CNN classification

2. RELATED WORK

There are various methods of building semantics graphs. Some of these methods are based on more traditional deep syntactic text analysis like RDF triples (subject–predicate–object) [4], other methods are based on unsupervised key phrase extractions and identifying statistically significant words [5] or on structuring asynchronous text streams [6].

Recently because of enormous progress of word embedding methods such as Word2Vec [2] some methods of building semantic graphs are based on word embeddings. For example, WordGraph2Vec method [7] is a semantic graph built on top of Word2Vec model that enriches text by adding target words for a specific context word in a sliding window.

Our Word2Vec2Graph model is similar to the WordGraph2Vec model [7] as in both models semantic graphs are built on top of Word2Vec. However in our semantic graph model we use pairs of words located next to each other in the document and mapping these words to vectors through Word2Vec model. For pairs of words we are calculating cosine similarities between words and building a graph based on threshold of pair similarities.

In recent years, there are some studies trying to integrate semantic graph structures with topic modeling. These models apply different methods of combining text with semantics graphs. Some studies integrate topic mining and time synchronization into a unified model [6] or combine semantic graphs with the textual information for topic modeling to estimate the probabilities of topics for documents [8]. Other studies are looking for topics through semantic graphs built on semantic relatedness between entities and concepts based on Wikipedia metadata [9]. In this paper to find topics we use a simple community detection method - graph connected components.

3. BUILD SEMANTIC GRAPH

To demonstrate our topic discovery method as data source we will use a document that consists of data about Creativity and Aha Moments that was manually extracted from several Wikipedia articles.

To build Word2Vec2Graph model and find document topics we will use Spark framework: Machine Learning and DataFrame libraries for Word2Vec model training and GraphFrame library for graphs. To process these methods, we will do the following:

- Retrain Word2Vec model.
- Extract pairs of words and calculate cosine similarities based on Word2Vec model.
- Build Word2Vec2Graph model.

Spark code is described in several posts of our blog[10].

3.1. Train Word2Vec Model

There are different approaches of using Word2Vec model for word embedding: using pre-trained model or training model on domain-specific corpus. Based on our observations, for topic finding Word2Vec models trained on domain-specific corpus work much better than pre-trained generic models. This observation corresponds with a study [11] that shows that domain-specific training corpuses work with less ambiguity than general corpuses for these problems.

To prove the difference, we trained two Word2Vec models. The first model was trained on generic corpus (News) and the second model was trained on combination of generic corpus and data about Stress extracted from Wikipedia (News + Wiki). In Table 1 you can see the differences of synonyms to words ‘Stress’ and ‘Rain’. As the word ‘Stress’ belongs to Stress corpus, the synonyms on these models are very different, but for a neutral word ‘Rain’ synonyms taken from these models are very similar.

Table 1. Examples of synonyms based on word2vec model corpuses: 'News' is word2vec model trained on generic corpus and 'News + Wiki' is word2vec model trained on combination of generic corpus and 'Stress' related corpus.

Stress		**Rain**	
News	**News + Wiki**	**News**	**News + Wiki**
risk	obesity	snow	snow
adversely	adverse	winds	rains
clots	systemic	rains	winds
anxiety	averse	fog	mph
traumatic	risk	inches	storm
persistent	detect	storm	storms
problems	infection	gusts	inches

Based on these circumstances, for topic discovery we will train the Word2Vec model on domain specific data corpus. Spark code for training and analyzing Word2Vec model can be found in our blog post [12].

3.2. Build Word2Vec2Graph Model

To build Word2Vec2Graph model, semantic graph on top of Word2Vec model, we will do the following steps:

- We will train Word2Vec model on the corpus that combines generic data (News) and domain specific data about Creativity and Aha Moment.
- From Creativity and Aha Moment data we will exclude stop words and tokenize other words.
- To discover document topics, instead of using a bag of words, we will look at pairs of words located next to each other in the document. To extract such pairs of words {word1, word2} we will use Spark Ngram function.
- For every word from word pairs we will get word vectors from Word2Vec model, i.e. for {word1, word2} pair we will map word1 to [word1 vector] and word2 to [word2 vector].
- Then we will calculate cosine similarities for word pairs, i.e. for {word1, word2} pair we will calculate cosine between [word1 vector] and [word2 vector].
- Finally, we will build a graph on word pairs with words as nodes and cosine similarities as edge weights. We will take only pairs of words with cosines higher than cosine similarity threshold 0.8.

Spark code for steps of building Word2Vec2Graph model can be found in our blog post [9].

4. UNCOVER AND VALIDATE DOCUMENT TOPICS

4.1. Uncover Document Topics

To detect document topics we will examine units of semantic graph that are separated from each other - graph connected components. Within each of these components we will find the most highly connected word using graph PageRank function.

