A Survey on Automatic Text Summarization Techniques

S Hima Bindu Sri¹, Sushma Rani Dutta²

¹Assistant Professor, ²Assistant Professor
Department of Computer Science and Engineering, Koneru Lakshmaiah University,
Department of Computer Science and Engineering, Koneru Lakshmaiah University.

E-mail: sushamadutta@klh.edu.in

Abstract. In recent years, there is a tremendous explosion in the amount of text data on the internet and in the archives of news articles, scientific papers, legal documents and even in online product reviews. Text summarization is playing an important role in automatic content creation, minutes of meeting generation, helping disabled people and also for quick online document reading. To achieve these, several automation techniques have been proposed in various researches. In this regard, performing an exclusive survey on different methods, approaches of automatic text summarization which are published in different articles in most recent three years.

Key words: Text summarization, Supervised, Unsupervised, Optimization, Natural Language Processing.

1. Introduction

The aim of the text summarization is to extract the summary of large documents of text. The great advantage of using summary is to reduce the reading time, reduces effort and cost. The process involved in this text summarization is to identify the important meaningful information available in many documents. This is to be done in 2 ways: Extractive and Abstractive. Extractive summarization can be done by selecting the important sentences from the original document. An abstractive summarization can be done by having awareness of the main concepts of the document and interpret them in a new way. This can be achieved through several supervised and unsupervised techniques such as CNN, RNN, LRC and several deep learning techniques were used which will be mentioned in different articles. Machine learning is also used to review the reviews of any e-commerce website like Amazon, it helps to know about the trends and behaviors of users to serve to the proper items and also provides reminders which are relevant [1] [2]. Then it will use those outcomes to display advertisements which are relevant to them [3][4]. The main advantage to use ML is, we need not to superintend every step of our project, instead it provides the machine to have the ability to make predictions [5] [6] [7].

The organization of the paper is as follows. Section II describes about Literature review and section III will discuss regarding machine learning techniques used for text summarization, after basics, we underlined all available automatic text summarization techniques and their performances in Section IV. Section V presents conclusion and future work and finally, Section VI presents references.
2. Literature Review

2.1 Machine learning techniques for text summarization

Supervised and unsupervised learning [8] are two types of machine learning techniques. Supervised learning means machine is learning under a supervisor and is trained with data which is already verified and labelled correct. In this learning technique, result is already known and confirmed. Once machine learns from true training data, then new unknown data is given to predict outcomes. It helps in optimization of various results according to criteria. They are mainly used to solve real world complex computational problems. Supervised learning is further classified into two techniques i.e., classification technique and regression technique. In regression technique, a single continuous output value is produced using training data. In classification technique, as the name says, it classifies or categorize input into some output category. Classification can be named as binary or multiclass classification depending on output categories. Author Cheng-Zen Yang in his paper proposed a supervised leaning technique to produce bug report summary.

Unsupervised learning technique, unlabelled data is given as input and algorithms are applied in order to get output. It is used in order to find out all possible patterns or classes in data. Model is not supervised in this technique. It is further classified into two different techniques, namely, clustering and association. Clustering technique will take input data, and after processing it, it will classify the data into various clusters based on common features. Association technique helps in connecting data objects or associating them in order to discover some relationship in a very large data set or database. Author Senthil Kumar proposed a method to produce bug report summary using unsupervised learning technique in his work “AUSUM: Approach for unsupervised bug report summarization”.

2.2 Evaluation parameters of generated summary:

- **Precision** - Precision is the ratio between the positive predictions which are predicted correctly and all the total predicted positive observations.

  \[
  \text{Precision} = \frac{TP}{TP + FP}
  \]

- **Recall (Sensitivity)** - Recall is the ratio between the positive predictions which are predicted correctly and all the observations in its class.

  \[
  \text{Recall} = \frac{TP}{TP + FN}
  \]

- **F1 score** - F1 score is the weighted average of precision and recall. Therefore, this score considers false positives as well as false negatives in to consideration. F1 score is more useful if we have an uneven class distribution.

  \[
  \text{F1 Score} = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}}
  \]

