International Journal of Data Mining & Knowledge Management Process (IJDKP)
Vol.11, No.2/3/4, July 2021

SEMANTICS GRAPH MINING FOR TOPIC DISCOVERY AND WORD ASSOCIATIONS

Alex Romanova

Melenar, LLC, McLean, VA, USA

ABSTRACT

Big Data creates many challenges for data mining experts, in particular in getting meanings of text data. It is beneficial for text mining 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 to determine word associations and discover document topics. 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, get unexpected word associations and uncover document topics. To validate topic discovery method we transfer words to vectors and vectors to images and use CNN deep learning image classification.

KEYWORDS

Graph Mining, Semantics, Topics Discovery, Word Associations, Deep Learning, Transfer Learning, CNN Image Classification.

1. INTRODUCTION

Big Data creates many challenges for data experts, in particular on text data mining: 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. Automation of topic discovery and word associations can solve document analysis problems as well as support other NLP problems such as search, text mining, and documents summarization.

For topic discovery most common traditional approaches 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 capable of capturing context of a word in a document, semantic and syntactic similarity, and therefore solving many complicated NLP problems such as finding semantically related pairs of words. Based on Word2Vecsemantic similarity between two words is generally taken as cosine similarity of word vectors. However word associations, unlike cosine similarities, are expected to be asymmetric [3]. In semantic graph model that we introduce in this study we will be able to find not just directed pairs of associated words but also lines of associated words of any size.

DOI:10.5121/ijdkp.2021.11401

Word embedding models are conceptually based on sequential, logical thinking but they 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 methods to gain the benefits of both. Based on this model we will analyze text documents, find unexpected word association lines and uncover document topics. Document topics observed as semantic graph clusters will not only uncover keyword sets, but they will also show relationships between words within topics.

By looking at semantic graph mining techniques from geometrical view, we can see the following benefits:

- Traditional text mining techniques are based on bags of words with no dependencies between the words. This can be considered as zero-dimensional data mining.
- Semantic graph pairs of words can be used to find word neighbors, paths between the words or lines of associated words. It can be considered as one-dimensional data mining.
- Graph clusters determine community relationships within semantic groups and can be considered as multi-dimensional data mining. As Aristotle said: "The whole is greater than the sum of its parts".

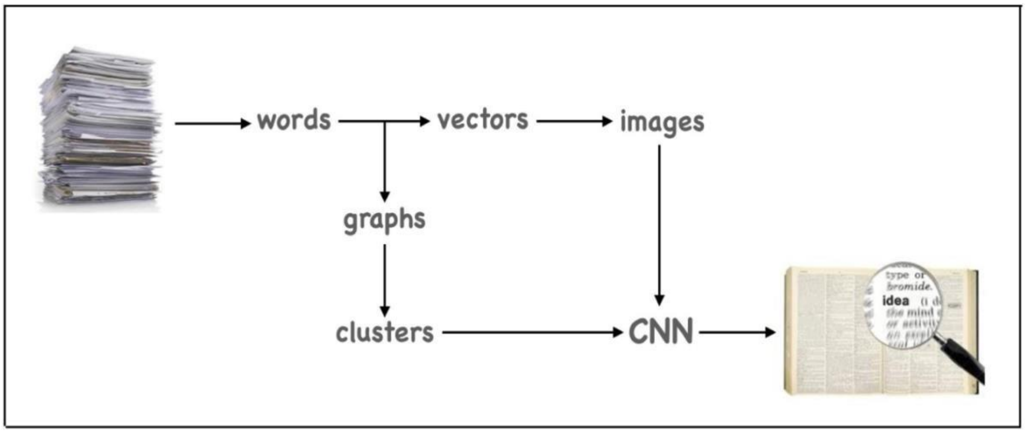

Figure 1. Finding text topics through a Word2Vec2Graph model and validating topics via CNN classification

As a language for building the Word2Vec2Graph model we use Spark — a powerful open source analytic engine [4] 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 by paths, and no nodes are connected to nodes from other subgraphs.

