Machine Learning as a New Search Engine Interface: An Overview

Taposh Kumar Neogy[1], Harish Paruchuri[2]

[1]Assistant Professor (Accounting), Institute of Business Administration (IBA), National University, Rajshahi, **BANGLADESH**
[2]Department of Computing Sciences, University of Houston-Clear Lake, 2700 Bay Area Blvd, Houston, TX 77058, **USA**

ABSTRACT

The essence of a web page is an inherently predisposed issue, one that is built on behaviors, interests, and intelligence. There are relatively a ton of reasons web pages are critical to the new world, as the matter cannot be overemphasized. The meteoric growth of the internet is one of the most potent factors making it hard for search engines to provide actionable results. With classified directories, search engines store web pages. To store these pages, some of the engines rely on the expertise of real people. Most of them are enabled and classified using automated means but the human factor is dominant in their success. From experimental results, we can deduce that the most effective and critical way to automate web pages for search engines is via the integration of machine learning.

Keywords: Machine Learning, Search Engine Interface, Search Technology

INTRODUCTION

Search engines are used to look for pages on the worldwide internet. They are designed to provide the most accurate results in microseconds. Before the advent of search engines, getting the right information from the internet was rather almost impossible. A search engine can be defined as a software program created to search for websites using the words (or phrases) users assign as "search terms". When it comes to Search Engine Optimization, there is an even greater role played regarding helping web pages compete for ranks and frequency of appearance based on the same assigned search terms. Google—the most popular search engine in existence—made looking for information on the internet precisely simple. Today, a good number of search engines have been advanced with machine learning techniques, allowing them to systematically classify and rank web pages. Machine learning technology is a concept that can be applied in a variety of fields related to website and web page ranking. Machine learning does not only give search engines a new interface but also enables them to handle different internet-based tasks automatically and effectively.

REVIEW OF RELATED LITERATURE

Web page ranking algorithm is a commonly known approach for web ranking in the cyber hemisphere. This algorithm helps users know exactly how the search engine operates. It

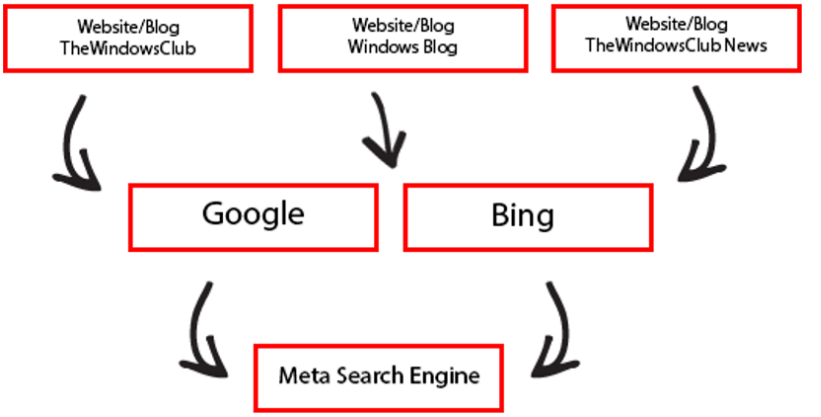
also gives us an idea of how a machine can learn itself while giving page priorities. The process is crucial to fulfilling the query of the user, who is then able to determine which pages are worth more to their research. With machine learning, we realize the complexity of page priority criteria in the best-used search engines. Page ranking is the system that shows us the way the internet is structured (Lakshmi Narayana et al., 2012). The web is choked with valuable resources for different types of people and different kinds of needs. That's the major reason online places like Google, Bing, and Ask have become an increasingly widespread way of sourcing information.

Users can upload images on a search engine to get information about it and find similar images (Adrakatti et al., 2016). Lots of data exist in image form on the internet, which means search engines can use machine learning to power-search for information about the images and get more actionable results than usual. Search engines are also designed to classify queries into various categories (Donepudi, 2014). They need to determine whether the qwerty is transactional, navigation, information-based, or under the other categories.

Through machine learning technology, the categories for the queries can be effectively identified. Depending on the category of the queries, the search engine may give the appropriate additional information. For example, if you search for "Westminster Abbey", you will get its information and also a Google Maps location for the same. Internet searches are either factorized based either on their keywords or their sentences (Wilson and Pettijohn, 2006). The task for a user is formulating a moderate set of phrases or keywords that will enable the search engine to pinpoint the most related results.

