M U L T I P L E C R I T E R I A D E C I S I O N M A K I N G

Vol. 15 2020

Mohit Kumar Barai[*]
Subhasis Sanyal[**]

DOMAIN SPECIFIC KEY FEATURE EXTRACTION USING KNOWLEDGE GRAPH MINING

DOI: 10.22367/mcdm/2020.15.01

Received: 7.01.2021 | Revised: 21.04.2021 | Accepted: 14.09.2021.

Abstract

In the field of text mining, many novel feature extraction approaches have been propounded. The following research paper is based on a novel feature extraction algorithm. In this paper, to formulate this approach, a weighted graph mining has been used to ensure the effectiveness of the feature extraction and computational efficiency; only the most effective graphs representing the maximum number of triangles based on a predefined relational criterion have been considered. The proposed novel technique is an amalgamation of the relation between words surrounding an aspect of the product and the lexicon-based connection among those words, which creates a relational triangle. A maximum number of a triangle covering an element has been accounted as a prime feature. The proposed algorithm performs more than three times better than TF-IDF within a limited set of data in analysis based on domain-specific data.

Keywords: feature extraction, natural language processing, product review, text processing, knowledge graph.

[*] Samsung Research Institute, Noida, India, e-mail: m.barai@samsung.com, mhtbarai547@live.com, ORCID: 0000-0002-7258-9825.

[**] Samsung Research Institute, Noida, India, e-mail: s.sanyal@samsung.com, subhasis_howrah@yahoo.co.in, ORCID: 0000-0003-1188-0907.

1 Introduction

Online consumer reviews consist of indefinite statements based on a specific product (Park, Kim, 2008). Open-ended comments exhibit reviewer's judgment of a product based on negative and positive polarity (Willemsen et al., 2011). This type of open-ended textual content is cognate with customer satisfaction. Customer satisfaction is a metric to quantify the degree to which a customer will react based on a subject or the aspects of the subjects. Here the subject is a product, and aspects are the features. Knowledge discovery (Feldman, Dagan, 1995) from textual corpus refers to the process of bringing-out thought--provoking patterns or knowledge from unorganized text documents. In this case, it is the review data. Now, gaining knowledge from a vast database can be challenging (Houari, Rhanoui, Asri, 2015). Therefore, the primary intent is to get the most talked-about features or aspects.

Here we propose a novel method using a graph-based approach to extract key features from product review data. To generate a language pattern, POS (Parts of Speech) tagging is imperative. As post-POS tagging, we can extract features that are nouns. After doing POS tagging, we need to extract features that are nouns. Nouns are to be considered the main feature we are looking for in the central database. As per Oxford (www 8), "a word (other than a pronoun) used to identify any of a class of people, places, or things (common noun), or to name a particular one of these to know as a noun". We can consider a noun as our main feature, which we should look for as our main subject or aspect of the subjects. For example, consider the statement, '*The battery life of this camera is too short*'. As we can observe, '*battery*' and '*camera*' are those two entities, we can consider these words as a subject (or aspect), and the user's review is based on these two subjects (or aspects) that are '*camera*' and '*battery*'; also, this corroborates that identifying domain product features that are talked about by customers by using the manually tagged POS belongs to nouns (Htay, Lynn, 2013). Hence, a noun can be considered a subject or an aspect. Our whole idea is based on pivoting the noun as the main feature.

Selecting a noun as a central feature can be considered as a bias. While developing this algorithm, we are incorporating this bias from our expertise in this particular domain. Pre-mentioned is a kind of inductive bias. The word '*bias*' suggests an awareness of the predetermined notion instead of the neutral evaluation of reality (Campolo et al., 2018). In this sense, the world around us is biased. Most machine learning techniques have a predisposition towards this projection of bias. This type of bias is historical bias. It is often explored by comparing the relation between features or aspects of the elongated domain

prowess. Zhao et al. (2017) show that if we compare the label '*cooking*' in a particular data set, it co-occurs inequitably in women more than men. Since most machine learning approaches rely on correlations, such biases may proliferate to learned models or classifiers.

Similarly, we can assume that when reviewing product review data for mobile phones, words like '*battery*' and '*camera*' co-occur with the mobile phone. Also, let's consider some other POS such as Adjective, Adverb, and Verb. We have developed this algorithm considering mobile review data from an e-commerce platform. As per Merriam-Webster (www 1), "An adjective is a word belonging to one of the major form classes in any of numerous languages and typically serving as a modifier of a noun to denote a quality of the thing named, to indicate its quantity or extent, or to specify a thing as distinct from something else", for instance, '*Camera is good*'. Here '*good*' is an adjective which denotes the quality of the thing named which is nothing but the camera. It is a noun. Also, as per Merriam-Webster (www 1) "a word that characteristically is the grammatical centre of a predicate and expresses an act, occurrence, or mode of being, that in various languages is inflected for agreement with the subject, for tense, for voice, for mood, or aspect, and that typically has full descriptive meaning and characterizing quality but is sometimes nearly devoid of these especially when used as an auxiliary or linking verb", for instance, '*Camera hangs a lot*'. Here '*hang*' is a verb that expresses an act, occurrence, or mode of being, that in various languages is inflected for agreement with the subject '*camera*'. Now, most reviews will have both positive and negative comments (Safrin et al., 2017). Also, it has been observed that in the case of a vast data set, people tend to use synonyms to describe product features or use the exact words. For instance, '*The camera quality is awesome*' and '*The camera is super*'. Here, '*awesome*' and '*super*' bear lexically similar meaning and are associated with a noun subject which is here '*camera*'. It conveys a positive sentiment. Similarly, in '*Camera is worst*' the word '*worst*' and '*super*' again have lexical antonym property that carries with the subject noun '*camera*'. During our research work, we have found out that if we build a graphical model considering all available nouns, adjectives, verbs, and adverbs as vertices and create an edge between each pair of vertices based on some relations, then we will develop relational triangles and the most significant number of triangles will contribute to the most talked about feature. If we can create a dictionary based on words with lexically similar meaning and associate it with our maximum number of triangles, we will likely encounter the most talked-about features. In this case, we are considering a weighted knowledge-based graph. Here we have also tried to use the *n*-gram model. The main goal of

the *n*-gram model is to predict the context from the target word; the model transposes the contexts and targets and attempts to predict each context word from its target word. The main objective becomes to predict the context. We can consider a forward and backward window like the *n*-gram concept for this surrounding the target word, which is to be used for context prediction. The contexts words are nothing but the noun; in our case, we are more interested in the context words, which are nouns, surrounded by adjectives, adverbs, and verbs. So, we can say that a noun is our target word. The backward and forward windows must have the same size. Now we are focusing on finding the relation between adjective, adverb, and verbs. We have observed that since the main corpus is about review feedback, it must be associated with words that convey positive or negative polarity. We tried to bind these words with their synonym and antonym properties. For this, we have built a dictionary and trained our model with it. The output is a triangle, from where we can consider the feature that people have talked about the most. The most significant number of triangles associated with a noun is the most talked-about feature. In the following section, we will describe our approach in more detail.

