Information Search Mechanisms for Government Entities using Machine Learning and Natural Language Processing Techniques

Ricardo Ponciano
TIMWE Lab
Parkurbis, Tortosendo
Portugal

João Santos
TIMWE Lab
Parkurbis, Tortosendo
Portugal

João Isento
TIMWE Lab
Parkurbis, Tortosendo
Portugal

ABSTRACT
Nowadays, huge quantities of data are produced on the Internet. That data can be used to create added value for the companies. Nevertheless, it is necessary to evaluate the quality of the gathered information in order to avoid the creation of inaccurate insights that can lead to wrong decisions. Thus, on this paper is presented a study of a new search information mechanism for government management entities, which will be able to categorize the retrieved information through the usage of Natural Language Processing and Machine Learning techniques. Web crawling mechanisms are also integrated to gather the information from web sources.

Keywords
Government, Machine Learning, Natural Language Processing, Classification

1. INTRODUCTION
The last few decades presented many technological developments that led to great transformations in human society. One of the most important technological evolutions was the dissemination of the Internet over the world as the global connection network. Nowadays, in many places, it is possible to access Internet through high speed connections of about hundreds of megabytes per second. In addition, Internet can be accessed almost anytime and anywhere, since the increasing use of smartphones, tablets, and other mobile gadgets allows the establishment of the connection on mobility scenarios. These facts help to justify the great amount of data that is produced, which is estimated to be around 2.5 quintillion each day [1]. Some of this information can be used in different business areas to better inform the decisions that need to be taken. In this way, multiple analysis and decision support tools were created for different business areas (ex: trade, industrial production, sales, marketing, management, banking, etc.) [2], [3], [4], [5]. Many of these tools are intended to present business evolution to managers through the evaluation of different measurable indicators. Furthermore, these tools also aim to provide greater control over the organization's structure, so that managers can have a more detailed view of the work carried out by their employees through a closer monitoring of the tasks that they perform [6]. There are even sources that report detailed studies on the benefits of using management tools, as highlighted in [7], where it is indicated that companies that assess and review the objectives of their workers at least 3 times a year they are 45% more likely to have above average financial performance, and also having a 64% probability of being more effective in retaining or reducing costs when compared to competing companies. However, the management and decision support tools highly depend on the information that is initially provided. If the quality of the information is poor, the output that the decision support tools are going to present to their users will be doubtful [8]. Without high-quality data, meaningful information cannot be generated, and consequently, no knowledge will be created. Government management entities are a good example where the lack of high-quality information can badly influence the policies that are going to be created [9], [10], [11]. And without a clear understanding of what the citizens want and value, it will be impossible to create policies and develop services that accurately respond to the requirements of today's society [12]. Thus, and given the widespread discontent regarding the policies that government entities apply, it is understandable that the government sector is one of the worst ranked in the consumer satisfaction index [13].

In order to improve the quality of the information that is made available to government entities, it is necessary to apply mechanisms that can only gather data that is really relevant to the business context. One of the available options is to use the Natural Language Processing (NLP) paradigm, which allows computer systems to understand the context of the received inputs, written or spoken [14]. Thus, it is a sub-area of Artificial Intelligence that seeks to make machines capable of communicating through the human language [15]. NLP can be used to classify the context of a given phrase, or even a whole document. It involves several steps, where entities, parts of speech (noun, adverbs, adjectives, etc) and other elements have to be identified [16]. Also, Machine Learning (ML) algorithms can be used on those type of solutions in order to help the system to learn the most relevant words of a given category over the time [16]. Then a system can understand what the context of a given input is, comprehend the referred action, and identify the most relevant words for the given context. On this paper is proposed a new information search engine that will help government entities to gather relevant information about different categories (health, culture, education, politics, etc) related to the local community. Thus, workers of government management entities can quickly access detailed information, or news articles, about a topic of interest for their daily tasks. Web crawling mechanisms will also be implemented on this solution in order to gather the needed information on the web sources [17]. The information search engine will be implemented on a decision support tool for government management entities and on a mobile application available for citizens, which is intended to help them to be better informed about important information related to the municipality and its management bodies. These interfaces belong to the ADAA ecosystem - Intelligent and Collaborative Government Management System - which is a

research and development project co-funded by European funds.