Irrespective of the kind of specification invested into the search query, not all the tens of millions of pages are going to be relevant to the user. Having to check one after the other to find the desired outcomes is time-consuming and laborious. The race for web users is now largely based on providing accurate and precise search results whenever called upon. The most-used search engines on the web have implemented some ideas to make information readily and easily accessible to the users. The providers are moving to make the results more direct and useful for those looking.

According to the paper (Donepudi, 2014), the creation of a metasearch engine known as SEReleC will result in an interface with which search engines will be refined and classified. By doing so, the search process—especially the results stage—will be narrowed into a sequentially linked fashion. This will culminate in a significant reduction in the number of web pages' users have to sift through.



Internet use is on an incredible climb, making "online" a crucial source of Intel about industries, companies, places, and cultures, among many other topics. As such, the role search engines play as an effective way to locate information of internet users becomes essential. A reasonable amount of studies has focused on the behavior of search engine users, and they are all useful in the development of more reliable search engines and viewpoints. These studies are most useful for users at the personal level, for governmental process and social-level marketing strategies. Such studies can be carried out through the analysis of the log file of search engines—where relations between the users and the engines are stored (Ujwala et al, 2012).

When it comes to looking for satisfying information online, the metasearch engine is a handy tool. Juxtaposed with self-sufficient search engines like Google and Bing, metasearch engines are capable of more extensive coverage. They are also better at fulfilling the requirements for retrieving the searched-for information, the moment a metasearch engine receives a query from a user, it will transfer the query to proper candidate member engines. The results realized from these engines will then be used to respond to the user. The most crucial task here is how to effectively choose the underlying search engines, obtain the results and get them back to the curious user.

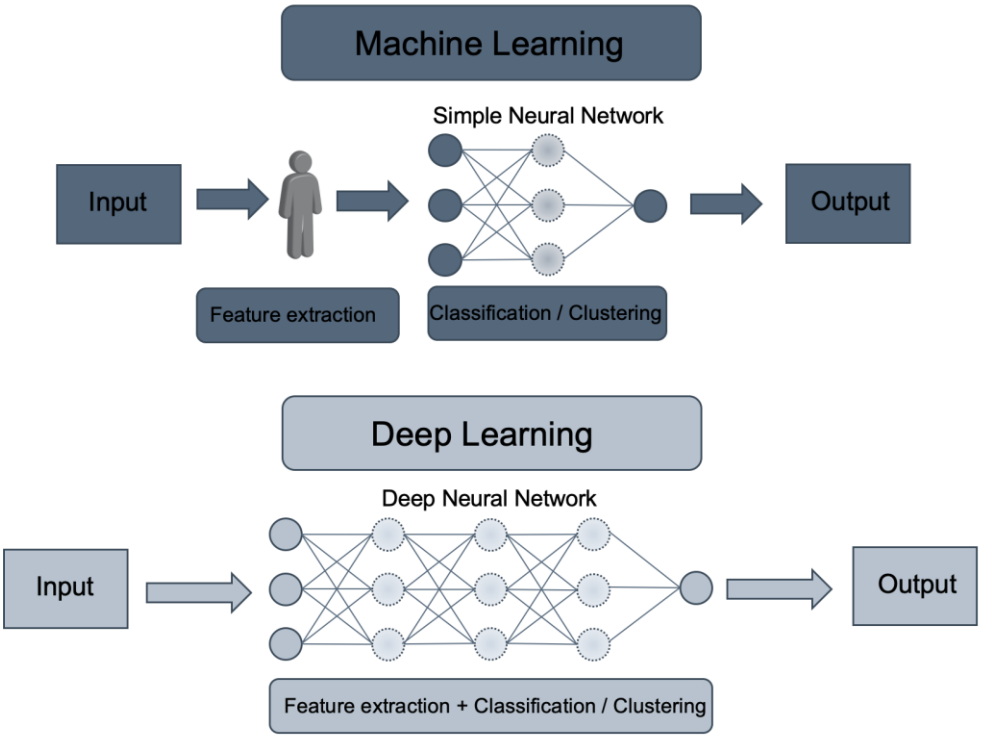
The internet-based smart tech advancements of today—such as wearable-tech and mobile phone apps—have enabled people to monitor things happening around them. For instance, some of these innovative devices have been used to monitor health statuses and infection cases systematically. Even before the advent of the coronavirus pandemic, researchers report that nearly 59 percent (Ujwala et al, 2012) of adults in the United States searched for health information on the internet. Nevertheless, the number went up to 75 percent recently, recording more than a billion Google searches for health-related information on the engine daily. Undoubtedly, people are depending more on these search engines for all their queries. Should those queries be tailored with machine learning algorithms, much value would be offered to the users?