- **ROUGE** - The abbreviation of ROUGE is Recall-Oriented Understudy for Gisting Evaluation. A set of measures are used for evaluation of machine translation and automatic summarization of texts. It evaluates by comparing the summaries which are produced automatically or the translations against a group of summaries references.
3. Automatic Text Summarization Techniques

Author AKukkar [9] introduced one effective approach to produce a flexible and productive bug report summary as well as to minimize load and work of developer. Author used particle swarm optimization (PSO) approach for searching the effective semantic text. Author tried to address four central points that are, extractive bug report summarization, to increase the ROUGE score by selecting effective semantic text, sparsity of data and reduction of information. Proposed methodology used collection of comments and some feature extraction methods to generate or produce the bug report summary. Multiple summary subsets are produced and optimal summary subset evaluated by PSO optimization technique. Author compared the proposed approach with existing Email Classifier (EC) and Bug Report Classifier (BRC). ROUGE score was selected as one of the evaluation criteria and was calculated for all approaches. At the same time, the ROUGE score was compared with three human-generated summaries of 10 bug reports of Rastkar dataset. As a result, PSO approach summary subset was less redundant, and included all important points needed to be present in a bug report.

Author Beibei Huai and team [10] gave new Intention-based bug report summarization approach, alias IBRS which is based on intention taxonomy. This work considered sentences intentions in order to generate summary report. Sentence intentions were classified according to their taxonomy levels into seven categories: bug Description, fix solution, opinion expressed, information seeking, information giving, meta/code and emotion expressed. Now, sentences are categorized in specific intention with the help of pattern matching and machine learning model. Finally bug report summary is produced. This summary was compared with BRC (Bug Report Classifier) and found better in terms of precision (5% improved), recall (3% improved), F-score (3% improved) and pyramid precision (5% improved).

Creating summary is selecting important topics of sentences as well as recognizing relevant relationships among those concepts which are mentioned in that text. The key problem is generalization which is identified by ATS task. Stating an example: summarization financial or medical reports are conceptually different from summarizing news articles. To solve the above issue, to achieve more relevant summary, this paper, author Hernández-Castañeda, Ángel [11] proposes (EATS). EATS is based on clustering technique which holds by GA i.e. Genetic Algorithm to get relevant topics in the proposed document. To identify key sentences in the clusters, this method includes Topic Modeling Algorithm (LDA) which is based on keywords those are generated automatically. This clustering technique needs LDA and Doc2Vec to map text to numeric vectors along with tf-idf and n-grams. This method is tested on DUC02 dataset to achieve the goal of producing summaries as close to as human generated summaries. TUCII dataset is used to test with multi domain and multi-language frameworks. This system achieves more effectiveness than earlier methods based on the evaluation results such as skip grams(ROUGE SU) and also (ROUGE-1), bi-gram (ROUGE-2). As a result, the obtained summaries showed matches of unique words as well as appends context through matching the adjoining words.

Shubhra Goyal Jindal et al.[12] suggested rapid automatic keyword extraction and term frequency-inverse document frequency method which helps to get proper keywords and key-phrases with an applicable score. For extraction of sentences, fuzzy C-means clustering can be utilized to bring out sentences. For sentence selection rule engine is used. Hierarchical clustering is used for redundancy removal, re-rank generated summary and to enrich the summary. The suggested approach is assessed on Apache Project Bug Report Corpus
Author Ashima Kukkar’s [13] worked on the bug report summarizing as well as analyzed whether the bug report summary helps in finding the appropriate severity of bug. Unsupervised bug report summarization system was followed and the approach used in this work is swarm intelligence. In order to generate bug report summary, step by step process is done. Initially, pre-processing of text was done which further includes tokenization, stop-word removal and stemming. After preprocessing, all unwanted data is removed and features are extracted. Then sentence score was calculated using n-gram technique. In the next step, sentences were combined to generate summary subsets and for every subset, a relative subset score was calculated. Particle Swarm Optimization (PSO) approach is used to select optimal summary subset. Again, features scoring was done of selected optimal summary subset and relative score was calculated using Ant Colony Optimization (ACO). Model training is also done using these optimized features. This data result was compared with three benchmark datasets and proved better results in terms of Precision, Recall and F-measure.