2 The objective of the study and the novelty of the work

Feature extraction in Natural Language Processing (NLP) using graph theory is a new research field. Many research workers have proffered countless ideas. Hitherto the associated work (Markov, Last, Kandel, 2007; Wang, Do, Lin, 2005) has given special attention to the collocation of words and their recurrence as graphs instead of the sentence's linguistic interpretation. One research paper (Sidorov et al., 2013) has propounded linguistic information and word order in a graph for text classification; unfortunately, the result was limited to minimal texts of between 8 to 13 tokens. Shi et al. (2017) have proposed an idea to extract key phrases using knowledge graphs. They emphasized the latent relationship between two key terms (nouns and named entities) without instigating many random noises. As per them, sizeable experiments over real data show that the proposed conviction outperforms the state-of-the-art methods, including the graph-based co-occurrence methods and statistic-based clustering methods. There are two types of keyphrase extraction, supervised and unsupervised. The majority of the supervised methods accentuate key phrase extraction as a binary classification task (Hult, 2003a; 2003b; Jiang, Hu, Li, 2009; Turney, 2002; Witten et al., 1999) and evaluate some other features, such as term frequency-inverse document frequency (TF-IDF) and the position of the first occurrence of a phrase, as the inputs of a Naive Bayes classifier (Russell,

Norvig, 2003). As per Shi et al. (2017, p. 1), "This is extremely expensive and time-consuming in domain-specific scenarios. To reduce manpower, investigating comparative unsupervised methods is highly desired. Thus, we focus on studying unsupervised methods to extract key phrases from a single input document (e.g., news and article)". In our proposed algorithm, we have amalgamated the concept of the knowledge graph and the term frequency based on a context of target words (noun), which is formed by an *n*-gram model. After that, we have attempted to create a relational triangle surrounding the target word. The maximum weighted triangle considers a target word (noun) which is the most talked-about feature with our proposed algorithm. The surplus words with low or no semantic meaning must be filtered out. Such words are known as stop words (Jaideepsinh, Jatinderkumar, 2016). While building a feature extraction algorithm apart from the default stop word, we need to remove some stop words manually. We are doing a feature extraction from mobile review data extracted from Amazon for a particular mobile phone from a specific company. Our objective is to find out the most negatively reviewed features. So, in this case, company names like '*Samsung*', '*Apple*' can all be considered stop words, since we are looking for the product features. We are not looking for the company that has created it. We are focused on evaluating the product. Manual removal of stop words is an uphill task; also, it can contribute to the degradation of the feature extraction model. With the proposed model, dependency on the stop word is somewhat eliminated. In their paper, Stuart Rose et al. (2010) proposed a key feature extraction algorithm, RAKE (Rapid Automatic Key Feature Extraction). Its input consists of a stop word list, a set of phrase delimiters, and word delimiters. It uses stop words and phrase delimiters to segregate the input text into candidate keywords, which are sequences of content words in the text. Co-occurrences of words within these candidate keywords identify word co-occurrence. It helps us to generate the score for candidate keywords. RAKE is a well known and widely used feature extraction algorithm, which tends to give compound words or phrases as key features that are not helpful while looking for particular words. When we apply it in mobile review data from an e-commerce website, we get compound outcomes of the type '*used camera*' or '*automatically camera close*' or '*good battery life*'. These phrases or compound words are not very helpful when we want to know about a specific feature or aspect. The same problem persists with another well-known algorithm YAKE (Yet Another Keyword Extractor) (Campos et al., 2020). Our proposed algorithm has overcome this challenge. It does not generate a compound word or phrase, but provides a single word as a critical feature. We can also consider TF-IDF (Term Frequency-Inverse Document Frequency) with Bag of Words

(BOW) for key feature extraction from text. TF-IDF is a product of the word frequency (Term Frequency) and of the measure how common or rare is that particular word in all the documents (IDF). The problem with this algorithm is that it does not capture semantics; hence, to extract a topic's features can be a tedious task. Our algorithm has tried to overcome this problem by extracting the probable features considering the semantics. This is the reason we have incorporated a concept of 'sentiN-gram', which is a fixed-sized forward and back window pivoting the probable key features (noun).

3 Literature review

Feature extraction from colossal data is a crucial task, and it is one of the parts of Natural Language Processing. Sammons et al. (2016) showed that implementing a machine-learning algorithm is unequivocal while extracting key features where the programmatic approach hinders the essence of key feature extraction. For decades, constructing a pattern recognition has required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data into a suitable internal representation or a feature vector in which the learning subsystem, often a classifier, could detect or classify patterns in the input (LeCun, Bengio, Hinton, 2015). Multiple graph-based approaches have been proposed in the field of Information Retrieval (I.R.); we have gone through some of them in our research. One of the papers (Devika, Subramaniyaswamy, 2019) dealt with a semantic graph-based keyword extraction model using a powerful social data ranking method. The authors used numeric graph metrics to associate the nodes' weight in the semantic graph. After that, they applied page ranking algorithm to arrange the nodes, which provided the most influential nodes. In another approach, the researchers provided a graph-based keyword extraction model using collective node weight (Biswas, Bordoloi, Shreya, 2018). They attempted to determine the importance of keywords by collectively taking various influencing parameters. This is one of the states of art in the field of knowledge graph.

A Knowledge Graph (K.G.) is a systematic representation of facts, consisting of entities and their relationships. Entities are real-world objects or abstract concepts. Relationships depict the relation between individual entities within a boundary. Semantic definitions of entities and their associations constitute types and properties with a comprehensible meaning (www 2). To learn unambiguous linguistic and semantic word relationships from highly distributed vector representations, a Knowledge Graph model provides an excellent result. In this paper, the researchers discussed using a knowledge graph to identify

concept prerequisites (Manrique, Pereira, Mariño, 2019). They proposed a four--step approach consisting of building a knowledge graph to find probable candidate concepts; create potential pictures; formulate a model to evaluate possible ideas, and validate the idea using ground truth concepts from different domains. In another paper, researchers proposed a global level relation extractor model using knowledge graph embeddings for document-level inputs (Kim et al., 2020). This model creates a local-level knowledge graph from the input document, which will predict the global level relation from an extensive record. The synchronization between these two levels has been achieved during training. During our literature review, we have seen that the use of knowledge graphs is very pertinent for feature extraction (Zhao, Pan, Yang, 2020; Xu et al., 2020; Wang et al., 2018; Jia et al., 2018; K-CAP '19, 2019). Using a knowledge graph gives a graphical semantic view of a topic and associated aspects of the subject. This is the reason we have incorporated the concept of knowledge graph in the formation of this algorithm. Bonatti et al. (2018) stated that "Human and Social Factors in Knowledge Graphs" provided more concrete insights as it could build on both academic and industrial research results, projects, and practical experiences. Knowledge graphs capture relevant domain knowledge, and with machine learning algorithms, we can train our model to find out a specific pattern within that particular domain. This concept of knowledge graph is the driving force behind our algorithm.