The remainder of the paper is organized as follows: in section 2 is presented relevant work related with the most important technologies and techniques used for the development of the information search engine, namely NLP, ML and web crawling; section 3 presents the methodology used to create the search information engine; and section 4 presents the results and conclusions of the work depicted on this paper.

2. RELATED WORK

As technology has evolved, human society has been trying to find new ways to benefit from disruptive inventions that could ease some everyday tasks. Thus, there have been developed mechanisms that give machines the possibility to understand the context of the inputs presented by human beings, which consequently makes the machines more capable of providing more adequate answers to users. To do so, artificial intelligence mechanisms need to be implemented on machines, and for this specific purpose, Natural Language Processing (NLP) mechanisms, and other related subjects, have to be considered [18], [19]. Nowadays, NLP mechanisms are widely used on intelligent conversational interfaces, usually called chatbots [20], [21], and on optimized Search Engines, such as Google, Amazon, Bing, among others [22]. On [22] are presented some issues related with classical search engines, namely token matching, contextualization of the query and query misunderstanding, which can lead to less recall and less precision. The solution given by the author considers the use of Deep Learning techniques through multiple steps. Some academic and scientific works focus on the use of NLP techniques in the development of intelligent chatbots, in which the knowledge obtained can be extrapolated to the creation of optimized information search systems. For example, in [23], the authors present a survey on the techniques used for the development of conversational systems. This type of systems requires contributions from different areas, such as speech recognition, speech parsing, NLP, identification of key terms, artificial intelligence, among others. In order to process and manipulate the text that results from speech recognition and its conversion (audio to text), it is necessary to organize the text into sentences and then segment it into words to extract the content. To achieve this purpose, some toolkits can be used, being the Natural Language ToolKit (NLTK) - free plugin for Python - one of the most famous. The NLTK is used to segment words in text strings, and then separate the text into parts of the speech by classifying words according to their positions and functions in the sentence. The classified words that result from this process are then processed to extract the meaning and produce an appropriate response to the user. In addition to this toolkit, the authors also identify a number of important techniques for creating chatbots capable of processing natural language, namely (i) parsing, (ii) pattern matching, (iii) Artificial Intelligence Markup Language (AIML), (iv) chat scripts, (v) relational databases, (vi) Markov chains, (vii) linguistic tricks, and (viii) ontologies. In [24], the authors discuss the implementation of intelligent chatbots, namely the data-driven chatbots, which use information from interactions between humans or between humans and chatbots to build the knowledge base that will allow the development of conversations with users. To achieve this, information retrieval (IR) or ML mechanisms can be used. IR-based chatbots work as a search engine, in which the query corresponds to the user's input while the search result is the chatbot's response, using a "question-answer" pair logic. Word space vector models and models based on Term Frequency-Inverse Document Frequency (TF-IDF) are regularly used in IR-based chatbots. As for the chatbots based on ML mechanisms, they can use sequence-by-sequence learning models (seq2seq) and reinforced learning models. The seq2seq models use Recurrent Neural Networks (RNNs), and the models are trained to map sequences of inputs into sequences of outputs. For this, a neural network is used in the encoder to read the input sequence and convert it into a contextual vector to be decoded at the output. On the other hand, the authors of the work depicted in [25] propose a novel algorithm for retrieving relevant documents using semantic web based on the NLP concept. Thus, the NLP is used to analyze the user query in terms of Parts of Speech, and the extracted terms are compared to the domain dictionary to identify the relevant domain of the user interest. Then, when a user performs a query, a parser is used to extract nouns, adjectives, prepositions, among other elements of the query and store them in a term list table. If most of the extracted words belong to one domain, then this domain is taken as the dominant domain; otherwise, the user has to confirm the dominant domain. On the next step, each document is split into sentences and each sentence is further divided into words. Each word is compared with term list and domain dictionary for matching. The ''sentence weight'' is incremented if a match is found. The same process is continued for the whole document and cumulative document weight is calculated. The experimental results of the proposed system achieved an accuracy of 95%. However, the time taken to classify a document is higher when compared with existing search engines. The text classification methodology is addressed in other works: for example in [26] are reviewed some of the real-world use cases and various methods that exist to develop a text classification solution, while categorizing them and listing the strengths and weaknesses of each one; in [27] is proposed a deep learning model for text classification, which takes advantage of a convolutional and recurrent layer model in order to maintain important information awareness through time and not only recognize relationships between words and sentences. Although achieving better accuracy than other traditional models (convolutional-only models, for example), by using less parameters and by using recurrent layers instead of pooling layers, the results become worse when the number of convolutional layers is increased. As the number of layers is increased, it is observed a loss of detailed information, which in turn creates problems when capturing long-term dependencies.