To provide the developers with the important information, authors CHENG-ZEN YANG et.al and YU-HAN CHUNG [14] have proposed a two-layer semantic model (TSM). TSM uses two layers to identify the informative sentences for summary. TSM model is trained based on the textual features which are generated from two-layers. The first layer uses an ENR model to maintain the semantic information. It gives informative sentences by using a rule-based-Algorithm called algorithm1. The second layer uses BRC textual model to get the textual features by using the informative sentences which are generated from first layer. Based on the textual features which are generated form second layer a supervised logistic regression model is used for training the TSM. The datasets which are used by this TSM summarizer is the BRC dataset which has 36 bug reports in total. By comparing the TSM model with the BRC and AUSUM, it is proven that the TSM is outperformed and has highest value of F1 measure that is 0.530. And the F1 values of others are 0.40 and 0.50 respectively. The relative improvement achieved by TSM when compared with BRC and AUSUM are 34.3% and 7.3% respectively.

Bug report summarization has some characteristics which effect the performance of the summarization technique. To resolve this problem the author Xiaochen Li and his team [15] have proposed an unsupervised approach called DeepSum. To get the effective summary from bug reports DeepSum summarizer works based on deep learning. The main part of the DeepSum is a stepped auto-encoder network, which concludes the bug report summary established from the hidden layers of the network. The data sets which are used for evaluating the summarizers are summary data set and authorship data set. DeepSum achieved good results in terms of metrics like F-Score and R-1 when compared with comparative algorithms. DeepSum outrun the other algorithms by up to 0.119 and 0.092 in terms of F-score and R-1 respectively when tested on SDS data set. And achieves equal results on ADS dataset when compare with comparative algorithms. And the F-score and R-1 values of DeepSum are 0.462 and 0.563.

In this paper the authors Akalanka Galapathi and John Anvik [16] extends the Rastkarel’s model. They have find out an extractive way to generate summaries for bug reports without using the complex features which are used in the Rastkar’s model. They have...
find out how precision and recall varying when using and not using complex features. These features are Length and Lexical. For sentence classification they have used a logistic regression model. For example they have used sklearn package for classification. To observe the performance of both individual and combined features, they have used different types of logistic regression models. By comparing author’s with Rastkar’s model, they have observed the following. Author’s model Precision, recall and F-Score values: 44%, 24% and 29%. Rastkar’s model Precision, recall and F-Score values: 57%, 35% and 40%. There is decrease in the performance due to usage of fewer features.

The authors of this paper are Beomjun Kim, Sungwon Kang and Seonah Lee [17]. Generally bug reports are the lengthy conversations. While reading this, developers face many problems. To solve this problem, the authors have used weighted page-rank algorithm to maintain multiple relationships. They have used Relation based Bug Report Summarizer (RBRS) with duplicates, blocks and depends on relations for automatically generating the bug report. For weighting they have used page rank algorithm. RBRS page rank score to each sentence in the bug report. They have used the bug report corpus as data source. Duplicate relations are used for improve the quality of the bug report. By comparing proposed model (RBRS) with the existing model Page Rank based Summarization Technique (PRST) . They have observed the following for precision, recall and F1-score for bug reports without duplicates relationships are 50.91%, 50.93%, 50.92% and 44.50%, 44.79% and 44.64% respectively. By comparing proposed model (RBRS) with the Existing model (PRST), the following is observed precision, recall and F1-score for bug reports with duplicates relationships the values are 51.69%, 51.82%, 51.76% and 49.71%, 49.77%, 49.74% respectively.

Generally, in software industry, software maintenance for fixing the bug reports, 45% of software development will be spent. A familiar Bug report called Bugzilla repository which had around 12,36,000 bug reports. The tasks like bug fixing, duplicate bug identification and bug-traing are tough tasks which will be time consuming. To perform above said tasks effectively He Jiang Et Al[18] suggested an efficient approach that is constructing attributes which are hardly influence the presentation of supervised algorithms for summarization of bug report. In this paper, to reveal the existing methods for constructing attributes, they first conduct a survey on the authors of 40 papers. Secondly, they suggested a new method called Crowd-Attribute (CA) to fully conclude attributes for bug report summarization in crowd sourcing. Thirdly, to achieve more accurate summaries of bug reports, they construct 11 new attributes successfully by trying the new method CA and suggested a recent approach named Logistic Regression with Crowd sourced Attributes (LRCA). In LRCA, 11 attributes are assessed to feed the statistical model called as logistic regression, over the training set. To predict its summary for a new bug report, the attributes of each sentence are calculated and fed into the trained model. The LRCA is tested to evaluate its effectiveness on a large-scale datasets with 1,05,177 bug reports and even experimented on the dataset SDS with 36 bug reports which are annotated. LRCA improved the result by 10.11%, 1.33%, 8.94%, and 5.89% with regard to recall, precision, F-score and Pyramid of supervised approach (BRC), respectively.

