Computational Grounded Theory

An Experiment: Human versus Machine

Clint Wolfs
Lectoraat Future-Proof Financial
Zuyd Hogeschool
Sittard, Netherlands
Clint.Wolfs@zuyd.nl

Eric Mantelaers
Lectoraat Future-Proof Financial
Zuyd Hogeschool
Sittard, Netherlands
Eric.Mantelaers@zuyd.nl

Martijn Zoet
Lectoraat Future-Proof Financial
Zuyd Hogeschool
Sittard, Netherlands
Martijn.Zoet@zuyd.nl

Rick Reijnders
Lectoraat Future-Proof Financial
Zuyd Hogeschool
Sittard, Netherlands
Rick.Reijnders@zuyd.nl

***Abstract*— Grounded theory has been a fundamental concept within qualitative research for decades. While human creativity forms an important element during the creation of new theories, there have been suggestions in which computers might support this creative process. As a result, the computational grounded theory framework was introduced. Currently, there is a lack of studies that evaluate practical performance implications of computational grounded theory approaches. This paper aims to contribute by evaluating the differences between a manual and an automated keyword extraction process; a process that is considered to be important during the first stage of the open coding process. Results indicate that the outcomes of the automated process are - to some extent - in line with the outcomes of the manual process. Nonetheless, phi coefficients do not exceed 0.21, meaning that the results do not perfectly agree with each other. As a result, some keywords might be left out while other unimportant words may be labeled as being a keyword. Therefore, although automatic keyword extractors can be helpful during the open coding process, results should still be cautiously interpreted. Moreover, the results indicate that elements of the computational grounded theory framework can be implemented in practice, without significant different results.**

Keywords*— *Computational grounded theory; automatic keyword extractors; qualitative research; theory development; coding process; validity; reliability; RAKE; PRE; SRE; TRE; MRE; Yake!; KBERT

I. INTRODUCTION

Content analysis is an established method in scientific research. One of the main challenges with content analysis in general, and hand-coding techniques in particular, is the resources (in terms of cost and time) associated with the data collection and data analysis. Therefore, one can question the scope and depth of textual data analysis. As a result, researchers have been investigating ways to minimize resources while maintaining reliability, validity and reproducibility. In addition, a more efficient process enables the researchers to collect and analyze more (diverse) data sources to begin with.

Two fields that have changed content analysis (and continue to do so) are (1) information science and (2) computational linguistics [1]-[6]. Both apply supervised machine learning, as well as unsupervised machine learning to text analysis. This has led to a debate on how to incorporate such methods in existing research processes and methods without compromising scientific integrity. More specifically, it has led to the question how computational linguistics can result in higher reliability, validity and reproducibility of the results.

Nelson [7] proposes a methodological framework called computational grounded theory which consists of three steps: 1) pattern detection using unsupervised methods, 2) pattern refinement using guided deep reading and 3) pattern confirmation using natural language processing. These steps consist of techniques that support the traditional coding process within grounded theory: 1) open coding, 2) axial coding and 3) selective coding. Often, these techniques are tested by comparing and evaluating the information retrieval of specific algorithms. However, for practical application, a comparison to the results of human coding is preferred [9]. Nonetheless, Nelson [7] states that human comparison has not been performed very often.

This paper aims to extend the understanding of the application of unsupervised methods for open coding. While we in line with previous research consider multiple unsupervised techniques, we compare these techniques to the results of human coding and treat this coding as our benchmark. With these premises, the specific research

 ISBN: 978-1-68558-082-7

question addressed is: "How do the results of unsupervised methods compare to human coded results?"

The remainder of this paper is organized as follows. First, section 2 discusses the literature review which is followed by the explanation of the research method in section 3. Section 4 describes the data collection and the results are presented in section 5. Section 6 the conclusions and corresponding discussion. Lastly, limitations and propositions for future research are presented in section 7.

II. LITERATURE REVIEW

As previously mentioned in the introduction, [7] propose a methodological framework in which computers might assist during the traditional process of grounded theory. The automated process of keyword extraction can be a practical interpretation of computer assisted grounded theory.

A. (Computational) Grounded Theory

Grounded theory is considered to be a fundamental concept within qualitative research. In contrast to the research often conducted within the quantitative field, grounded theory does not seek to prove or disprove theories that remain to be untested [9]. Rather, the aim of grounded theory is to construct theories [9] that can be tested using traditional quantitative research methods. Grounded theory consists of three main phases, being open coding, axial coding and selective coding [10]. During the open coding process, key words or key phrases that are believed to be related to some phenomenon are extracted from the qualitative data [11]. Through systematic analysis and constant comparison of the coded data, the relationships between phenomena can then be investigated during the axial coding process [11]. Thereby, overarching categories are created. Lastly, one single core category that overarches multiple of the underlying categories is created during the selective coding process [10].