RESEARCH METHODOLOGY

This research considers various machine learning techniques and puts them into a demonstration system. We turn it into a domain-specific engine in computer science we named Lofgren. The system is supposed to provide keyword search initiatives for more than 10,000 collected papers. The system also places the papers into the topic hierarchy of computer science, maps out the citation links between the papers, and provides bibliographic data about every single paper considered. Logically consequently, in addition to providing a platform for testing machine learning-based research, Lofgren will become a tool of value for other computer scientists.

Hopefully, it will complement similar types, like the Computing Research Repository. This it will do by providing functionality and coverage that cannot be gotten anywhere else on the World Wide Web.

The construction of a search engine can be split into three strategies of functionality. The first stage involves the collection of new information. At the second stage, the information undergoes an extraction process. The third and final stage is when the information extracted is presented in a publicly available web interface. Lofgren completes each of these stages by leveraging machine learning techniques described in this paper.



The first stage involves the sourcing and collection of research papers in the computer science field. When the spider crawls across the internet, from the home page of the computer science departments and laboratories. With the aid of reinforcement learning, the spider can collect the entire postscript documents that are needed by the engine. The second phase involved in the formation of a search engine is the extraction of relevant knowledge from every paper used on the platform. This stage passes the beginning of each paper—from the abstract—through an information extraction system. This system automatically finds the author, title, institution, publication date, and other header details.

Additionally, the information extraction stage locates the bibliography section of each paper. It also finds the identified individual references, all of which are broken down into suitable Eds. That's where information about the author(s), title, journal, and year of research come from. The third stage of building a search engine involves the provision of a user interface that is publicly available on a worldwide basis. This is where machine learning algorithms come into play.

For this, two methods are implemented to enable the search engines to easily locate papers. The first method involves the provision of a search engine over all the papers. This makes it easy to syntax-search for queries with signs like + and -. Just as commonly used is phrase-searching with ". The resulting matches for these queries will be ranked by the weighted log of term frequency accumulated across all the query terms. At this stage, the search engine is also designed to allow searches restricted to extracted Eds. Consequently, the response to queries is often delivered in less than a second. Each paper comes with a page for details about the relevant information—like title, author(s), and links to postscript papers, and traversable citation maps.

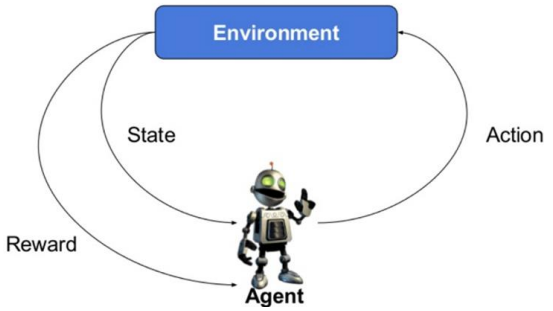
As for the second user interface in the third stage of a search engine build, the access method involves a topic hierarchy. This feature resembles what Yahoo provides, but has been specifically customized to research the science of computers. The handmade topic hierarchy contains 50 leaves, with varying depths of one to three. With text classification methods, each paper considered is systematically put into a topic node. The most cited papers in each research topic can be located and accessed by following the hyperlinks to traverse the topic hierarchy.

REINFORCEMENT LEARNING

In machine learning, reinforcement learning references a framework for learning optimal decision making for punishment or rewards (Kaelbling et al., 1996). Reinforcement learning differs from supervised learning because the learner in question is never told the correct action for any stage. The learner is only told how good or bad the selected option will be, expressing the results in a scalar reward format.

Each task is defined by a set of states, a set of actions, a state-action transition function, and a reward function. At each time step, the learner opts for action and is given a reward as a result, as well as the new state. The reinforcement learning technology aims to get used to a policy that is mapped from states to actions, all of which help to maximize the sum of rewards over a given time. Of all the repeated formulation of rewards over time, the most common is a discount sum of reward infinitely into futures. A discount factor, in question, is used (A. Lepetit, 2017) to express inflation, enabling the learner to make earlier rewards more valuable compared to later rewards.