In order to obtain relevant information from multiple web data sources, APIs or data crawling mechanisms have to be used. The work in [17] describes how web crawlers are used to gather information from the web. Thus, the authors refer the steps involved in the operation of web crawlers, from removing a URL from a list of URLs, determining the IP address of the hostname, downloading the related documents, to extracting the links available in the documents. The two main crawling methodologies are also referenced: the Blind Traversing Approach, in which the URL selected for crawling is the one that appears just after the URL where the crawling was just done; and the Best First Heuristic Approach, in which the next URL to be crawled depends on calculations or similarities with previously established conditions. In addition, some challenges involving web crawling, such as the large volume of existing data, the difficulty in extracting relevant content, the possible overload on servers, and issues related to access to private or copyrighted data are also mentioned. In [28] is presented a work that focuses on the

development of a web crawler for extracting information from Big Data sources. Thus, it is expected that the crawling tool will be able to browse hundreds of thousands of sources uninterruptedly in order to extract relevant information using the Depth First Search Algorithm. The importance of using crawling tools in Big Data sources is relevant for government services, since there are large amounts of dispersed information that need to be accessed in a quick, simple and orderly manner. However, and despite showing interesting results, the work presented needs improvements in terms of bandwidth usage and depth of search in sources. On the work presented in [29] is studied the use of web crawling mechanisms on Facebook for the analysis of user opinions in order to improve government services in a rural community in South Africa. The premise followed by the authors of this work is similar to the one we followed in the ADAA project, since they believe that if government entities properly monitor the opinions of citizens in relation to the government services provided to them, these services can be improved. At the functional level, the system proposed by the authors allowed the extraction of Facebook data, text extraction, text preprocessing, text indexing and search indexing. However, the results of this work do not demonstrate any direct relationship with government services, demonstrating only the possibility of viewing the most frequent words in posts on Facebook, and finding specific posts through keywords entered by the user.

After analyzing some existing work about NLP and ML mechanisms applied on the context learning, and web crawling mechanisms applied used to gather relevant information on web data sources, we concluded there does not exist a single and optimized solution for those purposes. Not many solutions are referred to the government sector, since optimized information search engines are usually applied in other areas. The studies about web crawling solutions that we found mention some difficulties about operating with huge datasets, and do not directly address relevant findings for the government sector. Thus, studies for the development of innovative information search engine tools for governments can still be performed.

3. SOLUTION IMPLEMENTED

In the ADAA project, there was built a module that uses ML and NLP with the intent of extracting relevant text from the web, evaluate its context, and the classify the retrieved information. An illustration of the different stages of the NLP process is shown in Figure 1. Three libraries have been used to develop the desired solution: (i) the Scikit-Learn [30], which is one of the ML libraries for Python that aims to assure the interoperability between numeric and scientific libraries, including NumPy and SciPy. This library provides algorithms for supervised and non-supervised learning by means of a stable Python interface. Some model groups offered by this library are comprised of supervised groups, parameters adjustments, multi-learning, dimension reduction, grouping, feature selection and cross-validation; (ii) the TensorFlow, which is an open source deep-learning library from Google [31], being composed of a computing structure used mainly for expressing algorithms that wrap sets of matrix operations. As neural networks are represented in form of different computational graphs, this global implementation is executed using TensorFlow in the form of a sequence of matrix operations, in other words, this library uses N-dimensional matrixes that correspond to the data being used. The use of distributed computation is the main advantage of TensorFlow, particularly on multi-processing situations. TensorFlow allows utilities for serial connected data processing, where the output of one element is the input of the next one. It is made of internal modules for visualization, serialization and mandatory inspection of diverse elements; and (iii) Keras [32], which is a high-level neural network API written in Python. It is a flexible and easy-to-use library, used to develop neural networks and is executed in conjunction with Theano, TensorFlow and Cognitive Toolkit. Keras supports recurrent networks and combinations that allow simple and fast prototyping. This library comprises a large set of neural network implementation blocks, optimizers, layers, activation functions and tools that allow to work with images and text data.