Despite of the proposed methodological framework of [7], the coding process often remains a manual process. In addition to the relatively high labor intensity of this manual process and the subjectivity across coders, there are also plausible limitations in terms of reproducibility [7]. Inconsistencies within coded elements from individual coders could lead to suboptimal and inaccurate results. As a result, independent coders (try to) follow guidelines and be as consistent as possible during the coding process [12]. Additionally, a retrospective assessment of the quality of the coding process is considered to be very important [13]. In its most simplistic form, the reliability of multiple coders can be assessed by computing the percentage of agreement. However, it is argued that this (relatively simple) measure can be misleading since it does not take coincidence into account [14]. As a result, Krippendorff proposed a more conservative method to determine the reliability by taking random chance into account [14]. Nonetheless, while these measures can be helpful, they are examples of repressive measures and checks. If these measures lead to the conclusion that the coding process is inconsistent, the labor-intensive coding process has to be redone in order to prevent unsubstantiated theory development. It would be more useful if inconsistencies can be minimized to begin with. A certain type of automation might form a plausible preventive measure.

B. Automatic Keyword Extraction

As previously mentioned, key words are selected at the beginning of the open coding process. Since these selected keywords form the foundation of the grounded theory process, it is important that these keywords are the result of a consistent process. By using a machine instead of a human, inconsistencies might be minimized. The selection of keywords is a process that could be done automatically in a variety of different manners. Automatic keyword extraction is the process in which an algorithm identifies the keywords within a collection of texts (corpus). These keywords should represent the most useful information within the corpus [15]. With the manual open coding process in mind, automatic keyword extraction algorithms could not only simplify this labor-intensive process but could also establish more consistent results. As of now, there are numerous algorithms available that each has its own approach in determining whether or not a word is a keyword [15].

C. Types of Automtic Keyword Extractors

Similar to manual coding, it is possible to use multiple estimators and aggregate their decision. Within this study, there are seven independent algorithms that will be used to estimate whether or not a word is a keyword**:** Rapid Keyword Extraction (RAKE), Position Rank Extractor (PRE), Single Rank Extractor (SRE), Topic Rank Extractor (TRE), Multipartite Rank Extractor (MRE), Yet Another Keyword Extractor (Yake!) and Key Bidirectional Encoder Representations from Transformers (KBERT).

RAKE assumes that key phrases usually occur in the beginning of a text corpus [16]. Because of this assumption, one important parameter is the phrase delimiter (',' and '.' for example) which is used to create so called 'candidate experessions'. These candidate expressions are part of a sentence/text corpus. Moreover, a second important parameter is a list with stopwords. This list is used to 1) remove irrelevant words from the tokenized corpus and 2) split the corpus to create the candidate expressions. The final score is calculated using both the words (exluding stopwords) and the candidate expressions. In addition, RAKE differentiates itself from comparable algorithms due to its simplicity [17], computational efficiency, speed and the ability to work on individual documents [16] Nonetheless, the plausible lack of stopwords (which is a parameter) might influence the output, resulting in less relevant results [16].

PRE is a graph based approach in which a vertex represents a token and an edge represent a relationship

 ISBN: 978-1-68558-082-7

between vertices [15]. For each individual word, PRE establishes a graph [18]. Moreover, PRE considers (in addition to word position) also the word frequency [16]. Based on this information, words that occur relatively often and early within the corpora, receive a greater probability of being a keyword [16]. This means that the assumption of RAKE could also be applicable to PRE [16]. In terms of performance, PRE seems to perform better compared to the TextRank alternative**.**

SRE generates a graph for each document based on the words in that document. Moreover, it computes the corresponding word scores that drive the decision on whether or not a word is considered to be a keyword [19].

Similar to SRE and PRE, TRE is also a graph-based approach. However, it tries to achieve better performance by assuming that each document relates to a specific topic. Indeed, the addition of this assumption generally leads to a better performance in terms of the precision and recall evaluation measures, compared to TRE [20].

MRE is built upon the foundation of TRE and therefore has similar assumptions. However, whereas TRE simply tries to find relationships between words based on different topics, MRE also tries to differentiate the importance of the relationships between words within those topics [21]. Results indicate that this approach leads to better performance, compared to SRE, PRE and TRE [18].