For topic discovery we will do the following steps:

- Calculate connected components using Connected Components function from Spark GraphFrame library.
- Calculate graph PageRank scores by Spark PageRank function.
- For each connected component find the word with highest PageRank score and use this word as a topic class word.
- Map words to vectors and label vectors with topic class words.
- Transform vectors to images for CNN classification.

Spark code for topic finding and vector labelings can be found in our blog post [13].

4.2. Validate Topics

To validate topic correctness we will apply CNN image classification method. Vectors from uncovered topics will be converted to images with topic class words labels. Based on CNN image classification we will compare topics with image classes. This validation method does not fully prove topic modeling technique because clusters will have some noise: if two words are getting into the same image cluster it does not mean that they are highly connected. But if two words are in different image clusters they obviously do not belong to the same topic.

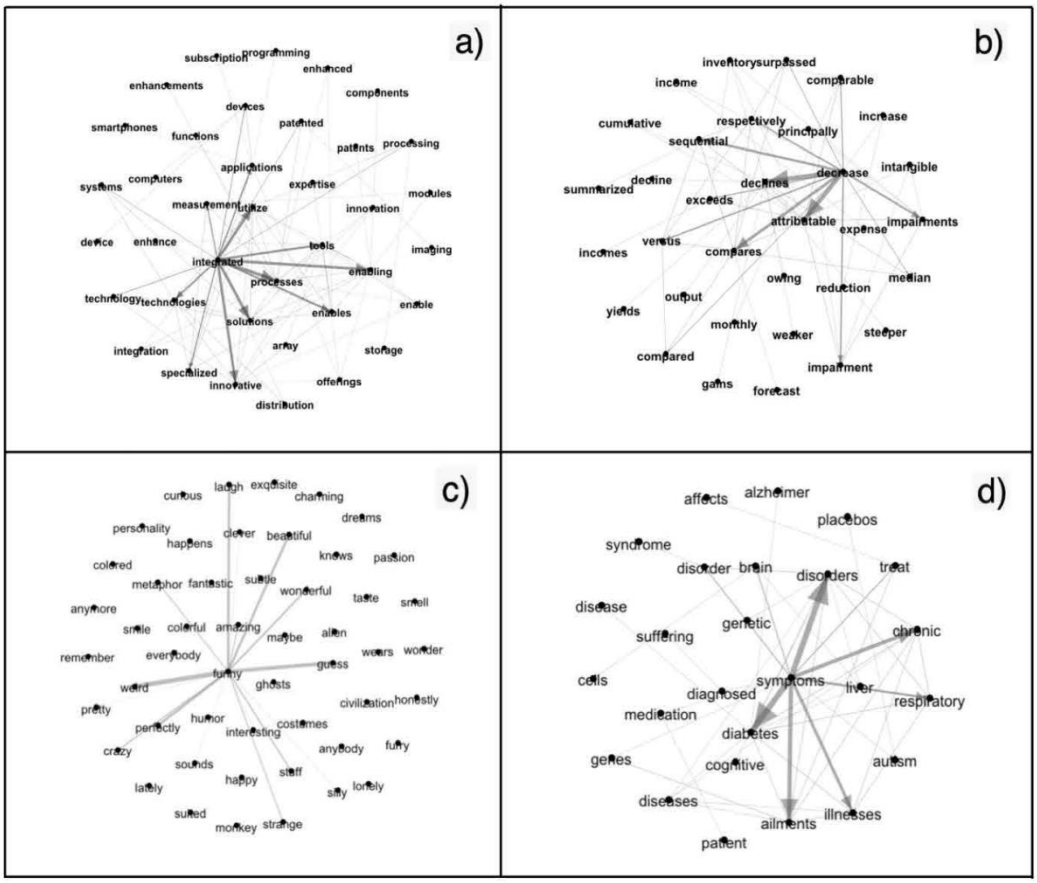

Figure 2. Subgraph topic examples: top PageRank words of topics: a) "integrated"; b) "decrease"; c) "funny"; d) "symptoms".

To convert vectors to images we will use Gramian Angular Field (GASF) - a polar coordinate transformation. The method was suggested by Ignacio Oguiza as a method of encoding time series as images for CNN transfer learning classification based on fast.ai library [14, 15]. To convert arrays to images and classify images we used open source code created by Ignacio Oguiza [16].

As usual, many graph connected components have very small sizes. For that reason for topics validation we used connected components with size bigger than 12 nodes. Our image classification model achieved accuracy about 91 percent.

4.3. Topic Examples

Topic examples are displayed in Figure 2. For each topic as a center of graph representation we use a topic class word and calculate a list of two degree neighbors (‘friends of friends’) around topics class words. Here are two degree neighbors for class word ‘symptoms’:

- symptoms -> brain; brain -> cells
- symptoms -> disorders; disorders -> cognitive

To find two degree neighbors we used Spark GraphFrame 'motif' technique [17] and transform the results to DOT language [18]. For graph visualization we used Gephi tool [19]. Spark code for graph visualization can be found in our blog post [13].