To validate topic correctness by method independent on semantic graph topic discovery, we will transform word vectors to images and use Convolutional Neural Network (CNN) 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, find unexpected associated word lines, and discover document topics.
- Proposed CNN transfer learning image classification method for topic validation.In the pages that follow, we will show:
- Studies related to semantic graph building methods and algorithms of semantics graph mining.
- Process of building Word2Vec2Graph model by training Word2Vec model, getting collocated pairs of words from data corpus and building a graph based on word pairs as edges and vector cosine similarities as edge weights.
- Topic discovery method by calculating connected components and using top PageRank words within components as topic class words.
- Topic correctness validation method by transfer learning CNN image 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) [5], other methods are based on unsupervised key phrase extractions and identifying statistically significant words [6] or on structuring asynchronous text streams [7].

Words association techniques are useful to extract word meaning from text data. Is some studies word associations are used to identify themes within sets of texts and are calculated based on statistical significancy of words within text subsets [8]. In other studies word associations are revealed through Word2Vec semantic similarity [9].

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 [10] 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 [10] 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 direct graph on pairs of words as edges and vector cosine similarities as edge weights. This allows us to find not only highly connected groups of words but also to find unexpected words associations.

In recent years, some studies are 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 [7] or combine semantic graphs with the textual information for topic modeling to estimate the probabilities of topics for documents [11].

Other studies are looking for topics through semantic graphs built on semantic relatedness between entities and concepts based on Wikipedia metadata[12]. For community detection of modern networks diverse methods are being used for sparse or dense graphs [13]. In this paper to find topics we are concentrated on sparse graphs and using a simple community detection method graph connected components.

CNN image classification techniques are very useful for image classification [14, 15]. In this study we used CNN image classification as independent method for topic validation. Transformation vectors to images was done with Gramian Angular Fields (GAF) methods, similar to techniques described in studies [16, 17].

3. METHODS

For this study we implemented the following methods:

- Retrained Word2Vec model on data corpus of interest.
- Built direct semantic graph using collocated word pairs as graph edges.
- Determined associated word lines.
- Created and analyzed graph clusters.
- Converted embedded vectors to GAF images and used CNN image classification to validategraph clustering accuracy.

For building and mining semantic graph we used Spark framework. Spark code is described in several posts of our technical blog [18].

3.1. Build Semantic Graph

In this paper we introduce a novel Word2Vec2Graph model —semantic graph model that combines word embeddings and graph functionalities.Word2Vec2Graphmodel will give us new insights like top words in text file — pageRank, word topics — connected components, word neighbors — 'find' function.

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.Spark Scala code is described in several posts of our technical blog [18].

3.1.1. Train Word2Vec Model

There are different practices 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 and word association computation Word2Vec models trained on domain specific corpus work much better than pre-trained generic models. This observation corresponds with a study [19] 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 ongeneric 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

One of the goals of semantic graph mining is to understand meanings of entity relationships. Based on these circumstances, to build Word2Vec2Graph model we will train Word2Vec models on domain specific datacorpora. Spark code for training and analyzing Word2Vec model can be found in our blog post [20].

3.1.2. Build Word2Vec2Graph Model

To build Word2Vec2Graph model we will do the following steps:

- We will look at pairs of words located next to each other in the document. To extract such pairsof words {word1, word2} we will use Spark Ngram(2) function.
- For words from word pairs we will get word vectors from Word2Vec model, i.e. for {word1, word2} pair we will map word1 to [word1, vector1] and word2 to [word2, vector2].
- Then we will calculate cosine similarities for wordpairs, i.e. for {word1, word2} pair we willcalculate cosine between [vector1] and [vector2].
- Finally, we will build a direct graph on word pairs with words as nodes, word pairs as edges and cosine similarities as edge weights.

Spark code for steps of building Word2Vec2Graph model can be found in our technical blog post [21].

