# Optimized search engine with semantic and tf-idf scores

**By: Rahul Madan**

**Abstract.** In recent years, semantic searching in tasks related to information retrieval has taken a major turn and has become a vast field of research.The social networking sites use semantic data to enhance the results. This provides us with a new perspective on how to improve the quality of information retrieval. As we are aware, many techniques of text classification are based on the TF- IDF algorithm. The World Wide Web has become colossal and its growth is also dynamic. Most of the people rely on the search engines to retrieve and share information from various resources. All the results returned by search engines are not always relevant as it is retrieved from heterogeneous data sources. Moreover a naive user finds it difficult to confirm that the retrieved results are significant to the user query.Therefore semantic web plays a major role in interpreting the relevancy of search results.

In this report, we will retrieve relevant Indian News Headlines while discussing new measures for interpreting semantic similarity between texts and queries based on the TF-IDF weighting. The algorithm enhances the already existing semantic web by using the weighted IDF feature of the TF-IDF algorithm and hence provides us with the most optimal results.

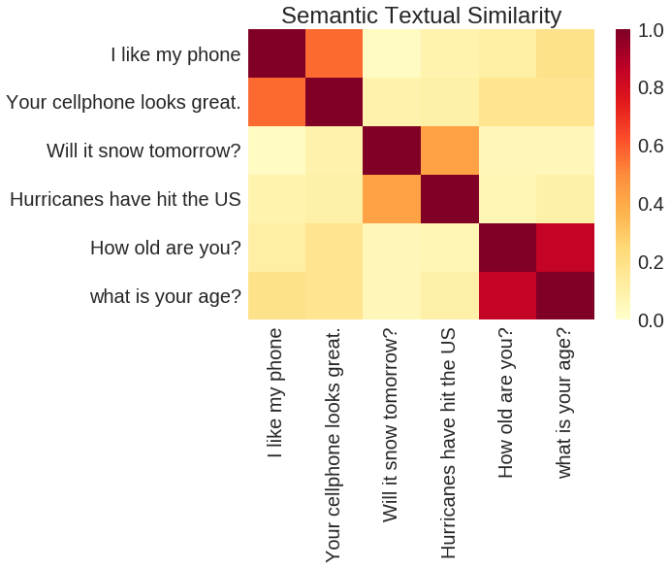
**Keywords** Semantics • Text-to-Text Semantic • Similarity • tf-idf weights • Ranking Algorithm

## 1. Introduction.

Supervised text classification in today's world has become a challenging task especially in fields of Information Retrieval, user profiles etc. Normally, the algorithms used for these text classification methods use statistical techniques like Naive Bayes Classifier, Rocchio Classifier, K nearest neighbour etc. But, these models ignore all semantics that reside within the original text. Semantics in original texts are very important as they help in classification of queries and hence improves the effectiveness. In this project, we will be involving a Semantic approach using text-to-text semantic matching. And so, we propose a new text-to-text semantic similarity measure integrated with TF-IDF weighting. Here, TF-IDF reflects how a feature is important to a document in a given corpus. Hence, our measure takes into consideration the need of a feature to each of the compared documents instead of its need to the overall corpus. This measure aggregates semantic similarities between concepts of the compared documents pair-to-pair. The automatic assignment of a batch of documents into specified classes by a learning system (classifier) that has been trained on similar data sets of test documents is known as document classification. Document indexing refers to the process of converting a document into a format that may be consumed by a categorization system. There are a variety of document indexing models, many of which rely on feature extraction, dimensionality reduction, or a combination of the two. The associated document is often represented as a feature vector encoding the presence of words, syntactic entities, or semantically connected tags, and a term weight is generated for each of these features in feature extraction. A new ranking

**Rahul Madan**
**Shiv Nadar University**
**Email: rm584@snu.edu.in**

algorithm has been developed. The algorithm utilizes the weighted IDF feature of the TF IDF algorithm. At first, search engines were lexical: the search engine looked for literal matches of the query words, without understanding of the query's meaning and only returning links that contained the exact query.




By using regular keyword search, a document either contains the given word or not, and there is no middle ground. On the other hand, "Semantic Search" can simplify query building, because it is supported by automated natural language processing programs i.e. using Latent Semantic Indexing - a concept that search engines use to discover how a keyword and content work together to mean the same thing.b LSI adds an important step to the document indexing process. LSI examines a collection of documents to see which documents contain some of those same words. LSI considers documents that have many words in common to be semantically close, and ones with less words in common to be less close.