4 Methodology

As we have said earlier, our algorithm is based on graph theory, knowledge graph, and *n*-gram model; also, we have integrated sentiment analysis. Sentiment analysis is a nexus of methods, techniques, and tools to identify and obtain personal information, such as opinion from natural language (Liu, 2009). Conventionally, sentiment analysis accentuates opinion polarity, i.e., whether someone has conveyed positive, neutral, or negative views towards something (Dave, Lawrence, Pennock, 2003). The quintessence of sentiment analysis has typically been a product or a service whose review has been made public on the internet (www 3). Hence, our primary focus is to extract the features based on the reviewer's sentiment in our research paper. We will now give the basics of Graphs, Knowledge Graph, and *n*-gram. A graph is denoted as $G = <V, E_i>$, where E_i can be defined as the set of vertices (nodes) V, and the interactions among pairs of nodes called links (edges) E. "A graph associated with each edge E (also called arc) is an ordered pair. Edge E is then directed from vertex U to vertex V, and an arrowhead on edge shows the direction. A graph is undirected if

the end vertices of all the edges are unordered (i.e., edges have no direction)" (www 4; www 5). A Knowledge Graph (K.G.) is a multi-relational graph composed of entities (nodes) and relations (different types of edges). Each edge is represented as a triple of the form (head entity, relation, tail entity), also called a fact, indicating that two entities are connected by a specific relation, e.g., (Alfred Hitchcock, director of, Psycho). Although effective in representing structured data, the underlying symbolic nature of such triples usually makes K.G.s hard to manipulate (Wang et al., 2017, p. 5). A linguistic model can take a list of words and attempt to predict the word that follows them. It outputs a probability score for all the words it knows. The *n*-gram model is a linguistic model. *N*-gram means a sequence of *N* words. The definition of *n*-gram is an unambiguous definition. For instance, 'good camera' is a 2-gram, 'Display is not good' is a 4-gram. While building an NLP model with the help of *n*-gram, we can assume that it will have a pretty good idea of the 'probability' of a word's occurrence after a specific word or before a specific word. Below is our training database with seven reviews given by a customer.

Table 1: Sample dataset to explain the notion of a Senti-*n*-gram

i)	Camera quality is average
ii)	Camera quality not good
iii)	I like the camera quality
iv)	Not satisfied with camera quality
v)	The camera quality is excellent
vi)	Camera quality is average
vii)	Camera quality is also good

From this, we can see that after the word '*camera*', only the word '*quality*' occurs, which this is expected, because our central database is based on product review data. So, the term '*quality*' has a special place while providing a product review to calculate the probability of the sequence, and we have:

$$\frac{|(W_1 W_2)|}{|(W_2)|}\,. \tag{1}$$

Here we calculate the probability of the word W_1 occurring after the word W_2; as stated earlier, the following algorithm adds the sentiment analysis concept. Consider the above database with seven review data from Table 1, where we can see the customers' feelings about the camera. So, all the sentences associated with '*camera*' must contain a word that describes a positive, negative or neutral sentiment. In Table 1, we can see that sentences (iii), (v), and (vii) all convey a positive sentiment, and this is due to the words: '*like*', '*excellent*', '*good*'

which occur next to '*camera*'. Here we can consider '*camera*' as a fixed element and assume an *n*-gram model before and after the '*camera*' is the same size. Within this *n*-gram, we can look for the words which convey positive and negative polarity. As we can see from our example, if we consider a window or an *n*-gram of 3 before and after the word '*camera*', we will undoubtedly find a word that conveys a '*camera*' sentiment. We call this model "Senti N-Gram". We can also see the terms such as '*camera*' and words that bear sentiment polarity value belong to a particular part of speech (such as a noun, adjective, verb, adverb). The following study is based on 200 data points. After doing the tokenization, the distribution of parts-of-speech has been observed. We have also used the Penn Treebank tag set for Parts of Speech (POS) tagging. A tag set is a set of part-of-speech tags used to label the parts of speech and other grammatical categories (case, tense, etc.) of each word token in a central text corpus. Below is the list of Penn Treebank tag sets (www 6).

Table 2: Penn Treebank tag set for Parts of Speech (POS) tagging

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Tag	Description
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Whdeterminer
WP	Whpronoun
WP$	Possessive whpronoun
WRB	Whadverb

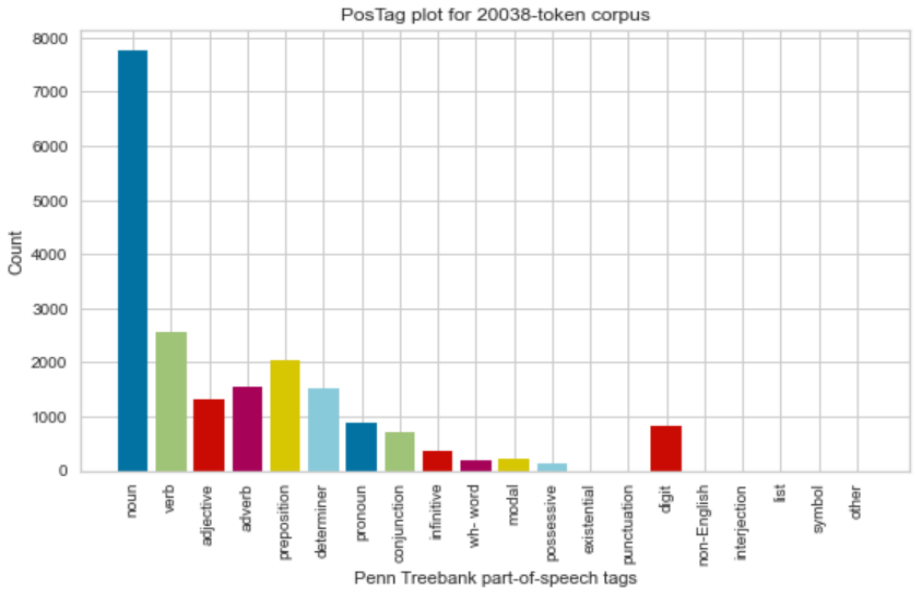

Figure 1: POS distribution of sample data

From this, we can see that Noun, Verb, Adjective, and Adverb have higher density in our main corpus.