Whereas PRE seemed to perform better compared to TextRank, Yake! seems to perform better than RAKE, TextRank and SRE. Comparable to most algorithms, Yake! starts with tokenizing the text corpus based on specified delimiters. Based on this list of words, five features are extracted: casing (does the word start with a capital letter, excluding the words at the start of a sentence), word position, word frequency, relatedness to context and the proportion of sentences that include the word. Due to the word position feature, the assumption that more relevant words occur in the beginning of a text corpus is (again) applicable. The five features are then aggregated into one number which will then be used to determine a final score [22].

KBERT originates from the Bidirectional Encoder Representations from Transformers (BERT) algorithm which can be used for the creation of word embeddings [23]. The input for the BERT algorithm includes three main elements: token, segment and position [24]. Therefore, BERT differentiates itself from most other word embedding architectures that merely use word vectors as input [24]. Initial performance results of a (fine-tuned) BERT classification model seem to be high with an accuracy of 97.6% according to a recent study [24].

As previously mentioned, most literature focusses on performance comparison between algorithms [7] while a comparison to the results of human coding is preferred [8]. Therefore, this paper aims to evaluate plausible differences between manual and automated keyword extraction. In the end, while automated results might be more reliable, it does not mean that the results are valid. A comparison with a manual process is in this context the only way to also take validity into consideration. Therefore, the following hypothesis will be tested:

H1.) There is no significant association between the results of the automated and manual text coding process.

III. RESEARCH METHOD

The goal of this study is to evaluate plausible differences between manual and automated text coding. More specifically, this paper aims to identify the differences between the automated and manual keyword extraction. While the consistency of the automated process will be higher, it does not mean that the algorithms identify the correct key words to begin with. Therefore, it is important to compare the automated results to the results of human coding. Texts will be assessed by the seven independent algorithms and two individual researchers.

A. Keyword Extraction

Regarding the automated keyword extraction, seven algorithms will be used to identify the most unique and relevant words within the text corpus (keywords). For each text corpus, the results of these independent algorithms will be compared to each other.

With regard to the manual keyword extraction process, two researchers will be selecting the most unique and relevant words from the same text corpus. For reliability concerns, the results of both researchers will be tested for consistency using the inter rater reliability coding method. In the situation of a disagreement, both researchers will directly discuss and adjust coding accordingly.

B. Comparison

In order to meet the primary objective of this study, the results of the manual and automated coding process need to be compared. This comparison will be made on two levels. First, the results of the manual coding procedure are compared to the results of each individual algorithm. In addition, the results of the manual coding process will be compared to an aggregated result. More specifically, if at least five out of seven algorithms identify a word as being a keyword, we conclude that the general automated approach identifies the word as a keyword.

In addition to descriptive statistics, differences between the categories will be tested on significance and effect size. While significance will be tested by a chi-square test, effect size will be determined by both a phi coefficient and an odds-ratio respectively.

IV. DATA COLLECTION

The data that will be used for this study, is formed by a collection of titles of news articles and online blogs. These data have been collected by the research group Future-Proof Financial of Zuyd Hogeschool. Over a period of thirteen months (January 10th 2021 - February 4th 2022), the data have been collected. Because the news articles and online blogs are collected from websites of accounting firms, most

news articles are related to accounting. On a daily basis, the URLs of articles and blogs have been automatically collected via the use of web scrapers. Using the URLs of all the blogs and articles, the titles can be extracted. The selection of accounting firms is based on a verified registration of accounting firms that is maintained by the Dutch government.

The final data table consisted of 29.672 rows that each represents an URL to an article or blog that was posted by one of the 181 sources (websites). Due to duplicate titles, 177 sources remain of which 19.209 URLs have been collected and will be taken into consideration during the analyses.

V. RESULTS

During the analysis, the 177 sources resulted in 19.209 titles and thus unique URLs. Moreover, all the titles combined consisted of 213.127 words which, on average, counted 6.28 characters. Figure 1 shows the relative distribution frequency for the number of characters in the words. While inspecting Figure 1, it is important to note that the Dutch language does not include spaces in word compositions. For example, 'small-scale investment deduction' is written as one single word: 'kleinschaligheidsinvesteringsaftrek'. Furthermore, the 213.127 words and 19.209 titles resulted in an average of 11.1 words for each title (i.e., text corpus). Figure 2 provides the relative distribution frequency for the number of words in the corpus. Since some publishers chose to use a brief introduction as title section, the distribution is severely right-skewed. Lastly, out of these 213.127 words, only 15.779 words were found to be unique throughout the whole data set.