3.2. Semantic Graph Mining

By comparing semantic graph mining methods with traditional text mining from geometrical view, we can see that traditional ‘bag of words’ method represents zero dimensional text mining, graph connections represent one dimensional text mining, and graph clusters represent multi-dimensional text mining.

In this study we will demonstrate:

- One-dimensional text mining techniques such as word neighbors, unexpected word associations and lines between the words.
- Multi-dimensional text mining techniques such graph clustering.

3.2.1. Lines between the Words

On Word2Vec2Graph model finding word neighbors can be done through Spark GraphFrame motif ‘find’ function. Using ‘find’ function for finding word neighbors is better understandable than more traditional way of finding word neighbors via Word2Vec model based on word synonyms.

```
   graph.find("(a) - [ab] -> (b)”)
```

Elegance of this style can be illustrated when looking for second degree neighbors, i.e. ’friends of friends’. Coding it via Word2Vec model would require self join word synonyms.

```
def foaf(graph: GraphFrame, node: String) = {
   graph.find("(a) - [ab] -> (b); (b) - [bc] -> (c)").
   filter($”a.id"=!=$"c.id").
   filter($”a.id"===node))}
```

The Spark GraphFrame motif ‘find’ method is conceptually similar to {subject - predicate -> object} and is better understandable than code for self-joining tabular data [22].

In addition to finding word neighbors this method is applicable to analysis of word to word associations. One of the ways to examine word to word connections is using Shortest Path GraphFrame function. The method we propose in this study will find word paths in direct graph using number of words as parameters.

Here is how to get single words between 'startWord' and‘endWord' :

```
val path=graph.
   find("(a) - [] -> (b); (b) - [] -> (c)”).
    filter($”a.id”===startWord &&$”c.id”===endWord)
```

and here is how to get any two words between 'startWord' and 'endWord':

```
val path=graph.
   find("(a) - [] -> (b); (b) - [] -> (c); (c) - [] -> (d)”).
    filter($”a.id”===startWord &&$”c.id”===endWord)
```

To find a predefined number ofwords (wordCount) in [startWord, endWord] line:

```
def formLine(wordCount:Int): String = {
   var line=new String
   for (i <- 1 to wordCount-1)
line+=“(x"+lit(i)+")-[]-
>(x"+lit(i+1)+");"
   line.substring(0, line.length() - 1)}
```

Examples of formLine function:

```
formLine(3)
(x1)-[]->(x2);(x2)-[]->(x3)
formLine(4)
(x1)-[]->(x2);(x2)-[]->(x3);(x3)-[]->(x4)
formLine(5)
(x1)-[]->(x2);(x2)-[]->(x3);(x3)-[]->(x4);(x4)-[]->(x5)
```

To get a predefined number ofwords (wordCount) in [startWord, endWord] line:

```
def findForm(graph: GraphFrame, startWord: 
String,endWord: String, wordCount: Int): 
DataFrame ={ val 
path=graph.find(formLine(wordCount))}
```

Detail explanation and Spark codeis published in our tech blog [23].

3.2.2. Graph Clusters

Finding graph clusters is very challenging process. In social network graphs this is called ‘community detection’. In this study we are using the simplest method — graph connected components. Graph connected components are subgraphs where every two nodes have path between them and none of the nodes is related to nodes outside of this subgraph.

In dense graph connected component practice the largest connected component usually contains large amount of graph nodes and therefore connected component method is not useful for community detection in dense graphs. On the contrary, community detection via this method works well for sparse graphs. Based on this we will propose two ways to come up with semantic graph clusters through connected components:

- Create sparse graphs based on small range threshold of word-vector cosine similarities.
- Create graphs with no limitations on word-vector cosine similarities but calculate connectedcomponents on small range edge weights.

In Experiments section of this paper we will demonstrate the first way for text topic discovery and the second way to observe unexpected word associations. The first method follows standard Spark GraphFrame Connected Component function and Spark code for second method is published in our blog post [24].