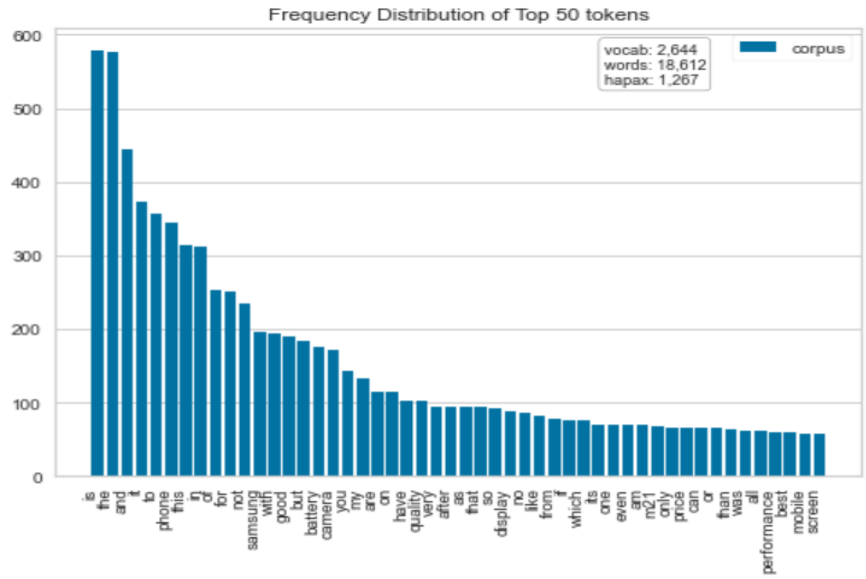

Figure 2: Frequency distribution of the top 50-word token

In the section below, we have described our proposed algorithm.

1. *From the main corpus of feedback data, do sentence tokenization.*
2. *Do word tokenization.*
3. *Do POS tagging.*
4. *Select nouns within a sentence.*
5. *Using nouns as target elements within a sentence, an n-gram model assumes a fixed window size in the target word's forward and backward direction.*
6. *By the above, pick up the forward and backward neighbour words for a fixed window size.*
7. *Consider all nouns, adverbs, adjectives, and verbs' as nodes or vertices of a graph.*
8. *Assume there is no direct relationship between the nouns (as we can consider these nouns as aspects or features of an entity; for instance, when we are looking for phone review data, camera, fingerprint, all these are aspects of the feature of a phone). They are independent. Hence there are no connections or edges between nouns (for instance, camera and fingerprint has no common edge between them).*
9. *We consider nouns as features to connect to the adverbs, adjectives, and verbs. Because the primary database is based on a product's feedback, adverbs, adjectives, and verbs bear a semantic context to nouns based on the review (subject entity). So, we can consider edges between nouns (Subject) and {adverbs, adjectives, verbs}(Description of Subject). This can be regarded as a 'knowledge graph'.*
10. *The main corpus is based on feedback data to bear positive, negative, and neutral polarity words. From this, we can say that it will bear synonymic and antonymic meaning among the words because different reviewers use different expressions, such as* '*good camera*', '*best camera*', '*bad camera*', '*worst camera*'. *Here* '*good*' *and* '*best*' *are synonyms that bear a positive sentiment; also,* '*bad*' *and* '*worst*' *are synonyms that carry a negative opinion,* '*good*' *and* '*best*' *are antonyms of* '*bad*' *and* '*worst*'. *Considering a product feature will generate positive and negative sentiment, so the spread of these words (adverbs, adjectives, and verbs) will be higher − higher possibilities of getting synonyms and antonyms.*
11. *We can relate two words (adverbs, adjectives, and verbs) based on synonyms and antonyms properties.*
12. *From 9 and 11, we can get a triangular relation (triangular graph).*
13. *The more triangles we can form with those words will be the most talked--about features.*

14. *An edge from a noun vertex to {adverbs, adjectives, verbs} will bear a weight similar to a number of occurrences of a particular noun and the adjacent {adverbs, adjectives, verbs} based on n-gram.*
15. *If W1 and W2 are the co-occurrences of that particular neighbour word within the window frame of a pivot word (Noun), then:*

$$\lambda = \max (W1, W2). \tag{2}$$

16. *Weight between any two nodes among Adverbs, Adjectives, and Verbs based on synonym and antonym property will always be 1.*
17. *We can define feature Strength as follows:*

$$(\text{Feature Strength})_i = \sum \lambda . \Delta. \tag{3}$$

Δ is the total number of triangles formed on the basis of the dictionary. A list has been created, based on some reoccurred words (adjective or adverbs or verbs) common in any review data to develop this dictionary. For this reason, an analysis has been done on review data from a different domain (such as Hotel review, Movie review, Product review), and the following dictionary has been created.

```
wordDict = {'good' : {'syn': ['good','impressive','attractive','fantastic','nice','gorgeous','great','outstanding','loved',
                              'better','amazing','best','great','satisfied', 'positive', 'liked', 'competitive','improved'],
            'any': ['bad','worse','pathetic','poor','average','wrong', 'scary', 'disgusting']},
 'suffering': {'syn': ['suffering','trouble','blaming', 'disappointed'], 'any' : ['troubleshooting']},
 'reasonable': {'syn': ['reasonable','economical','accepting','free'], 'any':['costly']},
 'clear': {'syn':['clear'], 'any': ['blurred']},
 'slow': {'syn': ['slow','running', 'low'],'any': ['stable']},
 'tough':{'syn':['tough','hard','strongest','strong'], 'any':['solved','resolved','easy']},
 'powerful': {'syn': ['powerful','huge']},
 'recommended': {'syn':['recommended','supported', 'natural', 'original'], 'any':['defective']}
}
```

Context dictionary based on Review data considering Synonym (‘syn’) and Antonym (‘any’).

