International Journal of Advanced Trends in Computer Applications
www.ijatca.com

Performance Analysis of Named Entity Based Search Engine using SVM and Neural Network

[1]Anant Kumar Malani
ME Research Scholar Computer Science & Engineering
University Institute of Engineering and Technology,
Panjab University, Chandigarh, India.
amalani83@gmail.com

[2]Dr. Mukesh Kumar
Assistant Professor Computer Science & Engineering,
University Institute of Engineering and Technology,
Panjab University, Chandigarh, India.
mukesh_rai9@yahoo.com

Abstract: *Keyword search has been proven an effective information discovery method for unstructured data (e.g. textual documents), semi-structured data (e.g. XML databases), and structured data (e.g. relational databases). For semi-structured and structured data, keyword search allows users without prior knowledge of schema and query languages (e.g. SQL for relational databases and Query for XML databases) to exploit the data, But in this paper, we will study how entity based search engines different than keyword based search engines along with their disadvantages and working*

Keywords: *keyword; search engines; entity; google.*

1. Introduction

Web crawlers have promoted watchword based hunt. Clients submit essential words to the web index and a positioned rundown of records is come back to the client. A distinct option for pivotal word inquiry is organized pursuit where clients coordinate their hunt by perusing order pecking orders [1]. Both models are colossally profitable – accomplishment of both magic word look and the characterization pecking order are obvious today. A lot of the world's endeavor information lives in social databases. It is vital that clients have the capacity to flawlessly pursuit and skim data put away in these databases also. Looking databases on the web and intranet today is basically empowered by tweaked web applications nearly attached to the mapping of the basic databases, permitting clients to direct inquiries in an organized way [2]. Samples of such pursuits inside, say a book shop's database may be "Books → Travel → Lonely Planet → Asia", or "Books→ Travel → Rough Guides → Europe". While such organized inquiries over databases are most likely helpful, dissimilar to the reports world, there is little backing for magic word look over databases. Yet, such a hunt model can be amazingly effective [3]. Case in point, we may like to hunt the Microsoft intranet on 'Jim Gray' to acquire coordinated lines, i.e., pushes in the database where 'Jim Gray' happen. Note that such coordinated columns may be found in more than one table, maybe even from diverse databases (e.g., location book and mailing records) [4].

Empowering magic word seek in databases that does not oblige learning of the outline is a testing undertaking. Note that one can't have any significant bearing methods from the archives world to databases in a clear way. For instance, because of database standardization, intelligent units of data may be divided and scattered over a few physical tables [5]. Given an arrangement of pivotal words, a coordinating line may need to be acquired by joining a few tables on the fly. Besides, the physical database plan (e.g., the accessibility of records on different database segments) needs to be utilized for building smaller information structures basic for effective magic word look over social databases [6].

2. Limitations of Keyword based Search Engines

Conventional search engines are very useful in finding data over internet [7]. From survey it has been found out that 25% of data that has been searched out is not accurate according to the algorithms used. In addition to their ability of high time consumption and not accurate results made them obsolete. Also one word has several meaning [8]. So some of the problems of traditional search engines has been mentioned below [9]:

- One word having various meanings
- Several words have various meanings.
- The availability of large dataset.
- Low precision and recall rate.

3. Is google an entity based search engine

Path once upon a time, Google positioned sites construct essentially with respect to the essential words found on the real page. On the off chance that you stuffed a cluster of magic words on the page, you'd wind up at the highest point of the list items [11]. They started utilizing connections as their essential positioning element. A connection was a "vote" and the more votes you got, the better positioning you'd see. Today, Google utilizes more than 200 "signs" in their web crawler positioning calculation. That incorporates joins, social offering, magic words, substance and a great deal more. At the same time, where is Google heading? What does the future look like for Google seek? As indicated by Amit Singhal, a Google Fellow and lead on Google Search, they're heading profound into the universe of elements. What's an element, you ask? An element is basically a solitary thing or idea that exists on the planet, as indicated by the Freebase site. Google procured Meta Web in 2010 and Freebase was a Meta Web venture. Meta Web made an arrangement of indexing named substances to make it simpler to recognize, interface and quest for references about that named substance. Google needs to take that framework much further and utilization elements to make query items that can recognize connection to issue you better indexed lists. Fundamentally, they are making a gigantic learning chart of interconnected substances and the properties connected with those elements [12].