3.3. CNN Image Classification

Word2Vec2Graph model is built on top of word embedding model where word vectors are transformed to graphs. Word vectors can also be transformed to images and CNN image classification can be used as independent validation method.

3.3.1. Transform Vectors to Images

As a method of vector to image translation in this study we used Gramian Angular Field (GAF) — a polar coordinate transformation based techniques [16, 17]**.**This transformation method works well for images classification and data visualization. We were inspired by practice suggested on fast.ai forum by Ignacio Oguiza as a method of encoding time series as images and using fast.ai library for CNN image classifications.

3.3.2. Train CNN Image Classification Model

For this study we used fast.ai CNN transfer learning image classification. To deal with comparatively small set of training data, instead of training the model from scratch, we followed ResNet-50 transfer learning: loaded the results of model trained on images from the ImageNet database and fine tuned it with data of interest [25, 26].Python code for transforming vectors to GAF images and fine tuning ResNet-50 is described in fast.ai forum [27].

4. EXPERIMENTS

4.1. Source Data

For this study we used two domain-specific data corpuses: one data corpus about Creativity and Aha Moments and another data corpus about Psychoanalysis.

"Psychoanalysis" data was used to recognize unexpected word associations and "Creativity and Aha Moments" data corpus was used for text topic discovery and for calculating word association lines.

4.2. Word Associations

Word2Vec2Graph technique of finding text topics is conceptually similar to Free Association that is a practice in psychoanalytic therapy. We will show some examples that prove this analogy. As a text file we will use text data about Psychoanalysis extracted from multiple articles in Wikipedia.

In Free Association practice, a therapist asks a person in therapy to freely share thoughts, words, and anything else that comes to mind. In traditional free association, a person in therapy is encouraged to verbalize or write all thoughts that come to mind. Free association is not a linear thought pattern. Rather, a person might produce an incoherent stream of words, such as dog, red, mother, and scoot. They may also jump randomly from one memory or emotion to another. The idea is that free association reveals associations and connections that might otherwise go uncovered. People in therapy may then reveal repressed memories and emotions [28].

Word associations plays another important role in text mining: lines of associated words show meanings of word to word connections to better understand themes of the document.

One of the problems of word association calculation through embedding space like Word2Vec is that word associations are expected to be asymmetric but cosine similarities are symmetric [9].

To solve this problem we will build Word2Vec2Graph model as direct graph on pairs of words located next to each other in the text.

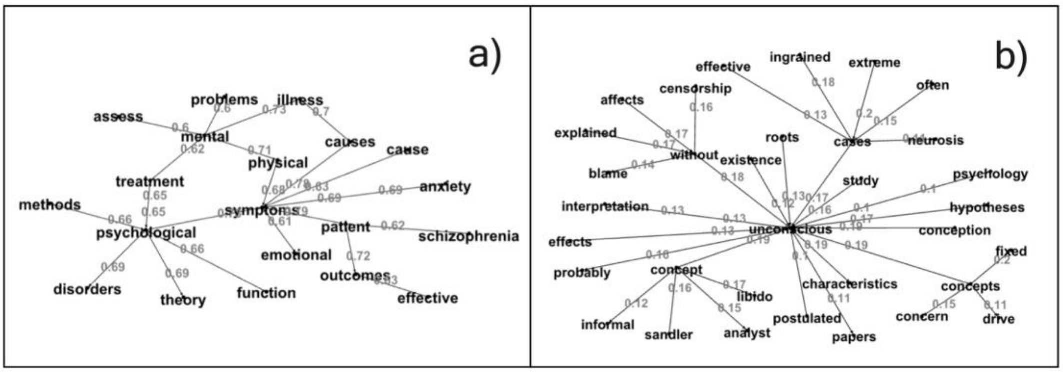

Figure 2. Free associations between words: semantics subgraphs on pairs of words with a) high cosine similarities; b) low cosine similarities.