Before applying the text classification mechanism based on NLP and ML mechanisms, a web crawler mechanism was developed in order to gather news articles from the web. The data sources (all Portuguese, and presented on Table 1) were static and segmented into seven categories: culture, sports, education, politics, health, society and uncategorized news. Each news article had an URL, title, date, category, abstract and body. The categorized news were used for the training model, while the uncategorized news belonged to the test model.

Then, the knowledge base module has been created and divided into two models: the training model and the testing model used to classify uncategorized news. In the training model, news obtained through the web crawler were inserted on a repository, with the category of each news article being previously known. A neural network algorithm and the concept of Term Frequency – Inverse Document Frequency (TF-IDF) were used to find the words with higher relevance. In the testing model, an initial pre-processing of unknown text was made using the generated model in order to correctly classify that same text.

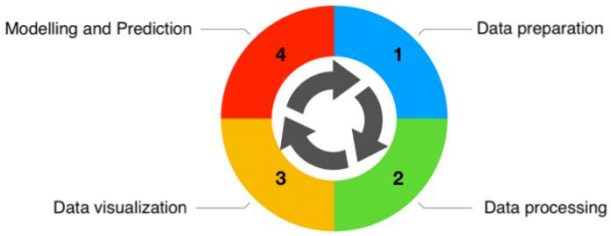

Figure 1: NLP Stages

Table 1. List of web sources used to retrieve the information needed to build the training module

https://www.rtp.pt/noticias/
https://tvi24.iol.pt/ultimas
https://www.jn.pt/tag/saude.html
https://www.jn.pt/tag/educacao.html
https://www.jn.pt/tag/cultura.htm
https://www.jn.pt/tag/politica.html
https://www.jn.pt/tag/desporto.htm
https://www.jn.pt/tag/nacional.html
https://www.educare.pt/noticias/
https://www.publico.pt/
https://www.jornaldofundao.pt/

https://www.abola.pt/
https://www.record.pt/
http://forumcovilha.pt/
https://radio-covilha.pt/
http://www.noticiasdacovilha.pt

3.1 Data Preparation

A text classifier is useless without accurate training data. Thus, it is necessary to assure the usage of meaningful data. ML algorithms can make predictions with the knowledge acquired from previous interactions. When teaching the algorithm that a specific set of tags is expected as an output for a given text, it can learn to recognize text patterns, such as a sentiment expressed by a tweet or a topic mentioned in a client review. In order to provide a truthful output, the data used as an input for the algorithm must be coherent and representative of the context that is necessary to study.

One of the reasons ML is becoming increasingly popular for the classification methodology is related to the great variety of open source libraries available for developers. Although a basic level of understanding and knowledge is required to create and build ML models, those libraries offer a reasonable level of abstraction and simplification for the users. Python, Java and R offer a vast selection of ML libraries that are actively developed and provide a diversified set of resources, performance indicators and other capabilities.

The data available on web sources can feed the ML algorithms, thus serving as the knowledge base of the solution. To achieve that purpose, web scrapping mechanisms, APIs or public datasets can be used. In the ADAA project, a web scraper was created in Java to gather news from a certain set of news websites, generating a database of data text archives. The web scrapping mechanism allowed the automated collecting of specific information present on third party websites by downloading and parsing the HTML code [33], [34]. In this sense, and for the specific case of the ADAA project, the web scrapper uses a crawl agent as a search engine to find categorized and uncategorized news from referral URLs of each website that will be used as a data source. To identify and gather the information, the Jsoup parser [35] was used, which is frequently implemented in Java due to its implementation simplicity.