In brief, LSI does not require an exact match to return useful results. Where a plain keyword search will fail if there is no exact match, LSI will often return relevant documents that don't contain the keyword at all.

### 1.1 Motivation behind the Topic.

Tf-idf algorithm is a bag-of-words algorithm and is generally used to find the relevant document but it doesn't actually know the meaning of the word and so we can say that "Orange" fruit and "orange" colour would be the same for this algorithm and hence we are integrating it with semantics to obtain the most optimized result.

**Pros of TF-IDF**

- You have some basic metric to extract the most descriptive terms in a document
- You can easily compute the similarity between 2 documents using it
- Easy to compute
- You have some basic metric to extract the most descriptive terms in a document
- You can easily compute the similarity between 2 documents using it

**Cons of TF-IDF**

- TF-IDF is based on the bag-of-words (BoW) model, therefore it does not capture position in text, semantics, co-occurrences in different documents, etc.
- For this reason, TF-IDF is only useful as a lexical level feature
- Cannot capture semantics (e.g. as compared to topic models, word embeddings

**Features of semantic based searching:**

- It can handle morphological variations: it should not matter whether you enter a search term in singular, plural, etc. Words like 'rain', or 'raining' or 'rained' should all lead to the same result.

- It can handle synonyms. ‘Disease’ should be understood as a synonym of ‘illness’, etc.

- It can handle generalizations: a question like ‘Which disease has the symptom of coughing?’ must retrieve all diseases where coughing is at least one of the symptoms.

- It matches concepts; it can build a relation between e.g. ‘headache’ and ‘migraine’.

- It matches knowledge. Here Hakia explains: ‘Very similar to the previous item, a semantic search engine is expected to have embedded knowledge and use it to bring relevant results (swine flu = H1N1, flu = influenza.)’

- It can handle natural language queries and questions.
- It points to the most relevant paragraphs or sections in a document: it does not stop when a document with certain keywords is retrieved; it selects the most relevant parts of it.

- The user can enter queries in a natural way; he or she does not need to use Boolean operators or quotation marks.

- It does not rely on user behaviour, statistics, linking etc. It analyses the content of documents.

### 1.2 Related Works on Semantic and tf-idf integrated searching

The authors [4] suggested an aggregation function that compares the semantic similarity of two groups of concepts by averaging the similarities of all pairs of concepts in these groups. Azuaje [5] and Wang presented a similar aggregation function that considers the maximal semantic similarity between each g1 concept and all g2 concepts, and vice versa. They suggested an algorithm where weights are assigned to consumers who are involved in text representation using a propagation technique. The authors also presented a new text-to-text similarity metric based on these weights, as well as the semantic similarity of concepts pair-to-pair, were taken into consideration. This new similarity measure is a prediction criterion that substitutes the vector space model's traditional text-to-text similarity criterion, such as Cosine. The authors claimed that using semantic similarities, they were able to cluster patents more effectively [6].

## 2 Objective.

We use semantic tag information with a web page in this project. When users submit a query, they additionally include a semantic description to help decipher the query. The user's query purpose can then be well understood by matching the semantic description between the query and the web page. In the academic realm, a greater understanding of the user's query leads to higher ranking outcomes.

### 2.1 Data Set or Corpus.

The dataset that we are using for our project is **Indian News Dataset** which is based on the continuous historical archive of notable events in the Indian Subcontinent from 2001 to 2020, recorded in real time by the journalists of India. It contains approximately 3.4 million events published by Times of India. A majority of the data is focusing on Indian local news including national, city level and entertainment.

CSV Rows: ***3,424,067***

**Link to dataset:-**
https://www.kaggle.com/therohk/india-headlines-news-dataset

**Table 1** Libraries and Packages used

| |
|---|
| Python 3.5+ |
| pip 19+ or pip3 |
| NLTK |
| TensorFlow-GPU |
| ANNOY |

**2.2 Methodology.**

1. TF-IDF model

**2.2.1 Preprocessing the Data:**

Data preprocessing is one of the most significant steps in text analytics. The purpose is to remove any unwanted words or characters which are written for human readability, but won't contribute to topic modelling in any way.