Adjusted word pairs in the stream of words are expected to be similar and if so in Word2Vec2Graph model these pairs would have high cosine similarities. To uncover unexpected associated word pairs we will look at graph edges with low weights.

4.2.1. Low Weight Graph Clusters

In 'Graph Clusters' part of 'Semantic Graph Mining' section of this paper we propose two ways of building graph clusters. To find unexpected word associations we will use a method of building graph with no weight limitations and calculating connected components with weight threshold parameters. Then we will compare high edge weight clusters with low edge weight clusters.

Figure 2 shows examples of graph clusters built on connected components (a) with cosine similarities higher than 0.6 and (b) with low cosine similarities, between 0.1 and 0.2. Words in hight cosine similarity cluster look semantically closer to each other than words in low cosine similarity cluster.

4.2.2. Associated Word Lines

Another Word2Vec2Graph method of finding word associations is described in 'Lines between the Words' part of 'Semantic Graph Mining' section. As the data corpus for experiment we will use text data about 'Creativity and Aha Moments'.

Looking at word associations in Figure 2 we can see then word pairs with high cosine similarities are well known similar words - they are called "synonyms" in Word2Vec model. On the contrary, pairs of words with low cosine similarity represent unexpected associations and therefore they are more interesting to look at.

To find word associations in text data we will do the following:

- Train Word2Vec model on 'Creativity and Aha Moments' data corpus.
- Extract collocated pairs of words.

- Map these words to pairs of vectors.
- From pairs of vectors select pairs with cosine similarities less than 0.33 and build direct graph.
- Calculate word association lines between words.

Here are examples of word association movements between the words ‘brain’ and ‘insight’ that demonstrate how to get from ‘brain’ to ‘insight’ based on ‘Creativity and Aha Moments’ data corpus. One word between:

```
brain -> right -> insight
brain -> activity -> insight
brain -> moments -> insight
```

Two words between the words ‘brain’ and ‘insight’:

```
brain -> require -> spontaneous -> insight
brain ->  thought ->  called -> insight
brain -> dominant -> problem -> insight
brain -> response -> associated -> insight
```

Three words between the words ‘brain’ and ‘insight’:

```
brain  -> require  -> neural-> activity  -> insight
brain  -> dominant  -> problem-> called  -> insight
brain  -> functions  -> creator-> ideas  -> insight
brain -> thinking -> likely-> solve -> insight
```

4.3. Uncover and Validate Document Topics

Finding text document topics within semantic graph can be done using various community detection algorithms. In this paper 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.

To validate topic correctness by method independent on semantic graph topic discovery, 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.

4.3.1. Uncover Document Topics

For topic discovery we will use the first method described in ‘Graph Clusters’ part of ‘Semantic Graph Mining’ section of this paper: we will create a sparse graph based on high threshold of word-vector cosine similarities. 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. We will do the following steps:

- Retrain Word2Vec model on Creativity and Aha Moments data corpus.
- Extract collocated pairs of words and calculate cosine similarities based on Word2Vec model.
- Build Word2Vec2Graph model on pairs on vectors with cosine similarities higher than 0.8.
- Calculate graph clusters 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 asa 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 [29].

4.3.2. 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.

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[16, 26].To convert arrays to images and classify images we used open source code created by Ignacio Oguiza[30].

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

4.3.3. Topic Examples

Examples of topics of “Creativity and Aha Moments” data corpus are displayed in Figure 3. For each topic as a center of graph representation we used a topic class word and calculated a list of two degree neighbors (‘friends of friends’) around topics class words. For example, here are two degree neighbors for the class word ‘symptoms’:

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

To find two degree neighbors we used Spark GraphFrame ‘motif’ technique [31]and transformed the results to DOT language[32]. For graph visualization we used Gephi tool [33]. Spark code for graph visualization can be found in our blog post [29].