The training data for the model was separated in different folders, where text archives for each category were added. Every category is represented by a folder. To prepare the data, the set was uploaded in a Panda library data frame using Python, with Jupyter Notebook as the interface, building in this way a model with four columns: filename, category, news and link.

In the news column, all the special characters, punctuation and digits were removed, and all the text was lowercased. Next, the dataset was split in training and testing sets in order to teach and test the classifier, with 80% of the data being considered for training and 20% for testing. Besides, the destination column was codded to allow ML models to be used.

3.2 Data Processing

The data processing was the next step of the adopted methodology, consisting in the transformation of the text data into vectors using an existing dataset.

For that purpose, the Tokenizer class was used. This class allows text vectorization by transforming each text entry in a sequence of integer numbers or in a vector where each token coefficient can be binary, based on the word count and on the TF-IDF score.

The TF-IDF score represents the relative importance of a term in a given document or in a complete text. The score is composed of two factors: the first evaluates the normalized term frequency (TF), and the second evaluates the inverse document frequency (IDF), calculated as the logarithm of the number of documents in corpus divided by the number of documents containing the term.

The destination column (category) was encoded to allow its usage in the learning models that were needed. The next step in the text classification methodology was the training of a classifier using the resources obtained in the previous step. The created model follows the Keras Sequential Model, which uses the Convolutional Neural Network on the input level to compute the output level. This allows the connection of each entry region of the neural network to an output neuron. Each layer applies different filters and combines the results, with the final model being stored in memory along with the vocabulary token archive.

The accuracy of the model is verified, and thus the model is only used if the score is greater than 80%. If the score is lower, the model is discarded, and new data is obtained. In the end of this process, a confusion matrix is created, as shown in Figure 2, which allows a deeper visualization of the algorithm performance [36]. Each line of the matrix represents the instances of an expected class, while each column represents the instances of a real class (or vice-versa) [37]. As it can be seen on the confusion matrix, some news about politics were wrongly categorized as health news (11%), while some news about society were incorrectly classified on sports (9%) and society (9%) categories.

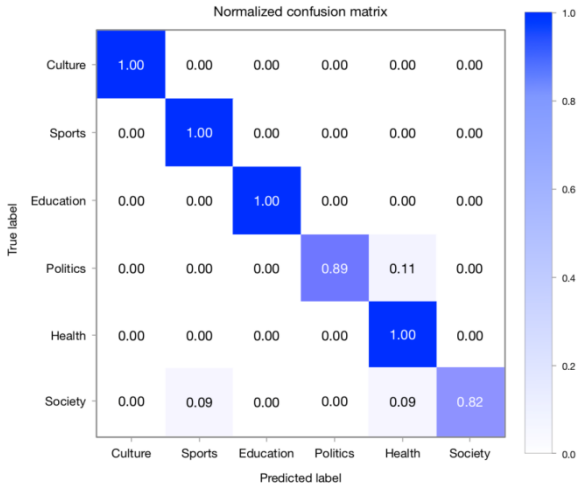

Figure 2: Confusion matrix for the data categorized during the performed study

3.3 Data Visualization

From the model that was already created, which contains the necessary components to allow the final text prediction, there was used the data frame structure early explained, imported the non-classified text and added a column of keywords to each news articles.

Thus, in the end is created a visual model of the classification process that allows a better understanding of the obtained

results.

3.4 Modelling and Prediction

Once the classification task is completed, it is necessary to convert the text data in order to allow its representation on a ML algorithm. For that purpose, a word poll model (Bag-of-Words: BoW) [38] can be used. The BoW model is simple to understand and implement, allowing the classification of a given text (used as a vector) and the retrieving of important information. It is a way of extracting text resources to use in the ML algorithms, such as the frequency of occurrence of words and the relevant features to be used for training. For this approach, words were used as tokens for each observation and thus, frequency of each token was discovered. A custom BoW archive was created, which can be updated every time there is the need to increase the model accuracy. However, in the first place, it was necessary to remove stop-words from texts.