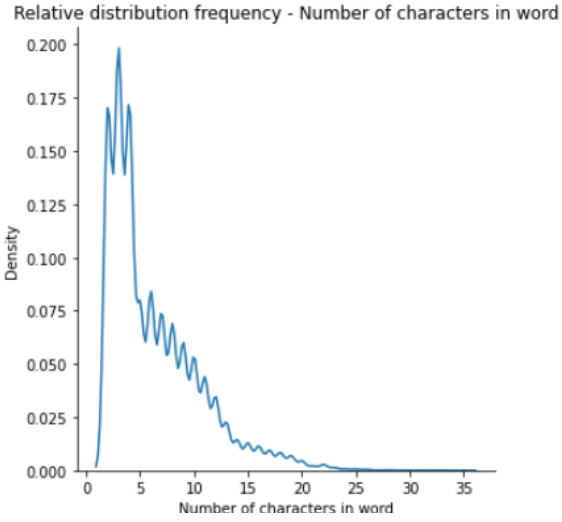

Figure 1. Relate distribution frequency of number of characters in words

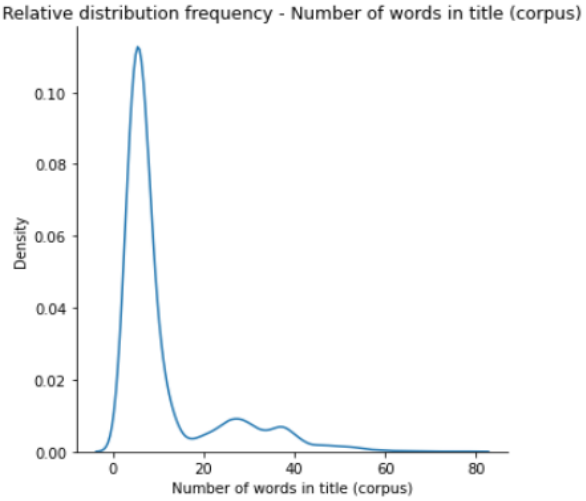

Figure 2. Relative distribution frequency of number of words in corpus

With regards to the statistical tests, all results turned out to be highly significant with chi-square values that range between +/- 3.000 up to +/- 10.000. This would imply that the expected frequencies differ significantly from the observed frequencies, meaning that the algorithms either significantly agree or disagree with the manual results. With phi coefficients ranging between 0.12 and 0.21, we can conclude there is not an extraordinary high or low association. Nonetheless, the positive coefficients indicate that the algorithms significantly agree with the manual results. A minimum of 2.49 and a maximum of 4.34 for the odds ratios confirm that it is more likely that the algorithms do not indicate the word as a keyword, given that the manual

TABLE I. STATISTICAL RESULTS - AUTOMATED V.S. MANUAL RESULTS

Comparison	Chi-square	p-value	degrees of freedom	n	phi coefficient	odds ratio
Aggregated assessment vs. Manual assessment	7.798.193	0.0	1	213127	0.191	4.019
RAKE vs. Manual assessment	2.984.758	0.0	1	213127	0.118	2.486
YAKE vs. Manual assessment	8.836.122	0.0	1	213127	0.205	4.335
PRE vs. Manual assessment	9.847.433	0.0	1	213127	0.215	4.206
SRE vs. Manual assessment	6.961.978	0.0	1	213127	0.181	3.633
MRE vs. Manual assessment	6.672.700	0.0	1	213127	0.177	3.475
TRE vs. Manual assessment	6.293.304	0.0	1	213127	0.172	3.379
KBERT vs. Manual assessment	6.564.973	0.0	1	213127	0.176	3.433

 ISBN: 978-1-68558-082-7

process did not indicate it as a keyword either. For example, the odds ratio of 4.34 indicates that it is 4.34 times more likely that Yake! does not indicate a word as being a keyword, given that the manual process did not indicate the word as a keyword either. Interesting to mention is that Yake! ($X2$ (1, 213,127) = 8,836.12, $p < 0.01$, $\varphi = 0.204$) and PRE ($X2$ (1, 213,127) = 9,847.43, $p < 0.01$, $\varphi = 0.215$) turned out to have the highest phi coefficient, while both are less computationally intensive compared to KBERT. Moreover, Yake! and PRE are also the only algorithms that – in terms of effect sizes - outperform the aggregated assessment ($X2$ (1, 213,127) = 7,798.19, $p < 0.01$, $\varphi = 0.191$) where at least 5 out of 7 algorithms have to agree before indicating it as a keyword. Table I shows the contingency table for the comparison between the manual results and the automated, aggregated assessment. Table II provides the results for each individual algorithm and the aggregated assessment. Table II shows that, regardless of the algorithm used, the results of the automated process are significantly associated with the results of the manual process. More specifically, the table shows only positive phi coefficients, meaning that manual and automated results are significantly in line with each other. This implies that we can reject the hypothesis stated above.