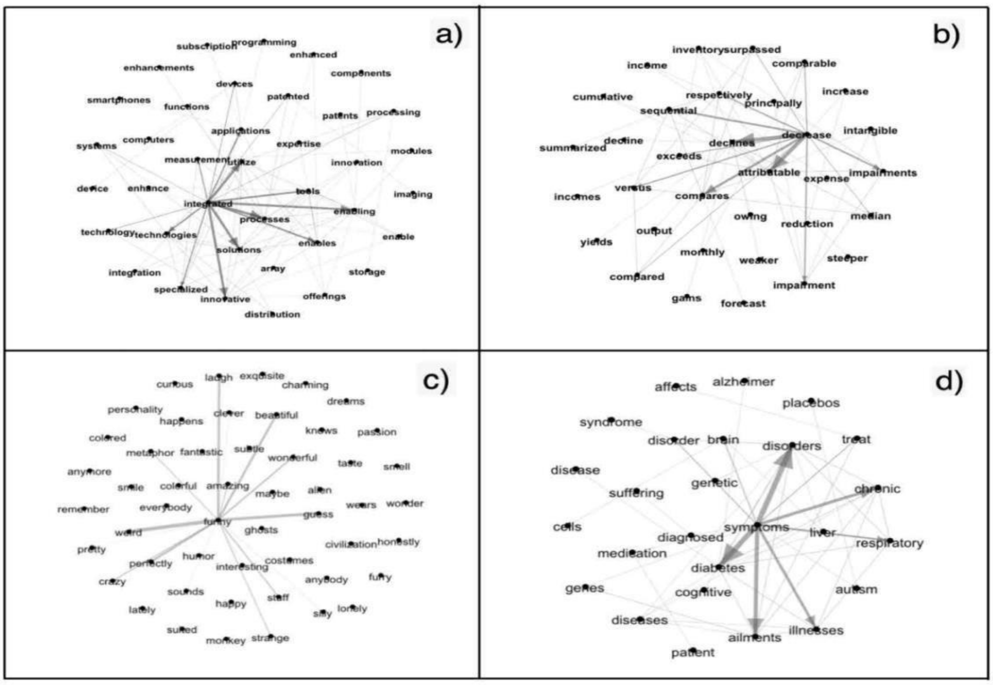

Figure 3. Subgraph topic examples: top PageRank words of topics: a) "integrated"; b) "decrease"; c) "funny"; d) "symptoms".

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 semantic graph model Word2Vec2Graphthat combines analytic thinking and holistic thinking functionalities. We demonstrated an ability of the Word2Vec2Graph model to analyze long documents, find unexpected word associations, calculated word association lines, and discover document topics. Document topics that are calculated as graph clusters not only reveal sets of topic keywords, but also show word relationships within topics. For topic validation we suggested independent method CNN transfer learning image classification.

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.

6. BROADER IMPACT

In this study for text data exploration we used a hybrid of independent techniques - semantic graph mining and deep learning image classification. Both of these techniques are implemented by transforming text to embedded vectors and transforming vectors to images for CNN image classification and transforming vectors to graphs for graph mining.

The combination of graph and CNN image classification practices can also be used for other data mining scenarios. In this study we started data exploration with graph mining and used CNN image classification as validation method. On the contrary data investigation can start with CNN image classification and use graph mining to uncover patterns on lower levels of granularity [34].

In addition to text data, both techniques can be applied to a variety of embeddable entities such aswords, documents, images, videos, and many other [35].