Surbhi Bhatia et al. [19] is providing the comparative knowledge about abstract techniques and extractive techniques. Abstractive type of summarization uses graph based technique for duplicate sentences. It follows a sequence of operations like constructing graphs, ensuring
sentence correctness by using some of the constraints and finally scoring the sentences individually by combining sentiments using SentiWordNet. Extractive type of summarization uses the technique of (PCA) Principle Component Analysis which reduces the number of dimensions of data and finds the summary based on the ranking of most relevant sentences. The Opinosis dataset (Garmin Nuvi 255W GPS). The present abstractive technique achieved 13% of performance improvement. The results of both the techniques are evaluated through ROUGE score. The future challenge for the researchers is to employ semi abstractive, extractive techniques to carry out the benefits of both approaches. Also, we can study excessively on threshold values, how they are affecting performance.

JIANLI DING et al. [20] addresses the problems of insufficient summary precision, insufficient use of semantic information and semantic loss in the generated summary. They proposed seq-to-seq architecture through dual-encoder is used to provide ample of semantic information. Here position embedding and word embedding technology. The information retrieval techniques such as Tf-Idf, parts of speech, key score are added to word’s feature. In this research, to solve the problem of semantic loss Gain-Benefits gate structure is designed in the encoder. The results are tested against LCSTS and SOGOU datasets (News headlines). The future enhancement could be, the performance of this model needs to be still improved when it is dealing with special names and also familiar nouns. So, further research can be done in area.

HAOJIE ZHUANG et al. [21] proposed a model which consists of three components. They are – encoding long input text to shorter form using a generator. Secondly used a discriminator to train the generator to produce summaries which are human-readable. This research focused on single-document summarization. The data set they have used is CNN/DailyMail. In the present years of research, abstractive methods which are based on deep learning have shown promising results, specifically the methods that are based on supervised learning CNN to encode the input, another feed forward neural network is used to generate the summary. RNN to balance between extractive and abstractive approaches hybrid pointer-generator network is used. Pointer network for copying words directly from input text and generator network generate new words and phrases. The proposed model GAN (Generative Adversial Network) involves 2 opposing networks: First generator- which generates data samples. Secondly- discriminator- which distinguish network real data from input text and the data samples which are generated from generator. The generator should be able to generate samples that are “most similar” to real data when they well-trained. The future work is to combine the proposed model GAN with LSTM-CNN based deep learning and also the generator to be replaced with a pre-trained BERT model and processing the input phrase-by-phrase my help further to improve the accuracy.
## 4. Different text summarization techniques