- Cleaning the data.
- Case Folding
- Word Tokenization
- Stop words removal & Lemmatization

**2.2.2 Searching Algorithms with formulae:**

- Calculating the ranking by cosine similarity ( tf-idf scoring):

  tf=> term frequency

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

  idf=> Inverse Document Frequency

$$idf(w) = log(\frac{N}{df_t})$$

- Calculating the weight of word in given document by multiplying tf-idf

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

- Finding word Semantic then finding sentence semantic and finally integrating it

- with word, sentence and paragraph semantics.

Semantic based Methodology:

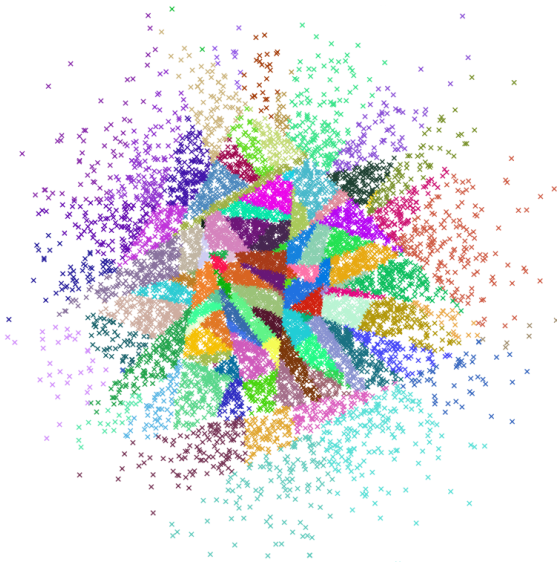
1. **Importing data** using kaggle API
2. **Selecting the desired pretrained model**- there are multiple models trained with different datasets and base algorithms. We tried 19 different models to find the most accurate one.
3. **Selecting the dimension**
   If you want to build an index using the original embedding space without random projection, set the projected_dim parameter to None. Note that this will slow down the indexing step for high-dimensional embeddings.
   Reducing the dimensionality of the embeddings with random projection means less time needed to build and query the ANN index but at the same time, lesser the accuracy.
   **There is a tradeoff between the training time and accuracy of the model.**
4. Creating test data- a total of

Modifying the sentences smartly to keep the meaning the same but changing the words origin. This happens in real life queries.

5. Converting the sentences into

### 2.3 Spotify's ANNOY

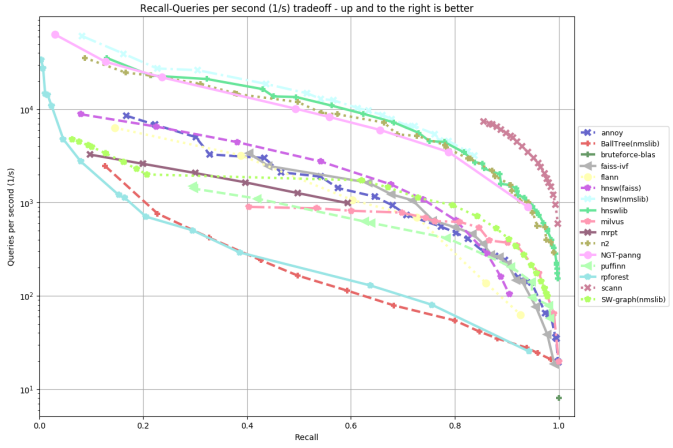
Approximate Nearest Neighbor techniques speed up the search by preprocessing the data into an efficient index and are often tackled using these phases:



There are many libraries that are used for Approximate Neighbour (ANN) searching but ANNOY is almost as fast as the fastest ANN Libraries.

Annoy (Approximate Nearest Neighbors Oh Yeah) is a C++ library with Python bindings to search for points in space that are close to a given query point. It also creates large read-only file-based data structures that are mapped into memory so that many processes may share the same data.

The main feature that ANNOY inherits is its ability to use static files as indexes i.e., you can share indexes across processes. If you want to find nearest neighbors and you have many CPUs, you only need to build the index once. It's what we use at *Spotify* to make music recommendations. Every user/item can be represented as a vector in f-dimensional space after executing matrix factorization methods. This library aids in the discovery of comparable users/items. In a high-dimensional space with several millions of tracks, memory use is a major concern.




#### 2.3.1 Pros of ANNOY

- We can tune the parameters to change the accuracy/speed tradeoff.
- It has the ability to use static files as indexes, this means you can share indexes across processes.

#### 2.3.2 Cons of ANNOY

- The exact nearest neighbor might be across the boundary to one of the neighboring cells.
- No support for GPU processing.