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] Andrew Cattle and Xiaojuan Ma, (2017) “Predicting Word Association Strengths”, 2017Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1283–1288.
[4] Bill Chambers &Matei Zaharia (2018) *Spark: The Definitive Guide: Big Data Processing Made Simple*, O'Reilly Media, Inc.
[5] Jurij Leskovec & Marko Grobelnik & Natasa Milic-Frayling, (2004). "Learning Substructures of Document Semantic Graphs for Document Summarization", LinkKDD 2004
[6] 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
[7] 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)
[8] Michael Thelwall, (2021) “Word Association Thematic Analysis: A Social Media Text volume 13, pages i-111
[9] Andrew Cattle and Xiaojuan Ma, (2017) “Predicting Word Association Strengths”, 2017 Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1283–1288
[10] 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).
[11] 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
[12] Jintao Tang & Ting Wang & Qin Lu Ji & Wang & Wenjie Li, (2011)."A Wikipedia Based Semantic Graph Model for Topic Tracking in Blogosphere", IJCAI’11
[13] Stavros Souravlas & Angelo Sifaleras & M Tsintogianni & Stefanos Katsavounis, (2021). "A classification of community detection methods in social networks: A survey", International Journal ofGeneral Systems 50(1):63-91
[14] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller: Deep learning for time series classification: a review. Data Min Knowl Disc 33, 917–963 (2019)
[15] Nima Hatami, Yann Gavet, Johan Debayle: Classification of time-series images using deep convolutional neural networks Conference: Tenth International Conference on Machine Vision (ICMV 2017).
[16] Zhiguang Wang, Tim Oates: 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)(2015)

[17] Zhiguang Wang, Weizhong Yan, Tim Oates: Time series classification from scratch with deep neural networks: A strong baseline. International Joint Conference on Neural Networks (IJCNN)(2017)

[18] "Sparkling Data Ocean - Data Art and Science in Spark", http://sparklingdataocean.com/

[19] Yoav Goldberg&Graeme Hirst (2017) *Neural Network Methods in Natural Language Processing*, Morgan & Claypool Publishers.

[20] "Word2Vec Model Training", http://sparklingdataocean.com/2017/09/06/w2vTrain/

[21] "Introduction to Word2Vec2GraphModel",http://sparklingdataocean.com/2017/09/17word2vec2graph

[22] Alex Romanova, (2020) "Building Knowledge Graph in Spark Without SPARQL", Database and Expert Systems Applications, DEXA 2020 International Workshops BIOKDD, IWCFS and MLKgraphs, Bratislava, Slovakia, September 14–17, 2020, Proceedings.

[23] "Find New Associations in Text",http://sparklingdataocean.com/2018/04/04/word2vec2graphInsights/

[24] "Word2Vec2Graph Model and Free Associations",http://sparklingdataocean.com/2017/12/24/word2vec2graphPsychoanalysis/

[25] Practical Deep Learning for Coders, https://course.fast.ai/ (2020).

[26] Jeremy Howard, Sylvain Gugger: Deep Learning for Coders with fast.ai and Py-Torch. O'Reilly Media, Inc. (2020).

[27] Time series/ sequential data study group, https://forums.fast.ai/t/time-series-sequential-data-study-group/29686 (2019)

[28] "GoodTherapy: PsychPedia: Free Association", https://www.goodtherapy.org/blog/psychpedia/free-association-in-therapy (2019).

[29] "Word2Vec2Graph to Images to Deep Learning",http://sparklingdataocean.com/2019/03/16/word2vec2graph2CNN/

[30] "Practical Deep Learning applied to Time Series", https://github.com/oguiza

[31] "Motifs Findings in GraphFrames", https://www.waitingforcode.com/apache-spark-graphframes/motifs-finding-graphframes/read

[32] "Drawing graphs with dot", https://www.ocf.berkeley.edu/~eek/index.html/tiny_examples/thinktank/src/gv1.7c/doc/dotguide.pdf

[33] "Visual network analysis with Gephi", https://medium.com/@EthnographicMachines/visual-network-analysis-with-gephi-d6241127a336

[34] "EEG Patterns by Deep Learning and Graph Mining",http://sparklingdataocean.com/2020/08/19/brainGraphEeg/

[35] Something2vec,https://gist.github.com/nzw0301/333afc00bd508501268fa7bf40cafe4e(2016)

AUTHOR

Alex Romanova Holds MS in mathematics from Faculty of Mechanics and Mathematics, Moscow State University and Ph.D. in applied mathematics from Facultyof 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/