Typical RL scenario



In search engine creation, spiders are agents built to explore the hyperlink graphs of the internet (Boyan et al., 1996). They are often used to find papers with which the search engines themselves will be populated. Through extensive spidering, a substantial coverage of the major search engines is obtainable. Online places like HotBot and AltaVista will be considered.

Given that the aim of general-purpose search engines is the provision of holistic web search capabilities, their simple goal—for the most part—is finding as many different web pages as possible in a split second. To understand how reinforcement learning is related to spiders, one needs to consider the common reinforcement learning task of a mouse crawling through a maze to locate several cheese pieces. The actions of the agent move among the grid squares of the maze, after which the agent is rewarded for finding each piece.

Such a process leads to strategies such as breadth-first search. On the other hand, supposing the task is to populate a domain-specific engine, an intelligent spider needs to

steer clear of hyperlinks leading to topic areas. Instead, they concentrate on finding and providing the links that lead to the papers of more interest. For Lofgren, accurate and effective spidering demands huge consideration. A good number of the pages in lots of computer science websites do not provide links to the research papers a user may be looking for. Instead, they are focused on online courses, admissions information, and homework. By avoiding branches and sub-branches of departmental websites, we can significantly enhance the efficiency and increase the count of research papers found. Although, the search engine needs to be given the right amount of time to find these papers. With learning reinforcement—which is an offshoot of the machine learning technology—efficient spidering can be performed.

There are several other system studies about web spidering. However, none of them have a framework that defines optimal behavior. According to Arachnid (Menczer, 1997), there is a collection of competitive, reproducing, and mutating components whose tasks are locating information on the internet. Cho et al. (1998) posit that there are several heuristic ordering metrics search engine users consider when choosing the link to crawl next when searching for web page categories. In addition, systems have used reinforcement learning for tasks unrelated to web spiders. WebWatcher (Joachims, Freitag, & Mitchell, 1997), a browser assistant, uses a blend of supervised and reinforcement learning to assist users with information by recommending the correct hyperlink to the categories being searched for.

APPROXIMATION

When it comes to building such search engines with the machine learning interface, how to apply reinforcement learning to spidering in a practically solvable way can be a problem. The challenge is; the state space is colossal, with two to the power of the number of on-topic research papers on the internet. In fact, the number of unique URLs (also known as the action space) with incoming links on the internet. As such, simplifiable assumptions need to be made to make the issue tractable, from where it will contribute to generalization. Be as it may, by defining the precise solution of the optimal policy and assuming explicitly, one would be able to better comprehend the inaccuracies that were introduced. One would also better understand how to select the areas of future work that will contribute to the improvement of the overall performance of the search engine.

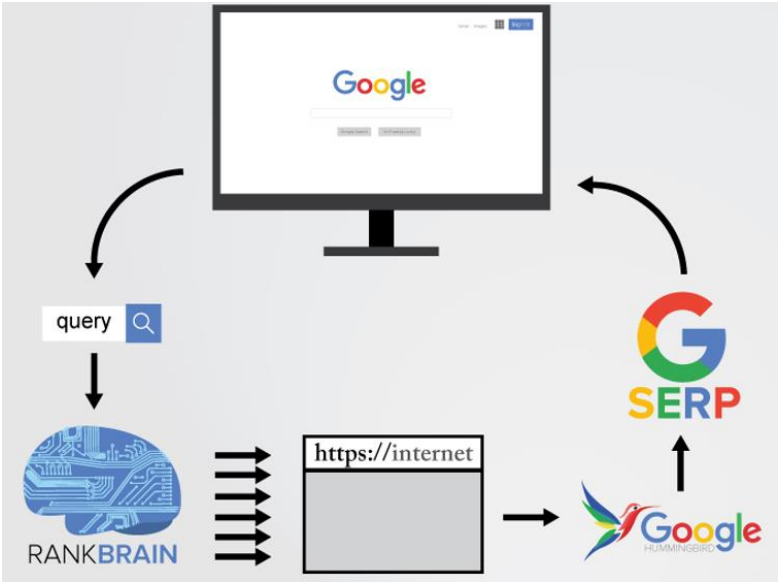
The first assumption made was that the state is not dependent on which on-topic papers have been consumed already by the user. That means all the states need to be collapsed into a unified whole. The second assumption made was that the relevant distinctions between actions on the system can be captured by the words around the hyperlink that corresponds to each section. The Q function becomes a mapping (Y Ren, 2013) that begins from "a bag of words" to a scalar. Efficient web spidering for machine learning-interfaced search engines involves only two other minor problems.