### Table 1 Different text summarization techniques

| Author/Ref | Processing Steps | Features | Approach | Dictionary | Evaluation measure | Result | Future scope |
|------------|-----------------|----------|----------|------------|--------------------|--------|--------------|
| A Kukkar [9] | Collection of comment, feature extraction methods, summary generation optimal summary subset selection | Collection of comments | Particle swarm optimization (PSO) | Rastkar dataset | Rouge-1, Rouge-2, Rouge-3 | 79.14, 75.83, 75.17 | To achieve more accurate summary, other optimization techniques will be applied |
| Beibei Huai [10] | Sentence intentions to generate summary | Intentions of Sentences | Intention-based Bug Report Summarization approach | BRC | Precision recall, F-score | 5%, 3%, 3%, 5% | Need to improve intention classifier characteristics |
| Angel Hernández-Castañeda [11] | Feature extraction | Word unigrams, Bigrams, features | TF-IDF, D2V, LDE, OHE | TAC11 | F-score | 0.249 | To improve performance apply more classification Techniques |
| Shubhra Goyal Jindal [12] | Preprocessing, Keyword and sentence identification, Clustering, Rule designing Clustering | Keyword and Sentence identification | TF-IDF, RAKE, Fuzzy-C means, Hierarchical clustering | BRC, APBRC | F-score, Precision, Recall | 80.1%, 78.22%, 82.12% | Analyzed the impact of different clustering algorithms. ROUGE evaluation metric is used for documents which includes text. |
| Ashima Kukkar [13] | Feature Extraction and sentence score calculation | Feature Score Calculation | PSO and Ant Colony optimization | Benchmark datasets | Precision recall, F-score | 79.74, 78.79, 79.76 | More study needed to be done on relation between bug summarization and severity classification. |
| Cheng Gen Yang [14] | Extracting the important information using Anthropogenic and Procedural classes | Textual features and informative sentences | TSM uses ENR and BRC textual model | BRC | F1 | 0.530 | Some informative sentences are classified into others that needs to be improved. A dictionary is needed for word processing |
| Author(s)                          | Methodology                                                                 | Dataset/Approach                                      | Metrics                        | Results   | Comment                                                                                           |
|-----------------------------------|-----------------------------------------------------------------------------|-------------------------------------------------------|--------------------------------|-----------|--------------------------------------------------------------------------------------------------|
| Xiaochen Li [15]                  | Vector generation Tokenization, Sentence selection, Word weighting, Sentence weighting | Deep sum a stepped auto-encoder network                | Summary Data Set and Authorship Data Set | F-score   | In the future there is more needed to be studied for automated model for examine the length of bug report |
| Akalanka Galapathi [16]           | Selecting highest probability sentences, Complex features, Length and lexical features | A logistic regression model                           | BRC                            |           | Some complex features with high variability are explored                                        |
| Beomjun Kim [17]                  | Analyzing duplicate, blocks or depends-on graph generation sentence score    | Relation based Bug Report Summarizer (RBRS)           | Bug report corpus              | Recall    | In future focus must be on larger corpus.                                                       |
|                                  | Selecting top scored sentences, Constructed 11 attributes by CA approach    | Logistic Regression with Crowd sourced Attributes (LRCA) | SDS-36 bug reports            | F-Score   | And also needed more study on preprocessing techniques to improve the quality of bug report summaries |
| He Jiang [18]                     | Constructing new Attributes, Ranking the sentences based on most relevancy. | Constructing graphs, ensuring sentence correction based on some constrints | SentiWordNet, Principle Component Analysis |           | Semi abstractive, extractive techniques will be working together to get the benefits of the 2 approaches, Study excessively on threshold values, how they are effecting performance the summary. |
| Surbhi Bhatia [19]                | Tf-Idf, Parts-of-speech, keyscore, Dual-Encoder                              | LCSTS & SOGOU dataset                                 | Rouge-1, Rouge-2, Rouge-L      |           | The performance is to be improved further when it is dealing with special names &nouns which are familiar. |
| JIANLI DING [20]                  | Encoding long input to shorter form by using a generator                    | GAN model in which we have Pointer network & generator network | CNN/DailyMail                     |           | Teaching the generator to compose / construct abstractive sentences properly is still a challenging problem to continue this research. If the generator to be replaced with a Pre-trained BERT model and processing the input phrase-by-phrase may |
| HAOJIE ZHUANG [21]                |                                                                          |                                                       |                               |           |                                                                                                  |
Yang Gao, Yue Xu[22] Representing the sentences as vectors Topic, sentence predictions CCTSenEmb DUC2005, DUC2006 Rouge-1, Rouge-2 38.65 7.73 The future work involves in upgrading the topic learning process to broaden CCTSenEmb usability.

5. Conclusion and future work

Researchers depend on effective summaries from large documents of text. The purpose of summaries is to produce a quick review of any topic and to work in detail on more relevant and appropriate paper. Developers work on different platforms but they need to deal with bug reports. While reading the generated and assigned bug report, a developer should understand what a bug is, why it is raised and how to deal with it to get the expected result. So, to resolve the bug issue, an appropriate summary of bug report should be available.

This paper includes survey of different bug report summarization techniques which help developers to deal with their domains. Also, a performance analysis is done for every technique and different metrics like precision, recall, F-score and ROUGE score were calculated. This work gives a better clarity to understand which summarization technique should be used to get more effective summary. In future, we will work upon various combinations of techniques in order to obtain better results.