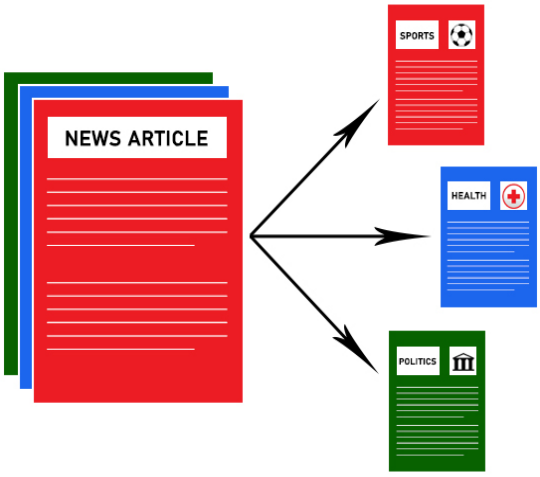

Figure 3: Diagram that depicts the data classification process

It is necessary to remove stop-words due to the particular nature of the words (ex: “a”, “and”, “but”, “how”, “or”, …), since they do not contain enough significance or importance to be used in the classification process. To save database space, or to improve the processing time, a good list of stop-words is needed in order to create a classification process with good accuracy. Thus, it is possible to easily remove them by storing a list of words and using it to pick out and trim the final text.

Also, as previously explained, words were used as tokens, in a process called Tokenization. In this process, a text sequence is broken up into small pieces (tokens), such as words, keywords, phrases or other relevant elements. With the same goal as in the stop-word process, special characters and punctuation were removed.

By using those steps and by applying them to all strings that were used, a vocabulary document that contains the extracted words as output vectors was created. Thus, each sentence was measured against the created list. This process may increase the value of each vector element which, as early explained, were used by the ML algorithms for classification and prediction purposes.

Despite being easy to use and to understand, and regardless of allowing great results in a quick way, the BoW model only counts word occurrence, despising the meaning and/or context of the sentence in which they occur. This can be problematic if the same BoW has multiple meanings in the classification methodology that is supposed to be achieved. Also, as the size of the non-classified text grows, the vector size will increase in size and can result in greater needs for computation time and power. After multiple tests, it was concluded that those limitations did not affect the final solution, since the custom BoW was projected for surpassing those same limitations since the beginning. The aim of this step is to get the final classification for each given text by using the created model. Having a higher prediction rate will allow to have less errors when predicting the content type, and in the end, allow the development of a fail-proof system. At the end of this process, all gathered articles are successfully sorted and classified as shown in the diagram depicted in Figure 3. In Figure 4, the Backoffice of the ADAA ecosystem is shown. Here, different categories and several keywords can be selected to perform a news search.

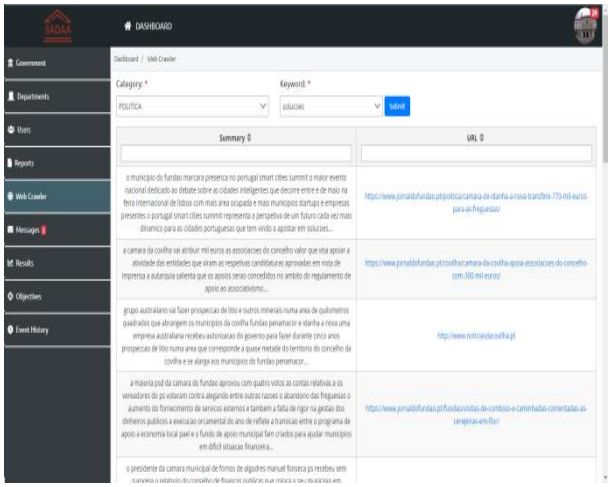

Figure 4: Search information feature implemented on the ADAA web decision and support tool for government entities

4. CONCLUSION

This paper depicted the work performed on the development of a new information search mechanism for the governmental sector. Thus, through the application of NLP and ML techniques, it was possible to categorize information gathered on web data sources, namely the classification of news articles on multiple topics, such as health, education, politics, among others. Web crawling mechanisms were used to gather the news articles from the web. First, the training dataset was constructed, which comprised 80% of the data gathered through the web crawling mechanisms, and which was already categorized. Then, the remaining 20% of the data was used for the test dataset, on which NLP and ML mechanisms were applied to find the most relevant words and to discover the context of the used text. Concepts such as BoW, TF-IDF and neural networks have been used in the process. In general, news were correctly categorized, with only some news about society (18%) and politics (11%) being considered for other categories. However, the overall precision of the categorization mechanisms was good. Some improvements could be implemented on the web crawling mechanisms, since the web sources used were static and could not be changed. Also, tests with bigger datasets could be performed in order to evaluate the performance of the crawling and classification mechanisms in more robust scenarios. The search information mechanism was implemented on the ADAA ecosystem, namely on a decision support tool for government entities and on a e-government application for citizens.