Applying reinforcement learning can also be hamstrung by the gathering of training data—which consists of a "bag of words", and future reward pairs. The last problem is mapping the interface with the training data obtained. Several means exist for tracking and collecting training information. Even though the agent can learn from experience online, they can be trained offline with collections of pre-discovered hyperlinks and research papers. In reinforcement learning lingo, this implies that the state transition function and the reward function are known. It means the Q function can be learned by dynamically programming in the collapsed, original state space (Riedmiller, 2005).

The mapping is represented with a collection of naive Bayes text classifiers. The mapping is performed by casting the regression issue as classification (Torgo & Gama, 1997). The discount sum of the future reward value of training information is discretized, after which the hyperlinks are placed into the bin that corresponds to their respective Q values. That is also carried out with the aid of dynamic programming. The neighborhood of a hyperlink can be defined into two bags of words. The first is the full text of the page where the hyperlink is located. And, the second is the anchor text of the hyperlink and portions of the said URL. For every hyperlink involved, the probabilistic class membership of each bin needs to be calculated. Then, the hyperlink's reward value is estimated by getting a weighted average of the reward value in the bins. This is done with the probabilistic class memberships.

GOOGLE RANKBRAIN

Google uses a machine-learning algorithm known as RankBrain to sort the search results it presents to users. With RankBrain, Google is also able to process and interpret search queries accurately. RankBrain is the third most important rank signal for the Google search engine, meaning its importance grows by the day.



Before RankBrain, Google was accustomed to scanning pages in search of the exact keywords a user searched with. But now, RankBrain, via machine learning, understands what users are looking for and provides them with completely accurate result sets. RankBrain is designed to figure out what users would like to see. Case in point, Google took note that plenty of people searching for "grey console developed by Nintendo" want to see a set of results that show gaming consoles. So when a user searches with such a query, RankBrain will bring up similar results for the keyphrase. All this is possible because machine learning has been woven into the fabric of search engine optimization.

FOR SEARCH QUERIES

Anytime a user wants to type in their question on the search box of a search engine, the most important thing for the search engine itself is figuring out what has been asked. Should the engine be unable to understand what the user is trying to ask, it might provide

inaccurate results (Movva et al., 2012). Likewise, a search engine will not be able to give the user appropriate answers if it does not understand the search query in the first place. Should any of this happen, the search engine might become more or less useless. This is where technologies like machine learning come into the picture.

Users are prone to spelling errors while typing out their queries in those search boxes. Being that not every user will type in their query correctly, machine learning can take care of the rest. So if a user spells a search query wrong, the search engine will display the correct spelling of the same phrase, with which you can make better web searches. A machine learning-supported search engine should be smart enough to identify the right spelling of the same word and provide it to the user before they make their search. Even if the user searches with a wrongly spelled query already, the search engines will correct it automatically before the first results page comes up.

OTHER APPLICATIONS

A project called WebKB (Craven et al. 1998) sheds the limelight on the collection and organization of data from the internet on an intelligence basis. The project strongly emphasizes the use of machine learning techniques like information extraction and text classification. These techniques are meant to promote easy reuse across the domains involved. To this, domains in the computer science departments and companies have been created. One called the CiterSeer project (Bollacker et al., 1998) is the development of a search engine for research papers in computer science. This engine provides similar functionality for the search and link of documents. Nonetheless, the domain does not yet provide a hierarchy of the eld.

CiteSeer is designed to focus on the domain for research papers. Nonetheless, it is not as focused as using machine learning techniques to automate the creation of a search engine. The New Zealand Digital Library project (Witten et al, 1998) has brought publicly available search engines for domains. This can vary from song melodies to computer science reports.

The emphasis was placed on this project owing to the creation of full-text searchable digital libraries. The creation here is not done with machine learning techniques that can be used to generate repositories autonomously. The web source for their libraries can be identified manually and no top-level component of the data is provided. Likewise, there is no extraction of information for the repositories and the citation links are not provided. Another example is project WHIRL, an effort to integrate various topic-specific sources into a single domain-specific search engine. Here, two demo domains, one for computer games and the other for North American birds were integrated from many sources. The reason for such an emphasis is the provision of soft-matching for information retrieval searching. The data is extracted from the internet's pages by handwritten means customized for each web source available. This research (Cohen and Fan, 1999) into WHIRL learns general wrapper extractors from its examples.