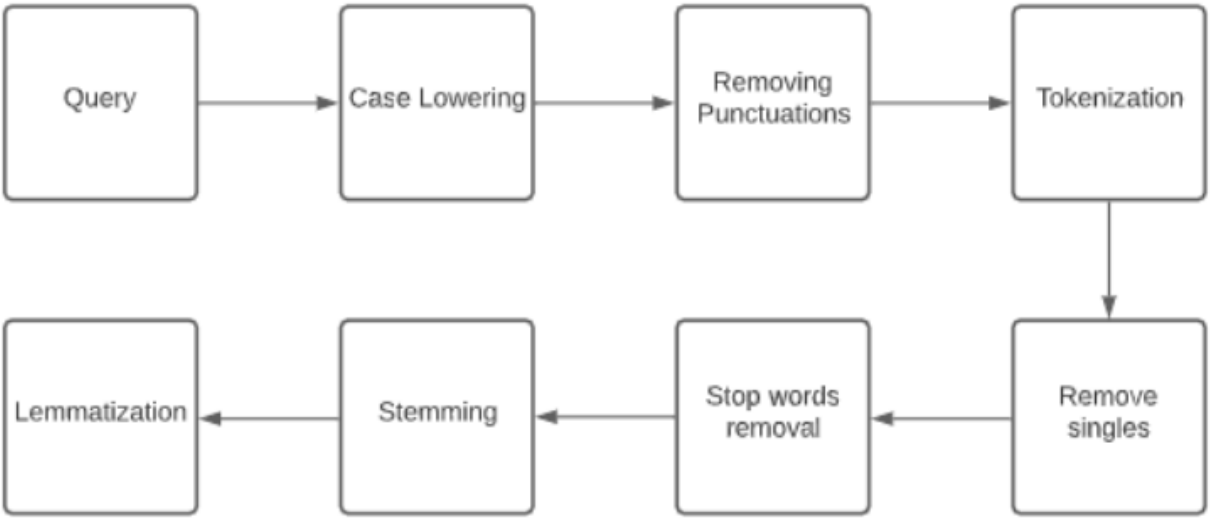



**Fig 1** Data Preprocessing Phases

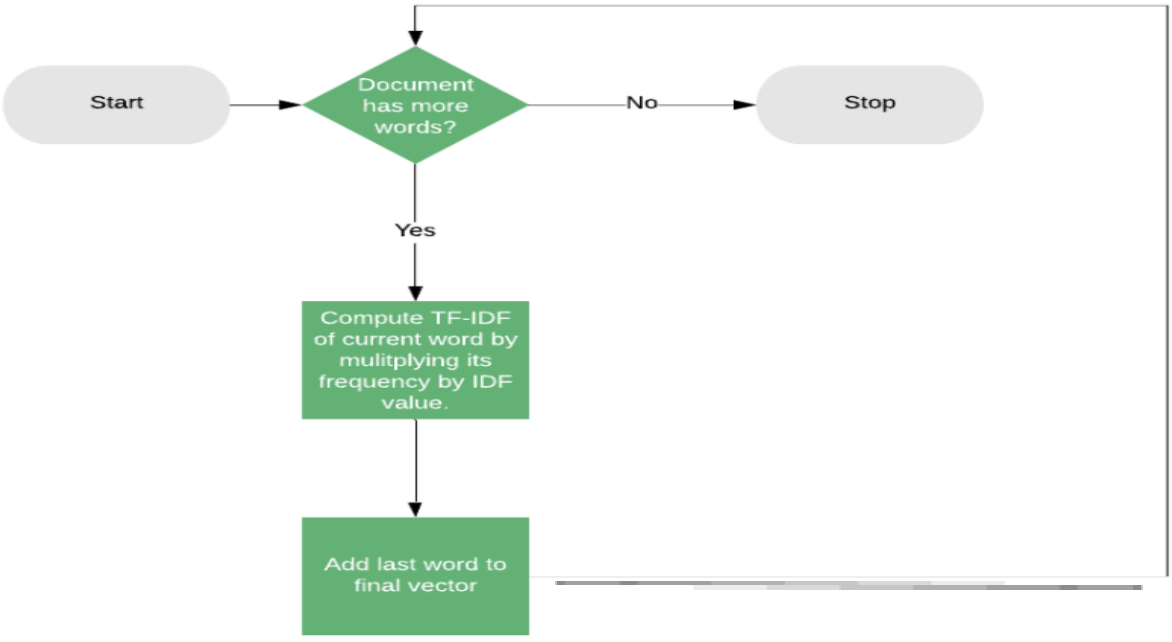



**Fig 2** TF-IDF working flowchart

## Proposed Model.

Building a search engine which would take account of semantics as well as ranking them on the basis of their term weighting.