4. Entity Search Problem

Given: An entity collection R = {R1; R2; : : : ;RM }
Input: Query: β – α (L1;L2; : : : ;Ll;R1;R2; : : ;Rn)
Output: t = {r1; r2;:::; rno: sorted by score (t), the tuple score of t with respect to β and α.

As input, an entity serach query is like standard keyword inquiries, however now clients can indicate entity sorts, R1, R2, : , and Rn, with addition to keywords L1, . . . , Ll. Alternatively, in a complete structure, an inquiry can likewise indicate a matching example α (e.g., ow in Q3), to confine when an event of an occurrence t = he1; e2; :::; emi is viewed as a matching tuple, and a scoring measure β, to determine how all the matching cases are positioned. Contingent upon the usage and application settings, the decisions of the β design and α can be framework constructed in or client indicated, and they together focus the positioning scores [13].

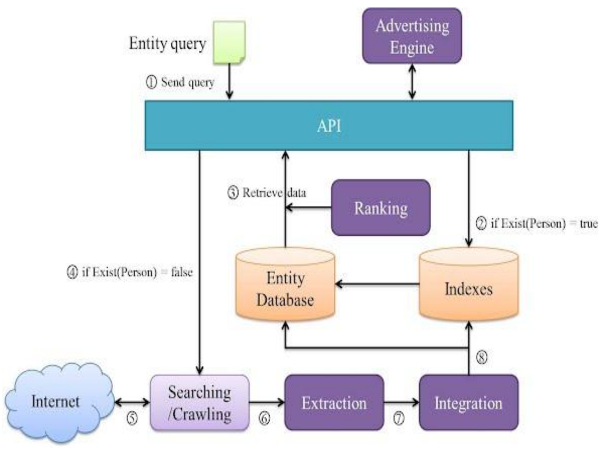

Figure 1: Entity Search Engine [14]

Named entity recognition is now firmly established as a key technology for understanding low-level semantics of texts. Its main role is to identify expressions such as date and time as well as names of people, places, and organizations. Those expressions are difficult to extract using traditional natural language processing (NLP) because they belong to the open class of expressions, i.e. there is an infinite variety and new expressions are constantly being created. Automatically extracting proper names is useful to many problems such as machine translation, information retrieval, information extraction, question answering and summarization. Especially named entity recognition play an important role in extracting answers candidates for question answering. The goal of named entity recognition is to classify names into some particular categories from text.

5. Proposed Work

Methodology

Named entity recognition (NER) is a subtask of data extraction that tries to find and arrange atomic component into content using predefined classifications, for example, the names of persons, associations, areas, articulations of time, amounts, financial qualities, rates, and so forth.

In the wake of improvements in investigation of technical data the amount of interrelated information is growing exponentially. In this way how to get related technical data from a lot of writing specifically is turning into a matter of great urgency. To perceive named entity from technical literature, for example, display no., configuration, RAM and so on turns into a fundamental step of data extraction in technical data [24].

Technical named entity is unique in relation to general NER. To start with the new named entity continues developing. Therefore, it is difficult to develop a complete word reference contains different sorts of technical named entity. Second numerous technical named entities are multi-word expresses; which makes qualities make it hard to focus the boundaries of named elements. Thirdly, same words have different meanings [25]. Fourthly, technical words that are connected by "and" and "or" are very difficult to find. So, excess of abbreviations lead to difficulty in item identification.

Named Entity Extraction Model

- Pre-processing: In the beginning, the input text strings ought to be divided into tokens with a basic tokenizer.
- Feature Selection: At this stride, rich features are chosen to encode the data of every word.
- Mask Method: The mask method delivers all the training occasions for the classification step.
- Representation: This stride is utilized to represent to every training illustration to learn. As indicated by the chosen features, the entire training data can be encoded as an arrangement of vectors.