5. ACKNOWLEDGMENTS

This work is part of the ADAA project, co-funded by CENTRO2020/P2020/EU, in the context of the Portuguese Sistema de Incentivos à I&DT Empresarial (project 038255).

6. REFERENCES

[1] B. Marr, "How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read," Forbes, 2018. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#7a5fa33760ba. [Accessed: 25-Feb-2020].

[2] A. Felsberger, B. Oberegger, and G. Reiner, "A Review of Decision Support Systems for Manufacturing Systems," in The International Conference on Knowledge Technologies and Data-driven Business 2016 - i-KNOW 2016 (i-KNOW 2016), 2016, pp. 1–8.

[3] M. Imran and A. Tanveer, "Decision Support Systems: Creating Value for Marketing Decisions in the Pharmaceutical Insustry," Eur. J. Bus. Innov. Res., vol. 3, no. 4, pp. 46–65, 2015.

[4] N. Lei and S. K. Moon, "A Decision Support System for Market Segment Driven Product Design," in Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol.9: Design Methods and Tools, 2013, no. August, pp. 1–10.

[5] A. Bhatia, "A Frame Work for Decision Support System for the Banking Sector – An Empirical Study of State Bank of Patiala," Int. J. Comp. Tech. Appl., vol. 2, no. 5, pp. 1368–1378, 2011.

[6] A. Lessard, "The Top 5 Advantages of an Effective Performance Management Program," govloop.com, 2017. [Online]. Available: https://www.govloop.com/community/blog/top-5-advantages-effective-performance-management-program/. [Accessed: 03-Mar-2020].

[7] S. Garr, "Performance Management for Better Business Results," Bersin, 2011. [Online]. Available: http://www.bersin.com/News/Details.aspx?id=14208. [Accessed: 22-Mar-2020].

[8] N. A. Azemi, H. Zaidi, and N. Hussin, "Information Quality in Organization for Better Decision-Making," Int. J. Acad. Res. Bus. Soc. Sci., vol. 7, no. 12, 2018.

[9] H. Chen, D. Hailey, N. Wang, and P. Yu, "A Review of Data Quality Assessment Methods for Public Health Information Systems," Int. J. Environ. Res. Public Heal., vol. 11, pp. 5170–5207, 2014.

[10] L. Daunis, "Poor-Quality Data Imposes Costs and Risks on Businesses, Says New Forbes Insights Report," Forbes, 2017. [Online]. Available: https://www.forbes.com/sites/forbespr/2017/05/31/poor-quality-data-imposes-costs-and-risks-on-businesses-says-new-forbes-insights-report/. [Accessed: 15-Mar-2020].

[11] D&B, "Transparent Government Demands Robust Data Quality," D&B, 2009. [Online]. Available: https://www.dnb.com/content/dam/english/dnb-solutions/sales-and-marketing/transparent_government_demands_data_quality.pdf. [Accessed: 20-Mar-2020].

[12] I. MORI, "What do people want, need and expect from public services?," 2020. [Online]. Available: https://www.ipsos.com/sites/default/files/publication/1970-01/sri_what_do_people_want_need_and_expect_from_public_services_110310.pdf. [Accessed: 21-Mar-2020].

[13] A. Customer Satisfaction Index, "Benchmarks by Sector," ACSI, 2019. [Online]. Available: https://www.theacsi.org/acsi-benchmarks/benchmarks-by-sector. [Accessed: 17-Oct-2019].

[14] Y. Goldberg, Neural Network Methods for Natural Language Processing. 2017.