TABLE II. CONTINGENCY TABLE – AUTOMATED AGGREGATED RESULTS | MANUAL RESULTS

	Aggregated assessment | Yes	**Aggregated assessment | No**
Manual assessment | Yes	6,548 / 3.1%	13,830 / 6.5%
Manual assessment | No	20,316 / 9.5%	172,433 / 80,9%

VI. DISCUSSION AND LIMITATIONS

Even though initial results seem promising, there are also several limitations to take into account while interpreting these results and corresponding conclusions. First of all, the data are related to one single area of expertise. While this eases the process of selecting coders for the manual text coding process, it also limits the degree to which the conclusions should be taken into consideration. It might be that results are optimal within the financial/accounting expertise but not so in the medical field. Moreover, only Dutch articles have been covered by the text coding process. This might limit the representativeness of the results. Most important reason is that most keyword extraction algorithms rely (to some extent) on the position of words. As a result, the algorithms might become inaccurate if a certain language relies on a different structure. Lastly, while the titles were often completely written in the Dutch language, there were instances in which a title also used English terms. This might have limited the accuracy of our results since most algorithms require defining the language of the text corpus.

VII. CONCLUSION AND FUTURE WORK

While the theoretical framework of computational grounded theory has been published several years ago, it seems that practical applications are mostly purely within the algorithmic field. As a result, a comparison between the performance of algorithms and the performance of the manual process is often left out. Moreover, while algorithms form an application of automation and therefore deliver more reliable results, it does not mean that algorithms also deliver valid results. By comparing manual and automated results, this study attempted to apply one single element of computation grounded theory in practice, outside of the purely algorithmic field. The effect sizes imply that, while the results of the manual and automated process are significantly associated, the phi coefficients are not necessarily extraordinary high or low. Nonetheless, no algorithm was found to be negatively associated with the results of the manual process, indicating that the manual and automated results are more often in line with each other than that they are not. This indicates that validity – to some extent – is warranted. As a result, automatic keyword extractors can be a helpful technique during the open coding process.

By automating the identification of important words that are labeled in the next stage of the coding process, the consistency across manual coders might be improved. Moreover, it seems to be plausible that the use of automatic keyword extractors leads to a less resource intensive process. Nonetheless, automatic keyword extractors should be used cautiously since it is likely that there still are false positives (found keywords that are not necessarily important) and false negatives (important words that are not found by the algorithm). This might have severe consequences for the next stages of the coding process and therefore, severe consequences for the theory development as a whole.

With regards to future work, limitations that are stated in the previous section could be taken into consideration. In addition to these limitations, this study solely focuses on one single part of the proposed computational grounded theory framework. It would be useful to investigate and compare machine and human performance with regards to other individual elements of the computational grounded theory framework. Lastly, it would be interesting to evaluate machine and human performance with regards to the computational grounded theory framework as a whole.

REFERENCES

[1] C. A. Bail, “Inside the Rituals of Social Science,” Theory and Society, vol. 43, no. 3–4, pp. 465–482, Jul. 2014, doi: https://doi.org/10.1007/s11186-014-9216-5.

[2] R. Biernacki, “The cultural environment: measuring culture with big data,” Reinventing Evidence in Social Inquiry, pp. 1–26, 2012, doi: https://doi.org/10.1057/9781137007285_1.

[3] P. DiMaggio, M. Nag, and D. Blei, “Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of U.S. government arts funding,” Poetics, vol. 41, no. 6, pp. 570–

606, Dec. 2013, doi: https://doi.org/10.1016/j.poetic.2013.08.004.

[4] J. W. Mohr and P. Bogdanov, “Introduction—Topic models: What they are and why they matter,” Poetics, vol. 41, no. 6, pp. 545–569, Dec. 2013, doi: https://doi.org/10.1016/j.poetic.2013.10.001.