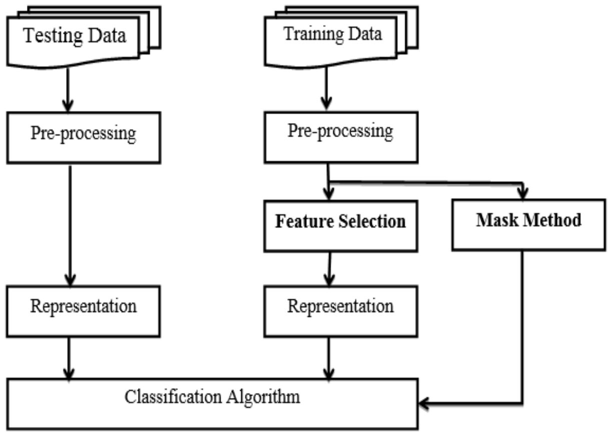

Figure 2: Proposed Flowchart

Classification Algorithm: The classification algorithm naturally figures out how to order testing information from the created training set at past stage. As of now, there are numerous classification methods and we select one of the well-known strategy. Then again, the chosen features additionally influent on the system performance. In addition to this it can be divided into two parts as following:

Feature Set

Textual information is the key point in getting the features but sometimes there is related elements that become features and they are described as below:

- **Previous NE information**
- **Possible NE classes:** It comprises the probable named class as features.

In genuine, training data is inadequate, since just a subset of the vocabularies can show up in the testing data. During testing, if a term is an obscure word (or one of its connection words is obscure), then the lexical related elements, as unigram, and bigram are disabled, because the term data is not found in the training data. For this situation, the name class of this word is mostly dictated by the remaining components. More often than not, this will abominable the system performance. The most widely recognized way for taking care of obscure word issue is to utilize distinctive feature set for obscure words and separating the training data into a few sections to gather obscure word illustrations.

Classification

The classification will be done using SVM and NN.

Classification using neural network:

```
a) Algorithm for training phase:
Input: network, training set
Step1: Setup NN and initialize the following
        parameters as: number_of_layer;
        epochs; learning_rate;
        permissible_error;
Step 2: do
        for each word in training set
            Extract its features;
            Fuse the extracted features into a single features matrix;
    Until a single feature vector matrix is built;
Step 3: do
        train the network about class labels and feature vectors;
    Until stopping criterion epochs is satisfied
output: a trained neural network.

b) Algorithm for testing phase:
input: a text stream.
Step 1: load the input text stream;
Step 2: extract its features;
Step 3: load the fused features database;
Step 4: compute similarity between text features and training set features;
output: set of similar text if present
        stop
```

- **Classification using SVM**

Support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.

6. Result and Analysis

Figure 3 shows the signal to noise ratio performance for entity based search for neural and support vector machines and neural is having high signal to noise ratio which should be high and SVM is having less signal to noise ratio than neural

$$\mathrm{SNR}_{db} = 10\log_{10}(SNR) \qquad (1)$$

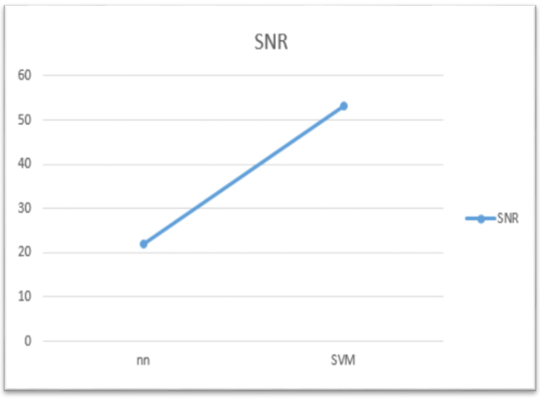

Figure 3: Signal to Noise Ratio

False Acceptance Rate

Figure 4 shows the False Acceptance Rate using neural network and support vector machine and shows that the FAR is having less measure using neural than FAR with Support vector machine. This measure should be less for high accuracy of entity based search process.

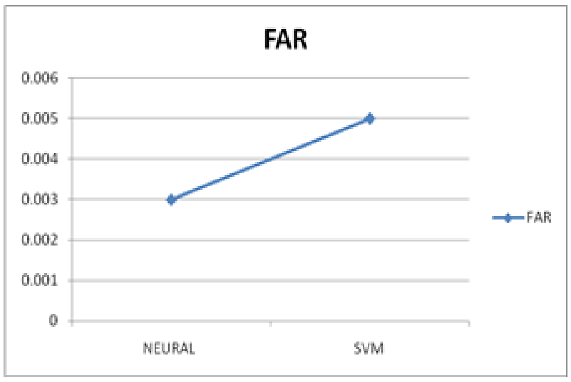

Figure 4: False Acceptance Rate

False Rejection Rate

False Rejection Rate: It is the probability that an identified value is a true positive.

$$\text{Positive Predictive Value} = \frac{TP}{TP+FP} \qquad (2)$$

False Positives (FP) are those values in database which are actually belongs to another category like T1 but have been detected by the algorithm as for category T2.