SUMMARY AND CONCLUSION

Web page ranking is a global scoring of web pages, irrespective of their various contents. The ranking is done based on the web page's location in the internet graph structure. With those web page ranking techniques, one can get research results, which gives preference to more important and central pages.

Ceteris paribus, search engine optimization will grow into a more resourceful tool in the coming years. It will also become more complex, leaving marketing little to no choice but to elaborate strategies to bring more content types, devices, and tools. However, regardless of the kind of elements one combines, the focus needs to be on the users and their needs. Hitherto, machine learning and artificial intelligence technologies will transform the factors considered for web page ranking and provide a better reflection of the needs and expectations of the users.

The information on the billions of web pages on the internet is showing exponential growth. As the amount grows, we opine that not only is the public in need of powerful tools to help filter through countless information streams. The creators of these tools also need some intelligence-based techniques that will enable them to create and maintain such search engine services. With this paper, we have been able to demonstrate that machine learning techniques have a huge role to play in the creation and maintenance of high-profile domain-specific search engines. The research into reinforcement learning, extraction, and text classification is provided towards this end.

A great deal of future work in every area of machine learning has been looked into already. Be as that may, we see a lot of other areas where machine learning can be used for the additional automation of domain-specific search engine build and maintenance. Case in point, text classified has the capability to decide which web papers are relevant enough to be a part of the domain. Unsupervised clustering can create a topic hierarchy automatically and just as well generate keywords. Citation graph analysis, on the other hand, helps in the identification of Seminoles papers. Our anticipation is that a suite of many machine learning techniques will be developed for the creation of domain-specific search engines to be done quickly, easily, and effectively.

REFERENCES

Bollacker, K. D.; Lawrence, S.; and Giles, C. L. (1998). CiteSeer: An autonomous web agent for automatic re- trieval and identification of interesting publications. In Agents '98, 116{123.

Boyan, J.; Freitag, D.; and Joachims, T. (1996). A machine learning architecture for optimizing web search engines. In AAAI workshop on Internet-Based Information Systems.

Cho, J.; Garcia-Molina, H.; and Page, L. (1998). Efficient crawling through URL ordering. In WWW7.

Cohen, W., and Fan, W. (1999). Learning page-independent heuristics for extracting data from web pages. In AAAI Spring Symposium on Intelligent Agents in Cyberspace.

Donepudi, P. K. (2014). Voice Search Technology: An Overview. *Engineering International*, *2*(2), 91-102. https://doi.org/10.18034/ei.v2i2.502

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. (1996). Reinforcement learning: A survey. Journal of Artificial Intelligence Research 237(285).

Lakshmi Narayana S., Suneetha Devi J., Bhargav Reddy I., Harish Paruchuri. (2012). Optimizing Voice Recognition using Various Techniques. *CiiT International Journal of Digital Signal Processing*, *4*(4), 135-141

Menczer, F. (1997). ARACHNID: Adaptive retrieval agents choosing heuristic neighborhoods for information discov- ery. In ICML '97.

Movva, L., Kurra, C., Koteswara Rao, G., Battula, R. B., Sridhar, M., & Harish, P. (2012). Underwater Acoustic Sensor Networks: A Survey on MAC and Routing Protocols. *International Journal of Computer Technology and Applications*, *3*(3).

Riedmiller, M. (2005). Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method.

Torgo, L., and Gama, J. (1997). Regression using classification algorithms. Intelligent Data Analysis 1(4).

Ujwala, D., Ram Kiran, D. S., Jyothi, B., Fathima, S. S., Paruchuri, H., Koushik, Y. M. S. R. (2012). A Parametric Study on Impedance Matching of A CPW Fed T-shaped UWB Antenna. International Journal of Soft Computing and Engineering, 2(2), 433-436.

Wilson, R. F.; Pettijohn, J. B. (2006). Search engine optimisation: A primer on keyword strategies. J. Direct Data Digit.

Witten, I. H.; Nevill-Manning, C.; McNab, R.; and Cunnningham, S. J. (1998). A public digital library based on full-text retrieval: Collections and experience. Communications of the ACM 41(4):71{75.

--0--

Source of Support: Nil, **No Conflict of Interest:** Declared