## Experimentation and Results:

We are here trying to make use of the pros of both the searching algorithms while minimizing the cons they have by using desired features of the algorithms.

- **Limitations:**
  Training time is very large and CPU intensive

- **Observations.**

TF-IDF performs very well when the words are the same, but semantic search works well even if the synonyms of words are being used in the query.

- **Ranking factor.**

Both TF-IDF and semantic scores give a similarity score.
We tested the below mentioned 19 sentence encoder pretrained models and compared the accuracy and recall to choose the most preferable model for our use case.

Table: sorted in decreasing order of score the algorithms achieved.

| Sentence encoder Model Name | Base Model |
|---|---|
| stsb-mpnet-base-v2 | mpnet-base |
| stsb-roberta-base-v2 | roberta-base |
| stsb-distilroberta-base-v2 | distilroberta-base |
| nli-mpnet-base-v2 | mpnet-base |
| stsb-roberta-large | roberta-base |
| nli-roberta-base-v2 | roberta-base |
| stsb-roberta-base | roberta-base |
| stsb-bert-large | bert-large-uncased |
| stsb-distilbert-base | distilbert-base-uncased |
| stsb-bert-base | bert-base-uncased |
| nli-distilroberta-base-v2 | distilroberta-base |
| paraphrase-xlm-r-multilingual-v1 | XLM-R |
| paraphrase-distilroberta-base-v1 | distilroberta-base |
| nli-bert-large | bert-large-uncased |
| nli-distilbert-base | distilbert-base-uncased |
| nli-roberta-large | roberta-large |
| nli-bert-large-max-pooling | bert-large-uncased |
| nli-bert-large-cls-pooling | bert-large-uncased |
| nli-distilbert-base-max-pooling | distilbert-base-uncased |

**NumPy.**

NumPy is a Python library used for working with arrays. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with the latest CPU architectures. NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++. Works on multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
Numpy was used to handle large corpus of data without running out of memory or wasting CPU resources for processing data.

**Sklearn.**

The most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. It is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. TfidfVectorizer from the sklearn library was used to find the cosine scores of query and corpus data more efficiently after preprocessing the data.

**Apache Beam.**

Apache Beam is an open source unified programming model to define and execute data processing pipelines, including ETL, batch and stream processing. Apache Beam is an open source, unified model for defining both batch- and streaming-data parallel-processing pipelines. The Apache Beam programming model simplifies the mechanics of large-scale data processing. Using one of the Apache Beam SDKs, you build a program that defines the pipeline. Apache Beam is a unified programming model for batch and streaming data processing jobs. This made the process of finding embeddings for a large corpus faster.

**Conclusion.**

Semantic ranking looks for context and relatedness among terms, elevating matches that make more sense given the query. Language understanding finds summarizations or captions and answers within your content and includes them in the response, which can then be rendered on a search results page for a more productive search experience.

Semantic search is not beneficial in every scenario and so is TF-IDF, but certain content can benefit significantly from its capabilities. The language models in semantic search work best on searchable content that is information-rich and structured as prose. A knowledge base, online documentation, or documents that contain descriptive content see the most gains from semantic search capabilities.

We offered the new technique through our project, which is based on an existing semantic web algorithm that determines the weighted score of tags. To improve the semantic web that uses tags, we used the IDF feature of the TF-IDF algorithm. We got our dataset from kaggle.com for Indian news headlines and used the above mentioned algorithm for our experiments. We will continue to work on improving the algorithm in the future. Extracting the good features of TF-IDF and Semantic search algorithm and combining them into a single search algorithm can be used for a number of language jobs while only adding a minor layer to the main model.