[5] P. A. Reed and J. E. LaPorte, “A content analysis of AIAA/ITEA/ITEEA conference special interest sessions: 1978-2014,” Journal of Technology Education, vol. 26, no. 3, Jul. 2015, doi: https://doi.org/10.21061/jte.v26i3.a.2.

[6] K. M. Meyer and T. Tang, “#SocialJournalism: Local news media on twitter,” International Journal on Media Management, vol. 17, no. 4, pp. 241–257, Oct. 2015, doi: https://doi.org/10.1080/14241277.2015.1107569.

[7] L. K. Nelson, “Computational grounded theory: A methodological framework,” Sociological Methods & Research, vol. 49, no. 1, pp. 3–42, Dec. 2020, doi: https://doi.org/10.1177/0049124117729703.

[8] K. Benoit, M. Laver, and S. Mikhaylov, “Treating words as data with error: Uncertainty in text statements of policy positions,” American Journal of Political Science, vol. 53, no. 2, pp. 495–513, Dec. 2009, doi: https://doi.org/10.1111/j.1540-5907.2009.00383.x.

[9] J. Mills, A. Bonner, and K. Francis, “The development of constructivist grounded theory,” International Journal of Qualitative Methods, vol. 5, no. 1, pp. 25–35, Mar. 2006, doi: https://doi.org/10.1177/160940690600500103.

[10] A. Moghaddam. “Coding issues in grounded theory,” Issues In Educational Research, vol. 16, no. 1, pp. 52–66, Apr. 2006.

[11] C. Goulding, "Grounded Theory: some reflections on paradigm, procedures and misconceptions," working paper, University of Wolverhampton., Telford, UK, 1999 [Online]. Available: https://wlv.openrepository.com/bitstream/handle/2436/11403/Goulding.pdf?sequence=1&isAllowed=y

[12] J. Nassar, Viveca Pavon-Harr, M. Bosch, and I. McCulloh, “Assessing data quality of annotations with krippendorff alpha for applications in computer vision,” Dec. 2019.

[13] K. Krippendorff, " Computing Krippendorff's Alpha-Reliability," working paper, University of Pennsylvania., Philadelphia, PA, USA, 2011 [Online]. Available: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1043&context=asc_papers

[14] M. Zoet, J. Versendaal, P. Ravesteyn, and R. Welke, ‘Alignment of business process management and business rules’, 2011.

[15] S. Beliga, A. Meštrović, and S. Martinčić-Ipšić, “An overview of graph-based keyword extraction methods and approaches,” Journal of Information and Organizational Sciences, vol. 39, no. 1, pp. 1–20, 2015.

[16] M. G. Thushara, Tadi Mownika, and Ritika Mangamuru, “A comparative study on different keyword extraction algorithms,” in 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Dec. 2019, pp. 969–973. doi: https://doi.org/10.1109/ICCMC.2019.8819630.

[17] S. Anjali, N. M. Meera, and M. G. Thushara, “A graph based approach for keyword extraction from documents,” in 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Dec. 2019, pp. 1–4. doi: https://doi.org/10.1109/ICACCP.2019.8882946.

[18] M. Ravikiran, “Finding black cat in a coal cellar – keyphrase extraction & keyphrase-rubric relationship classification from complex assignments,” Dec. 2020.

[19] X. Wan and J. Xiao, “Single document keyphrase extraction using neighborhood knowledge.,” in AAAI, 2008, vol. 8, pp. 855–860.

[20] A. Bougouin, F. Boudin, and Béatrice Daille, “TopicRank: Graph-based topic ranking for keyphrase extraction,” in International Joint Conference on Natural Language Processing (IJCNLP), Dec. 2013, pp. 543–551. Available: https://hal.archives-ouvertes.fr/hal-00917969

[21] F. Boudin, “Unsupervised keyphrase extraction with multipartite graphs,” in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 667–672. doi: https://doi.org/10.18653/v1/N18-2105.

[22] R. Campos, Vítor Mangaravite, A. Pasquali, Alípio Mário Jorge, C. Nunes, and A. Jatowt, “YAKE! Collection-independent automatic keyword extractor,” pp. 806–810, 2018, doi: https://doi.org/10.1007/978-3-319-76941-7_80.

[23] Y. Wang, L. Cui, and Y. Zhang, “How can BERT help lexical semantics tasks?,” Dec. 2019.

[24] M. Tang, P. Gandhi, Md Ahsanul Kabir, C. Zou, J. Blakey, and X. Luo, “Progress notes classification and keyword extraction using attention-based deep learning models with BERT,” Dec. 2019.

 ISBN: 978-1-68558-082-7