Topic visualization demonstrates an additional strength of using semantics graphs to uncover document topics: graph clusters that not only reveal sets of keywords in topics, but also demonstrate word relationships within topics.

5. CONCLUSION AND FUTURE WORK

In this paper we introduced a novel Word2Vec2Graph model that combines analytic thinking and holistic thinking functionalities in semantic graph. We demonstrated an ability of the Word2Vec2Graph model to analyze long documents and discover document topics as graph clusters that not only reveal sets of topic keywords, but also show word relationships within topics. For topic validation we suggested a novel CNN image classification method independent on semantic graph techniques.

In the future we are planning to do the following:

- Use more advanced word embedding models, like BERT, in particularly, examine phrase embedding process. Evaluate new Spark NLP library [1] that allows to fine tune various word embedding models and combine them with graph and machine learning models in Spark.
- Apply Word2Vec2Graph model to NLP problems that benefit from graph capacity to examine relationships between objects, such as entity disambiguation, semantic similarity, question answering, and others.
- Experiment with mapping words to vectors and vectors to images and classifying words and sequences of words through CNN image classification methods.

REFERENCES

[1] Alex Thomas (2020) *Natural Language Processing with Spark NLP*, O'Reilly Media, Inc.
[2] T Mikolov & I Sutskever & K Chen & GS Corrado & J Dean, (2013) "Distributed representations of words and phrases and their compositionality", Neural information processing systems.
[3] Bill Chambers &Matei Zaharia (2018) *Spark: The Definitive Guide: Big Data Processing Made Simple*, O'Reilly Media, Inc.
[4] Jurij Leskovec & Marko Grobelnik & Natasa Milic-Frayling, (2004). "Learning Sub-structures of Document Semantic Graphs for Document Summarization", LinkKDD 2004
[5] Juan Martinez-Romo & Lourdes Araujo & Andres Duque Fernandez, (2016). "SemGraph: Extracting Keyphrases Following a Novel Semantic Graph-Based Approach", Journal of the Association for Information Science and Technology, 67(1):71–82, 2016
[6] Long Chen and Joemon M Jose and Haitao Yu and Fajie Yuan, (2017) "A Semantic Graph-Based Approach for Mining Common Topics from Multiple Asynchronous Text Streams", 2017 International World Wide Web Conference Committee (IW3C2)
[7] Matan Zuckerman & Mark Last, (2019) "Using Graphs for Word Embedding with Enhanced Semantic Relations", Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13).
[8] Long Chen & Joemon M Jose & Haitao Yu & Fajie Yuan & Dell Zhang, (2016). "A Semantic Graph based Topic Model for Question Retrieval in Community Question Answering", WSDM'16
[9] Jintao Tang & Ting Wang & Qin Lu Ji & Wang & Wenjie Li, (2011). "A Wikipedia Based Semantic Graph Model for Topic Tracking in Blogosphere", IJCAI'11
[10] "Sparkling Data Ocean - Data Art and Science in Spark", http://sparklingdataocean.com/
[11] Yoav Goldberg & Graeme Hirst (2017) *Neural Network Methods in Natural Language Processing*, Morgan & Claypool Publishers.

[12] "Word2Vec Model Training", http://sparklingdataocean.com/2017/09/06/w2vTrain/
[13] "Word2Vec2Graph to Images to Deep Learning", http://sparklingdataocean.com/2019/03/16/word2vec2graph2CNN/
[14] Jeremy Howard, Sylvain Gugger (2020) *Deep Learning for Coders with Fastai and PyTorch*, O'Reilly Media, Inc.
[15] Zhiguang Wang & Tim Oates, (2015) "Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks", Association for the Advancement of *Artificial* Intelligence (www.aaai.org).
[16] "Practical Deep Learning applied to Time Series", https://github.com/oguiza
[17] "Motifs Findings in GraphFrames", https://www.waitingforcode.com/apache-spark-graphframes/motifs-finding-graphframes/read
[18] "Drawing graphs with dot", https://www.ocf.berkeley.edu/~eek/index.html/tiny_examples/thinktank/src/gv1.7c/doc/dotguide.pdf
[19] "Visual network analysis with Gephi", https://medium.com/@EthnographicMachines/visual-network-analysis-with-gephi-d6241127a336

AUTHOR

Alex Romanova Holds MS in mathematics from Faculty of Mechanics and Mathematics, Moscow State University and Ph.D. in applied mathematics from Faculty of Geography, Moscow State University, Moscow, Russia. She is currently a data scientist in Melenar, an expert in Knowledge Graph, NLP, Deep Learning, Graph Mining and Data Mining. Sharing her experience in technical blog: http://sparklingdataocean.com/