True Positives (TP) are those values in database which are belongs to category like T1 and have been detected by the algorithm as well for category T1 too.

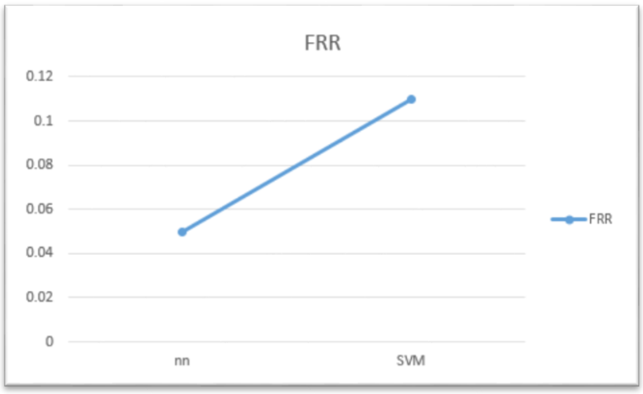

Figure 5: False Rejection Rate

Figure 5 shows the FRR measurement on the basis of search using neural and support vector machine and shows that the FRR in neural is having low value than the FRR using support vector machine.

Precision

Figure 6 shows the precision measurement on the basis of search using neural and support vector machine and shows that the precision in neural is having higher value than the precision using support vector machine.

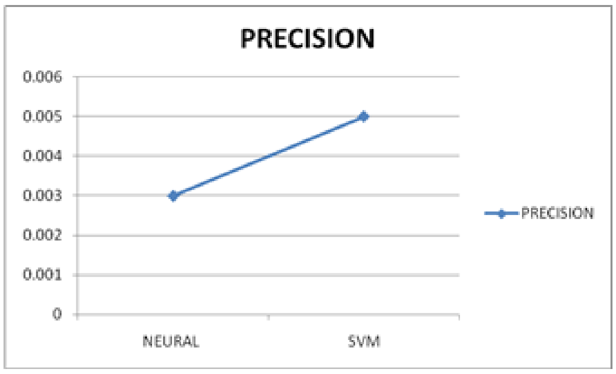

Figure 6: Precision

Precision rate (p) is defined as the ratio of correctly detected categories to the sum of correctly detected categories plus false positives.

$$p = \frac{\text{correctly detected categories}}{\text{correctly detected categories} + FP} \qquad (3)$$

Recall Rate

Figure 7 shows the Recall Rate measurement on the basis of search using neural and support vector machine and shows that the Recall Rate in neural is having high value than the FRR using support vector machine.

Recall rate (r) is defined as the ratio of the correctly detected categories to sum of correctly detected categories plus false negatives.

$$r = \frac{\text{correctly detected categories}}{\text{correctly detected categories} + FN} \quad (4)$$

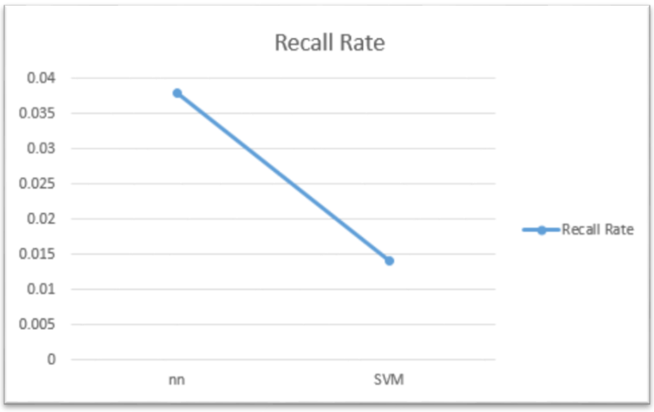

Figure 7: Recall Rate

Error Rate

It is the subtraction of total unrecognized categories from Total number of categories. Figure 4.6 shows the Error Rate using NN and SVM.

$$\text{Error Rate} = \frac{\text{Unrecognized categories} - \text{Total no.of categories}}{\text{Total no.of categories}}$$