References

[1] Dutta, Sushama Rani, et al. "IoT in Autism Detection in Its Early Stages." Internet of Things: Enabling Technologies, Security and Social Implications. Springer, Singapore, 2021. 47-58.

[2] Dutta, Sushama Rani, Sujoy Datta, and Monideepa Roy. "Using cogency and machine learning for autism detection from a preliminary symptom." 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2019.

[3] Dutta, Sushama Rani, et al. "A Machine Learning-Based Method for Autism Diagnosis Assistance in Children." 2017 International Conference on Information Technology (ICIT). IEEE, 2017.

[4] Giri, Soumyajit, Sushama Rani, et al. "A real time epidemic alert generation system for rural areas using WBANs and kiosks." 2017 International Conference on Information Technology (ICIT). IEEE, 2017.

[5] Dutta, Sushama Rani, and Monideepa Roy. "EDOT: Context-aware tracking of similar data patterns of patients for faster diagnoses." 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2017.

[6] Dutta, Sushama Rani, and Monideepa Roy. "Providing context-aware healthcare services using circular geofencing technique." 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2016.
Dutta, Sushama Rani, et al. "Handling Scarcity of Data in Autism Identification using Binary Imputation Method" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019.

Donalek, Ciro. "Supervised and unsupervised learning." Astronomy Colloquia. USA. 2011.

Kukkar, Ashima, and Rajni Mohana. "An optimization technique for unsupervised automatic extractive bug report summarization." International Conference on Innovative Computing and Communications. Springer, Singapore, 2019.

Huaie, Beibei, et al. "Mining Intentions to Improve Bug Report Summarization." SEKE. Vol. 2018. 2018.

Angel Hernandez-Castaneda et al. "Extractive Automatic Text Summarization Based on Lexical- Semantic Keywords". In: IEEE Access 8 (2020), pp. 49896–49907.

Shubhra Goyal Jindal and Arvinder Kaur. "Automatic Keyword and Sentence-Based Text Summarization for Software Bug Reports". In: IEEE Access 8 (2020), pp. 65352–65370.

Kukkar, Ashima, Rajni Mohana, and Yugal Kumar. "Does bug report summarization help in enhancing the accuracy of bug severity classification?." Procedia Computer Science 167 (2020): 1345-1353.

Yang, Cheng-Zen, Cheng-Min Ao, and Yu-Han Chung. "Towards an Improvement of Bug Report Summarization Using Two-Layer Semantic Information." IEICE TRANSACTIONS on Information and Systems 101.7 (2018): 1743-1750.

H. Jiang, X. Li, Z. Ren, J. Xuan and Z. Jin, "Toward Better Summarizing Bug Reports With Crowdsourcing Elicited Attributes," in IEEE Transactions on Reliability, vol. 68, no. 1, pp. 2-22, March 2019, doi: 10.1109/TR.2018.2873427.

Galappaththi, Akalanka. Automatic sentence annotation for more useful bug report summarization. Diss. Lethbridge, Alta.: University of Lethbridge, Department of Mathematics and Computer Science, 2020.

Kim, Beomjun, Sungwon Kang, and Seonah Lee. "A Weighted PageRank-Based Bug Report Summarization Method Using Bug Report Relationships." Applied Sciences 9.24 (2019): 5427.

Jiang, He, et al. "Toward better summarizing bug reports with crowdsourcing elicited attributes." IEEE Transactions on Reliability 68.1 (2018): 2-22.

Bhatia, Surbhi. "A Comparative Study of Opinion Summarization Techniques." IEEE Transactions on Computational Social Systems (2020).

Ding, Jianli, et al. "Generative Text Summary Based on Enhanced Semantic Attention and Gain-Benefit Gate." IEEE Access 8 (2020): 92659-92668.

Zhuang, Haojie, and Weibin Zhang. "Generating semantically similar and human-readable summaries with generative adversarial networks." IEEE Access 7 (2019): 169426-169433.

Gao, Yang, et al. "Jointly learning topics in sentence embedding for document summarization." IEEE Transactions on Knowledge and Data Engineering 32.4 (2019): 688-699.