Creating this dictionary aims to generate a lexicon-based database that will hold contextual meaning, both positive and negative, from the perspective of review data; for instance: ‘good’ can be associated with ‘satisfied’ or ‘improved’. This relation is based on the synonym property; all of this bears positive sentiment. Similarly, ‘disgusting’ has an antonymic relationship with ‘good’. A sentence can have multiple pivot words. Next, we find sentiment polarity of the neighbour words using a lexicon-based sentiment analyzer such as VADER. As we have said, each adjective, noun, verb, or adverb can be considered as a node inside a graph. Each node can be tagged as follows:

Word	POS	Sentiment Score

```
['macromode', 'NN', 0.0],
['camera', 'NN', 0.0],
['shoot', 'NN', -0.34],
['blurred', 'VBN', 0.0],
['images', 'NNS', 0.0],
['heating', 'NN', 0.0],
['issue', 'NN', 0.0],
['and', 'CC', 0.0],
['camera', 'NN', 0.0],
['quality', 'NN', 0.0],
```

Figure 3: Node structure of the graph with an example from the main corpus

All available nouns in a text cannot be considered as a feature. We must look for the nouns which occur the most in the entire database. We draw an edge from the noun (Probable feature) to the other words with POS adjectives, adverbs, and verbs on the basis of their occurrence in the previously mentioned window. It can be considered the edge's weight for multiple word occurrences concerning the feature noun based on the neighbour window.

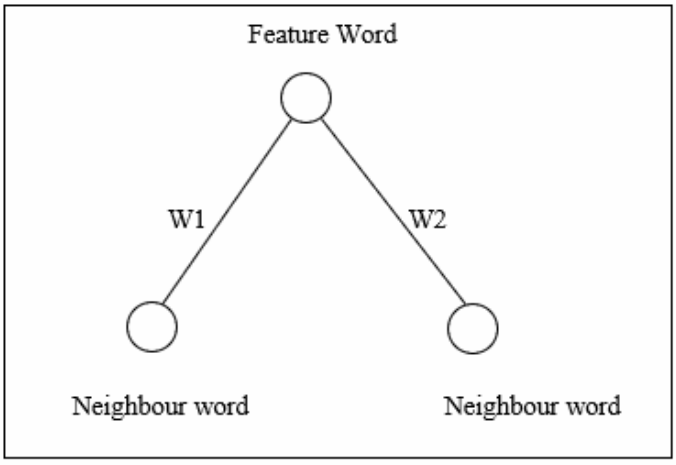

Figure 4: Initial graph structure

W1 and W2 are the co-occurrences of that particular neighbour word within the window frame of a pivot word considering all sentences within the database.

Example:

Sentence 1: '*Back camera pretty good, but front camera low light output is low*'.

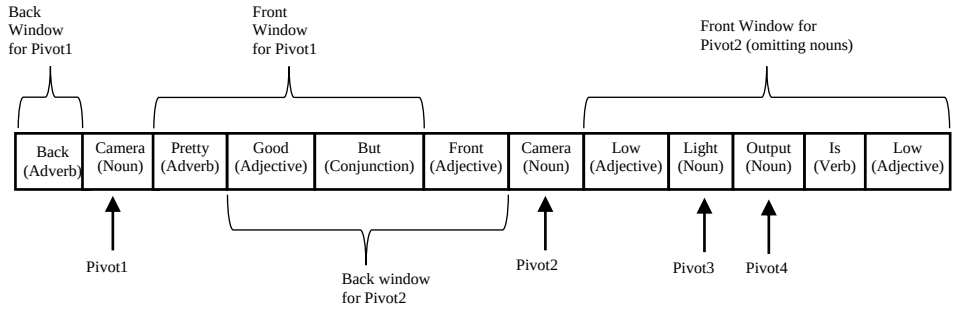

Figure 5: A review sentence structure with pivots element Noun

We consider a window of 4 from the left and right of the pivot word.

Pivot	Left of Pivot			Right of Pivot		
camera	back			pretty	good	but
camera	**good**	but	**front**	**low**	light	output
light	**front**	camera	**low**	output	is	**low**
output	camera	**low**	light	is	**low**	

Figure 6: Pivot word and sentiN-gram

Sentence 2: '*Nice rear camera and nice selfie camera but front camera struggles at night*'.

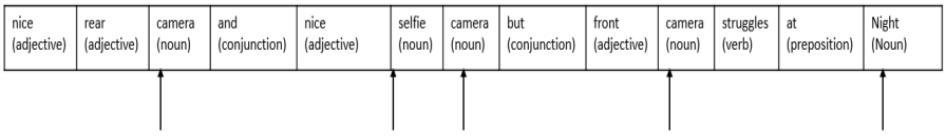

Figure 7: Pivot Nouns

Here we consider a window of 4 from the left and right of the pivot word.

We consider any noun as a feature of the product. In this case, the product is a mobile phone. We have the following nouns: [camera, light, output, selfie, night] from the two sentences above.

Pivot	Left of Pivot			Right of Pivot		
camera	**nice**	**rear**		and	nice	selfie
selfie	camera	and	**nice**	camera	but	**front**
camera	and	**nice**	selfie	camera	but	**front**
camera	camera	but	**front**	struggles	at	night
night	camera	struggles	at			

Figure 8: Pivot word and sentiN-Gram

To develop the proposed algorithm, we have used the programming language Python 3.8 on Windows 10 Home (64 bit) and different libraries to collect and extract the features. Some of the libraries used are Pandas, VADER, TextBlob, NumPy, NLTK, Spacy, Gensim, Scikit-learn, etc. The hardware used was an Intel i5 processor 2.40 GHz with 4 GB RAM.

Knowledge Graph Representation:

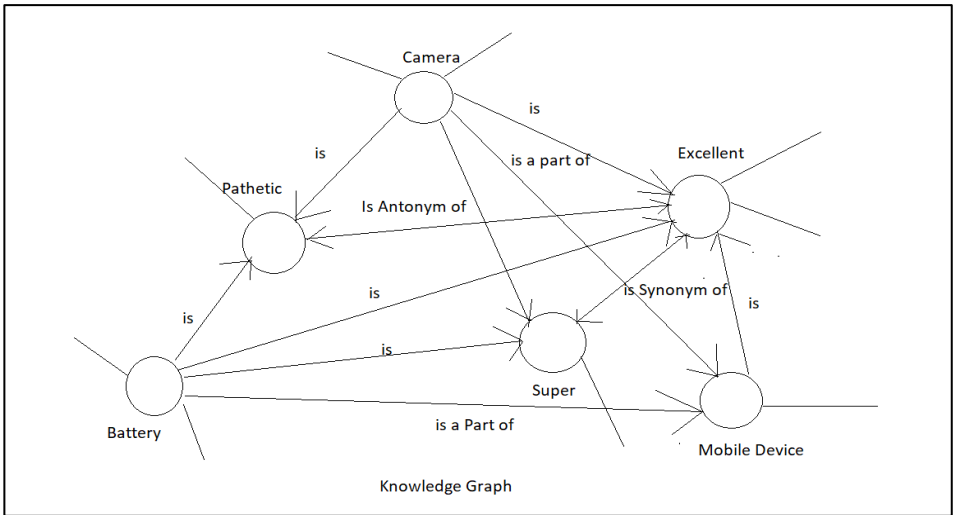

Figure 9: Relational triangle based on features (Part 1)