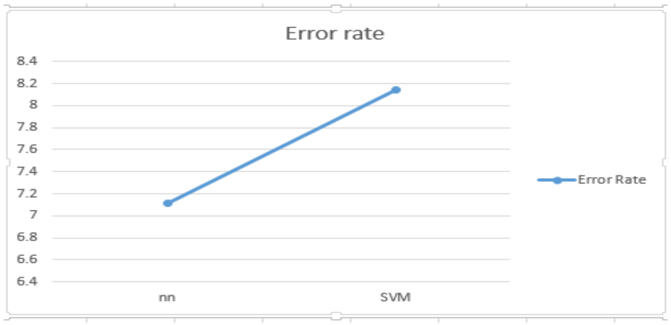

Figure 8: Error Rate

Table.1 Performance Comparison Table

Parameters	SVM	NN
SNR	53.44	22
FAR	1.17	13.29
Precision	.78	.086
FRR	.11	.05
Recall	.014	.038
Error rate	8.14	7.14

7. Conclusion and Future Scope

The role of named entity recognition is to classify names into some particular categories from text by machine learning or statistical method. Support vector machine and neural network are the power tool in machine leaning was widely used in the text categorization for named entity recognition. SVM and NN has been evaluated on the basis of Error rate, FAR, FRR, SNR , Precision and Recall rate to recognize the entities using training algorithm. From the results of the practical problem with support vector machine, we can clearly report the good general performance of the neural network machine has been achieved in comparison to SVM.

Although best efforts have been made but there is a lot of scope for improvement in the work carried out here. In future the work can be extended in which data mining methods can be used like K-Means clustering. *K-means* clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *K-means* clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

References

[1] D. Tumer, M. A. Shah and Y. Bitirim "An Empirical Evaluation on Semantic Search Performance of Keyword-Based and Semantic Search Engines: Google, Yahoo, Msn and Hakia," Fourth International Conference on Internet Monitoring and Protection, pp. 51-55, May 2009 .

[2] M. Tang and Y. Sun, "Evaluation of Web-Based Search Engines Using User Effort Measures," Library and Information Science Research Electronic Journal 13(2), 2003.

[3] M. Andago, P.L Phoebe and A. M Thanoun, "Evaluation of a SemanticSearch Engine against a Keyword Search Engine Using First 20 Precision," In Proceedings of the 29th Annual International Conference on Research and Development in Information Retrieval, ACM Press, pp. 735–746, 2010.

[4] K. C.-C. Chang, B. He, and Z. Zhang, "Toward Building a meta querier: extracting and matching web query interfaces". In Proceedings CIDR, pp. 1098-1099, April 2005.

[5] W. W. Cohen, "Some practical observations on integration of web information", In WebDB (Informal Proceedings), pp. 55–60, 1999.

[6] S. Raghavan and H. Garcia-Molina, "Crawling the hidden web", In VLDB, pp. 129–138, 2001.

[7] L. J. Seligman, A. Rosenthal, P. E. Lehner, and A. Smith, "Data integration: Where does the time go"? IEEE Data Eng. Bull., 25(3):3–10, 2002.

[8] Z. Zang, B. he and K. C. C. Chang, "Light-weight domain-based form assistant: Querying Web databases on the fly", In Proceedings of VLDB 2005.

[9] Pablo N. Mendes, Max Jakob, Andres Garcia-Silva and Christian Bizer, "DBpedia spotlight: Shedding light on the web of documents", Proceedings of the 7th International Conference on Semantic Systems (ISemantics), 2011.

[10] Axel-Cyrille Ngonga Ngomo, Norman Heino, Klaus Lyko, Rene Speck and Martin Kaltenbock, "SCMS –

Semantifying Content Management Systems", The Semantic Web–ISWC 2011, Springer, pp. 189–204, 2011.

[11] Sameer Singh, Amarnag Subramanya, Fernando Pereira and Andrew McCallum, "Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical models", Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, Association for Computational Linguistics, pp. 793–803, 2011.

[12] Nadine Steinmetz, Magnus Knuth and Harald Sack, "Statistical Analyses of Named Entity Disambiguation Benchmarks", Proceedings of 1st International Workshop on NLP and DBpedia, Vol. 1064, Sydney, Australia, October 2013.

[13] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Furstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater and Gerhard Weikum, "Robust Disambiguation of Named Entities in Text", Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, Scotland, pp. 782–792, 2011.

[14] http://publish.illinois.edu/arise-adsc/systems/