[15] D. Jurafsky and J. H. Martin, Speech and Language Procesing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, 2nd ed. 2009.

[16] P. Barba, "Machine Learning for Natural Language Processing," Lexalytics, 2019. [Online]. Available: https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing. [Accessed: 10-Mar-2019].

[17] M. S. Ahuja, J. S. Bal, and Varnica, "Web Crawler : Extracting the Web Data," Int. J. Comput. Trends Technol., vol. 13, no. 3, pp. 132–137, 2014.

[18] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural Language Processing: State of The Art, Current Trends and Challenges," no. August 2017, 2018.

[19] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Almost Unsupervised Text to Speech and Automatic Speech Recognition," in 36th International Conference on Machine Learning, 2019.

[20] A. Ali and M. Z. Amin, "Conversational AI Chatbot Based on Encoder-Decoder Architectures with Attention Mechanism," in Artificial Intelligence Festival 2.0, 2019, no. December.

[21] Hubtype, "Rule-Based vs AI Chatbots," Hubtype, 2019. [Online]. Available: https://www.hubtype.com/blog/rule-based-vs-ai-chatbots/. [Accessed: 25-Mar-2020].

[22] P. Bhavsar, "On Semantic Search," Medium, 2019. [Online]. Available: https://medium.com/modern-nlp/semantic-search-fuck-yeah-e371c0f639d. [Accessed: 15-Mar-2020].

[23] S. A. Abdul-Kader and J. Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems," Int. J. Adv. Comput. Sci. Appl., vol. 6, no. 7, pp. 72–80, 2015.

[24] M. Mnasri, "Recent advances in conversational NLP : Towards the standardization of Chatbot building," arXiv - Cornell Univ., 2019.

[25] S. Pandiarajan, V. M. Yazhmozhi, and P. P. Kumar, "Semantic Search Engine Using Natural Language Processing," in Advanced Computer and Communication Engineering Technology, vol. 315, no. November 2015, 2015, pp. 561–571.

[26] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," Inf., vol. 10, no. 4, pp. 1–68, 2019.

[27] A. Hassan and A. Mahmood, "Efficient deep learning

model for text classification based on recurrent and convolutional layers," Proc. - 16th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2017, vol. 2017-Decem, no. December 2017, pp. 1108–1113, 2017.

[28] R. S. Devi, D. Manjula, and R. K. Siddharth, "An Efficient Approach for Web Indexing of Big Data through Hyperlinks in Web Crawling," Sci. World J., vol. 2015, pp. 1–9, 2015.

[29] S. I. Mfenyana, N. Moroosi, M. Thinyane, and S. M. Scott, "Development of a Facebook Crawler for Opinion Trend Monitoring and Analysis Purposes : Case Study of Government Service Delivery in Dwesa," Int. J. Comput. Appl., vol. 79, no. October, pp. 32–39, 2013.

[30] L. Buitinck et al., "API design for machine learning software: experiences from the scikit-learn project," ArXiv, pp. 1–15, 2013.

[31] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), 2016, pp. 265–283.

[32] A. Gulli and S. Pal, Deep Learning with Keras. Packt Publishing Ltd, 2017.

[33] K. Sahin, "Introduction to Web Scraping With Java," ScrapingBee, 2019. [Online]. Available: https://www.scrapingbee.com/blog/introduction-to-web-scraping-with-java/. [Accessed: 08-Mar-2020].

[34] R. Gori, "Web Scraping the Java Way," Stack Abuse, 2019. [Online]. Available: https://stackabuse.com/web-scraping-the-java-way/. [Accessed: 24-Mar-2020].

[35] Jsoup, "jsoup: Java HTML Parser," Jsoup, 2020. [Online]. Available: https://jsoup.org. [Accessed: 09-Mar-2020].

[36] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," Remote Sens. Environ., vol. 62, no. 1, pp. 77–89, 1997.

[37] D. M. W. Powers and Ailab, "Evaluation : From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," J. Mach. Learn. Technol., vol. 2, no. December, pp. 37–63, 2007.

[38] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: A statistical framework," Int. J. Mach. Learn. Cybern., vol. 1, no. 1–4, pp. 43–52, 2010.