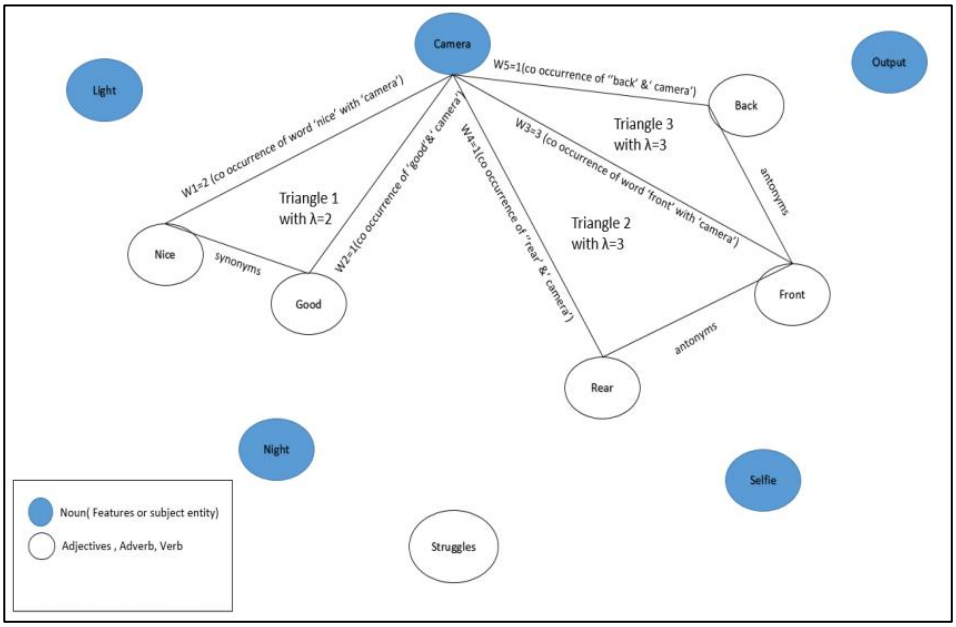

Figure 10: Relational triangle based on features (Part 2)

5 Discussion of data and result

The fundamental objective of this paper is to develop a key feature extraction algorithm. The most commonly used feature extraction algorithm in Natural Language Processing is Bag-of-Words with TF-IDF. As Ramos (2003, p. 1) said, "TF-IDF calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. Words with high TF-IDF numbers imply a strong relationship with the document they appear in, suggesting that the document could be of interest to the user if that word were to appear in a query". But the major disadvantage of this method is that the most frequent TF-IDF words of a document may not make sense while extracting key features. Words like 'this', 'if', 'the', 'or', 'what' are most frequent; they are called Stop words. Even after the elimination of these words, content-related domain-specific words with high levels of frequency, like 'communication', 'team', 'message' or 'product', etc., occur. These words do not provide any significance to the content of each review. When we try to predict the text's context using TF-IDF, the outcome is not productive. An alternative method using a graph-based approach has been widely used for Text Mining and Information Retrieval tasks (Vazirgiannis, Malliaros, Nikolentzos, 2018). These representations exploit

concepts and techniques inherited from graph theory (e.g., node centrality and subgraph frequency) to address limitations of the classical bag-of-words representation (Aggarwal, 2018). A text can be represented as a graph in numerous ways. For instance, considering all words in a text as vertices connected by a directed edge (one-way connection). Those edges can be labeled using the relation of the words in a dependency tree. Another rendering of text can use undirected edges, for example, when representing word co-occurrences. In this way, one can capture structural and semantic information of a text, mitigate the effects of the 'curse-of-dimensionality' phenomenon, identify the most critical terms of a text, and seamlessly incorporate data from external knowledge sources (Giarelis, Kanakaris and Karacapilidis, 2020). In a recent paper, Giarelis, Kanakaris and Karacapilidis (2020, p. 1) suggested that "These approaches combine statistical tests and graph algorithms to uncover hidden correlations between terms and document classes. However, while they take into account the co-occurrences between terms to identify the most representative features of a single document (something that is not the case in traditional statistical methods), they are not able to assess the importance of a term in a corpus of documents". These problems can be obliterated if we do feature extraction from a product review data with some presumption like POS tagging and considering noun as the main feature. It also adds the logic of *n*-gram, which helps us construct a knowledge graph, as we have mentioned above. The number of relation triangles helps identify the most frequent feature and can identify the most positive or negative reviewed feature. This can be found by traversing onto the side nodes of the relevant nodes (polarity wise). The efficiency is much higher. It can remove the dependency of stop word removal altogether, which is precisely the 'Curse of dimensionality'. Here a comparative study has been done to check the effectiveness of crucial feature extraction via TF-IDF over the proposed algorithm. We have a master database of probable key features. These key features are chosen by experts who have had domain knowledge of the mobile industry for more than ten years. Below is the list of most probable features talked about by customers while providing mobile-related feedback.

```
feature_dict = {
    'camera': ['camera','selfie','front','video','photo','picture','rear','macro','image','clarity','resolution','focus',
               'photography',
               'recording','zoom'],
    'display': ['display','resolution','hz','amoled','fluid'],
    'battery': ['battery','heat', 'charge', 'charged', 'capacity'],
    'charging': ['charging','slow','charger','heating','speed','power','heat','hanging','charged','heated','hot'],
    'performance': ['performance','ram','speed','lag','hanging','hang','slow'],
    'fingerprint': ['fingerprint','finger','face','touch','rear','recognition','lock'],
    'processor':['processor','slow','ram','heating','speed','lag','hang','memory'],
    'gaming': ['gaming','ram','heating','lag','heat','hanging','hot','hang','ram','slow'],
    'sensor': ['sensor','finger','face','touch','brightness','security'],
    'sound':['sound','speaker','audio','voice','dolby','clarity','volume','atmos','recording'],
    'network':['network','speed','internet','sim','signal','voice','wifi','connectivity','heating'],
    'calling':['calling','heat','sim','voice','signal','wifi','volume','connectivity']
}
```

Figure 11: Reference keyword dataset

To compare TF-IDF's behavior and the proposed algorithm with a given data, the Jaccard Similarity Coefficient (J.C.) has been introduced. It is a statistic used to understand the similarities between sample sets. The measurement focuses on the similarity between finite sample sets and is formally defined as the size of the intersection divided by the size of the sample sets' union. Its mathematical representation is:

$$J(A, B) = \frac{|A| \cap |B|}{|A| \cup |B|}, \tag{4}$$

where A and B are two finite sets (A and B don't have to be the same size).

$$J(A, A) = 1 \text{ (Similar set)}, \tag{5}$$

$$J(A, B) = 0 \text{ if } |A| \cup |B| = 0. \tag{6}$$

Now consider the most frequent 300 keywords selected using the TF-IDF algorithm and the proposed algorithm and find the Jaccard Coefficient w.r.t. the reference keyword.

J (Output from TF-IDF, Reference data) = .03125
J (Output from Proposed Algorithm, Reference data) = .1134

Hence, J (Output from Proposed Algorithm, Reference data) > J (Output from TF-IDF, Reference data). We have also compared the result with the reference database after removing stop words with the most frequent 300 words. The result is:

J (Output from TF-IDF, Reference data) = .03418
J (Output from Proposed Algorithm, Reference data) = .11009

This proves that the proposed algorithm has the edge over TF-IDF. Also, after comparing the reference data with the proposed algorithm, it has been observed that it has an accuracy of 59%, while TF-IDF has an accuracy of 17%.

If we consider the 300 most frequent features extracted by TF-IDF and our algorithm, out of 63 reference features (golden features), TF-IDF has 11 similarities. Our proposed algorithm has 37 similarities; also, we have compared the behavior with a well-known recently developed algorithm YAKE (Yet Another Keyword Extraction) (Campos, 2020). We have found that if we consider a single word extracted by YAKE, then the similarities with our respective golden dataset are 29. This further proves the superiority of our proposed algorithm.

6 Conclusion & future work

When we are extracting features from ever-growing review data to check which is/are the highest affected modules, the curse of dimensionality is the biggest challenge, since even though our thinking (reviews) about a product is alike (based on sentiment polarity, Positive, Negative, and Neutral), we express it differently. So, the extraction of keywords using Natural Language Processing becomes highly provocative. But with the proposed algorithm, we have found a way where we can reduce the effect. This algorithm consists in the merging of sentiment analysis, knowledge graph, and *n*-gram, which forms a relational triangle, and the highest occurrence of the triangle leads to feature extraction. It has been shown that this algorithm has the edge over TF-IDF. It has an accuracy of 59%, while TF-IDF has an accuracy of 17%.

While proposing our algorithm, we have not considered those sentences in which words are preceded by a negation word. In the future, we will work on that and will try to tune our algorithm accordingly. Also, this particular algorithm has been attuned on product review data for most affected modules. We will therefore try to propose a generalized model. Also, we suggest a graph--based dictionary to find out the synonym and antonym relation between words.

Declaration

1. Funding: Not Applicable
2. Conflicts of interest/Competing interests: Not Applicable
3. Availability of data and material: Available
4. Code availability: Available

References

Aggarwal C.C. (2018), *Machine Learning for Text*, Springer, Cham.

Biswas S.K., Bordoloi M., Shreya J. (2018), *A Graph-based Keyword Extraction Model Using Collective Node Weight*, Expert Systems with Applications, 97, 51-59, https://doi.org/10.1016/j.eswa.2017.12.025.

Bonatti P., Decker S., Polleres A., Presutti V. (2018), *Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web* (Dagstuhl Seminar 18371), Dagstuhl Reports, 8, 29-111.

Campolo A., Sanfilippo M., Whittaker M., Crawford K. (2018), *AI Now 2017 Report*, Symposium and Workshop, January, AI Now Institute at New York University.

Campos R., Mangaravite V., Pasquali A., Jorge A., Nunes C., Jatowt A. (2020), *YAKE! Keyword Extraction from Single Documents using Multiple Local Features*, Information Sciences, 509, 257-289, DOI: 10.1016/j.ins.2019.09.013.

Dave K., Lawrence S., Pennock D.M. (2003), *Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews*, Proceedings of the 12th International Conference on World Wide Web, 519-528.

Devika R., Subramaniyaswamy V. (2019), *A Semantic Graph-based Keyword Extraction Model Using a Ranking Method on Big Social Data*, Wireless Netw, https://doi.org/10.1007/s11276-019-02128-x.

Feldman R., Dagan I. (1995), *Knowledge Discovery in Textual Databases* (KDT), Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, August 20-21, AAAI Press, 112-117.

Giarelis N., Kanakaris N., Karacapilidis N. (2020), *An Innovative Graph-Based Approach to Advance Feature Selection from Multiple Textual Documents*, Artificial Intelligence Applications and Innovations, 583, May 6, 96-106, DOI: 10.1007/978-3-030-49161-1_9.

Houari M., Rhanoui M., Asri B. (2015), *From Big Data to Big Knowledge: The Art of Making Big Data Alive*, 1-6, DOI: 10.1109/CloudTech.2015.7337001.

Htay S.S., Lynn K.T. (2013), *Extracting Product Features and Opinion Words Using Pattern Knowledge in Customer Reviews*, The Scientific World Journal, Vol. 2013, Article ID 394758, 5 pages, https://doi.org/10.1155/2013/394758.

Hulth A. (2003a), *Improved Automatic Keyword Extraction Given More Linguistic Knowledge*, EMNLP, 216-223.

Hulth A. (2003b), *Reducing False Positives by Expert Combination in Automatic Keyword Indexing*, RANLP, 367-376.

Jaideepsinh K., Saini J. (2016), *Stop-Word Removal Algorithm and Its Implementation for the Sanskrit Language*, International Journal of Computer Applications, 150, 15-17, DOI: 10.5120/ijca2016911462.

Jia Y., Qui Y., Shang H., Jiang R., Li A. (2018), *A Practical Approach to Constructing a Knowledge Graph for Cybersecurity*, Engineering, 4(1), 53-60, https://doi.org/10.1016/j.eng.2018.01.004.

Jiang X., Hu Y., Li H. (2009), *A Ranking Approach to Keyphrase Extraction*, SIGIR, 756-757.

K-CAP '19 (2019), Proceedings of the 10th International Conference on Knowledge Capture, September, 131-138, https://doi.org/10.1145/3360901.3364441.

Kim K., Hur Y., Kim G., Lim H. (2020), *GREG: A Global Level Relation Extraction with Knowledge Graph Embedding*, Applied Sciences, 10, 1181.

LeCun Y., Bengio Y., Hinton G. (2015), *Deep Learning*, Nature, 521, 436-44, https://doi.org/10.1038/nature14539.

Liu B. (2009), *Handbook Chapter: Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing*, Marcel Dekker, Inc., New York, NY, USA.

Manrique R., Pereira B., Mariño O. (2019), *Exploring Knowledge Graphs for the Identification of Concept Prerequisites*, Smart Learning Environments, 6, 21, https://doi.org/10.1186/s40561-019-0104-3.

Markov A., Last M., Kandel A. (2007), *Fast Categorization of Web Documents Represented by Graphs*, Advances in Web Mining and Web Usage Analysis, 4811, 56-71.

Park D.-H., Kim S. (2008), *The Effects of Consumer Knowledge on Message Processing of Electronic Word-of-mouth via Online Consumer Reviews*, Electronic Commerce Research and Applications, 7, 399-410.

Ramos J. (2003), *Using TF-IDF to Determine Word Relevance in Document Queries*, Computer Science, Proceedings of the First Instructional Conference on Machine Learning, 1-4.

Rose S., Engel D., Cramer N., Cowley W. (2010), *Automatic Keyword Extraction from Individual Documents*, DOI: 10.1002/9780470689646.ch1.

Russell S.J., Norvig P. (2003), *Artificial Intelligence − A Modern Approach: The Intelligent Agent Book*, Prentice-Hall.

SAC '07 (2007), Proceedings of the 2007 ACM Symposium on Applied Computing, March, 807-811, https://doi.org/10.1145/1244002.1244182.

Safrin R., Sharmila K.R., Shri Subangi T.S., Vimal E.A. (2017), *Sentiment Analysis on Online Product Review*, International Research Journal of Engineering and Technology (IRJET), 4, April, 2381-2388.

Sammons M., Christodoulopoulos C., Kordjamshidi P., Khashabi D., Srikumar V., Vijayakumar P., Bokhari M., Wu X., Roth D. (2016), *Edison: Feature Extraction for NLP, Simplified* [in:] N. Calzolari, K. Choukri, H. Mazo, A. Moreno, T. Declerck, S. Goggi, M. Grobelnik, J. Odijk, S. Piperidis, B. Maegaard, J. Mariani (eds.), Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016, European Language Resources Association (ELRA), 4085-4092.

Shi W., Zheng W., Yu J.X., Cheng H., Zou L. (2017), *Keyphrase Extraction Using Knowledge Graphs*, Data Science Engineering, 2, 275288, https://doi.org/10.1007/s41019-017-0055-z.

Sidorov G., Velasquez F., Stamatatos E., Gelbukh A., Chanona-Hernández L. (2013), *Syntactic Dependency-Based N-grams as Classification Features* [in:] I. Batyrshin, M.G. Mendoza (eds.), Advances in Computational Intelligence, MICAI 2012, Lecture Notes in Computer Science, 7630, Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-37798-3_1.

Turney P.D. (2002), *Learning to Extract Keyphrases from the Text*, CoRR, cs. L.G./0212013.

Vazirgiannis M., Malliaros F., Nikolentzos G. (2018), *GraphRep: Boosting Text Mining, NLP, and Information Retrieval with Graphs*, Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2295-2296.

Wang Ch., Ma X., Chen J., Chen J. (2018), *Information Extraction and Knowledge Graph Construction from Geoscience Literature*, Computers & Geosciences, 112, 112-120, https://doi.org/10.1016/j.cageo.2017.12.007.

Wang Q., Mao Z., Wang B., Guo L. (2017), *Knowledge Graph Embedding: A Survey of Approaches and Applications*, IEEE Transactions on Knowledge and Data Engineering, 29(12), December 1, 2724-2743, DOI: 10.1109/TKDE.2017.2754499.

Wang W., Do D.B., Lin X. (2005), *Term Graph Model for Text Classification*, Advanced Data Mining and Applications, 19-30.

Willemsen L.M., Neijens P.C., Bronner F., de Ridder J.A. (2011), *"Highly Recommended!" The Content Characteristics and Perceived Usefulness of Online Consumer Reviews*, Journal of Computer-Mediated Communication, 17(1), October 1, 19-38, https://doi.org/10.1111/j.1083-6101.2011.01551.x.

Witten I.H., Paynter G.W., Frank E., Gutwin C., Nevill-Manning C.G. (1999), *KEA: Practical Automatic Keyphrase Extraction*, Proceedings of the Fourth ACM Conference on Digital Libraries, 254-255.

Xu J., Kim S., Song M., Jeong M., Kim D., Kang J., Rousseau J.F., Li X., Xu W., Torvik V.I., Bu Y., Chen Ch., Ebeid I.A., Li D., Ding Y. (2020), *Building a PubMed Knowledge Graph*, Scientific Data, 7, 205, https://doi.org/10.1038/s41597-020-0543-2.

Zhao H., Pan Y., Yang F. (2020), *Research on Information Extraction of Technical Documents and Construction of Domain Knowledge Graph*, IEEE Access, 8, 168087-168098, DOI: 10.1109/ACCESS.2020.3024070.

Zhao J., Wang T., Yatskar M., Ordonez V., Chang K.W. (2017), *Men also Like Shopping: Reducing Gender Bias Amplification Using Corpus-level Constraints*, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2979-2989.

(www 1) https://www.merriam-webster.com/dictionary/adjective (accessed: 1.11.2020).

(www 2) Ji S., Pan S., Cambria E., Marttinen P., Yuar P.S. (2021), *A Survey on Knowledge Graphs: Representation, Acquisition and Applications*, IEEE Transactions on Neural Networks and Learning Systems, Xiv:2002.00388 (accessed: 8.11.2020).

(www 3) Mäntylä M.V., Graziotin D., Kuutila M. (2018), *The Evolution of Sentiment Analysis − A Review of Research Topics, Venues, and Top Cited Papers*, Computer Science Review, 27, February, 16-32, arXiv:1612.01556 [cs.CL] (accessed: 10.11.2020).

(www 4) http://web.onda.com.br/abveiga/capitulo4-ingles.pdf (accessed: 11.11.2020).

(www 5) Mutlu E.C., Oghaz T.A., Rajabi A., Garibay I., *Review on Learning and Extracting Graph Features for Link Prediction*, arXiv:1901.03425 (accessed: 11.11.2020).

(www 6) https://www.sketchengine.eu/penn-treebank-tagset/#:~:text=English%20Penn%20Treebank%20part%2Dof%2Dspeech%20Tagset&text=Atagset%20is%20a%20list%20of,(case%2C%20tense%20etc.) (accessed: 12.11.2020).

(www 7) Hellström T., Dignum V., Bensch S. (2020), *Bias in Machine Learning – What Is It Good for?* https://arxiv.org/pdf/2004.00686.pdf (accessed: 12.11.2020).

(www 8) https://www.lexico.com/definition/noun (accessed: 9.11.2020).