## Inference and Result.

| | | msmarco-dist | paraphrase-M | all-MiniLM-L1 | paraphrase-M | multi-qa-mpn | msmarco-dist | multi-qa-mpn | multi-qa-Mini | stsb-mpnet-b | stsb-xlm-r-mult | stsb-distilbert-b |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Target Data** | **Query** | **Model 1** | **Model 2** | **Model 3** | **Model 4** | **Model 5** | **Model 6** | **Model 7** | **Model 8** | **Model 9** | **Model 10** | **Model 11** |
| Extra buses to clear tourist traffic | additional buses to lighten tourist traffic | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| IT will become compulsory in schools | information technology to become required in school | 5 | 6 | 4 | | | | | | 2 | | |
| Powerless north India gropes in the dark | weak north country gropes in less light | 8 | 5 | 8 | | | 9 | 7 | 6 | 4 | 4 | 4 |
| Cholera outbreak source traced | Cholera pandemic starting found | 5 | 7 | 2 | | | | | | 1 | | |
| Indian American boy confesses to parents murder | Indian American boy admits to mother father killing | 6 | 6 | | | 8 | | | | 3 | | |
| PNB raises rates | bank increases rates | 7 | 7 | 4 | 10 | 8 | 2 | | 2 | 1 | 2 | 4 |
| India cagey about food imports | country cagey for imported food | 7 | 4 | 6 | 5 | 1 | | | | 4 | | |
| Blood donation camp organised | Blood plasma donating camp executed | 2 | 8 | | 6 | 4 | 4 | | | 4 | | 4 |
| Treasury dept to be computerised in year | department to be moved digital in year | | 8 | 5 | 7 | | 8 | | | 2 | | 8 |
| Labour of Law | Laws of Labours | 5 | 9 | | 8 | | 8 | | | 3 | | 8 |
| Ministers under threat | Fearing ministers | 6 | 4 | 7 | 8 | | 9 | | 2 | 3 | | 9 |
| Illegal arms flood Delhi | non allowed guns and ammunation spread over in state | 3 | 3 | | 5 | | 4 | 5 | | 1 | | 4 |
| PM appeals for help | Prime minister asks assistance | 4 | 6 | 4 | 4 | 8 | 3 | 6 | 4 | 2 | | 3 |
| The Learning Curve | The Learning Curve | 4 | 7 | | 5 | 1 | 6 | 7 | 6 | 2 | | 6 |
| Hospitals are far from equipped | Hospital do not have required equipments | 3 | 8 | 3 | 5 | 9 | 7 | 8 | | 2 | | 7 |
| Christians protest over inaction | Christian revoting about the inaction of government | 1 | 10 | | | 4 | 8 | 8 | 5 | 4 | 4 | 8 |
| Delhi hosts festival of short films | State organizes special days of small movie clips | 1 | | 1 | | | 10 | 5 | | 3 | 8 | 10 |
| Helmets a must from April | Helmets made compulsory from next month | 2 | 5 | | | | | 4 | 7 | 1 | 8 | |
| The Page Three Politician | The third sheet of Politician | 3 | 3 | 6 | | 5 | 8 | 5 | | 2 | 9 | |
| Gender equality still a dream in Asian; Pacific countries | still a dream of Gender equality in Asia and Pacific | 5 | 2 | | 4 | 6 | 5 | 5 | 4 | 1 | 4 | |
| Canada to let in 235;000 immigrants in 2002 | western country to let in migrants in 2002 | 6 | | | 8 | 7 | 4 | | | 2 | 3 | |
| Mother Teresa monument near Seattle | Mother Teresa statue near a state in the united state | 7 | 1 | 6 | 8 | 8 | 5 | 5 | | 2 | 6 | 5 |
| Cellphone market abuzz with rising subscriber base | market of mobile phone abuzz with increasing users | 8 | 5 | 7 | 9 | 8 | 5 | 9 | | 1 | 7 | 6 |
| Their bamboo music is hard to beat | Bamboo music is tough to defeat in game | 8 | 6 | 8 | 4 | 5 | 6 | 3 | 5 | 5 | 8 | 7 |
| Six policemen killed in ambush | Six country protectors murdered in ambush | 5 | 3 | 1 | 3 | 4 | 3 | 3 | 6 | 6 | 10 | 8 |
| Police fail to nab gang of thieves | Police fail to catch gang of robbers | 9 | 10 | | 10 | 8 | 10 | | 9 | | | 8 |
| US asks Taliban to halt desecration of statues | United states requests Taliban group to pause desecrating monuments | 5 | 4 | 10 | 7 | 5 | 1 | | 8 | 1 | | 5 |
| Hotel bill inflated; claim Jharkhand ministers | bill inflated of tourism stay; claimed by ministers | 5 | 3 | 5 | 8 | 2 | | 8 | 8 | 2 | 5 | 4 |
| Sleazy films storm Bollywood! | unpleasant movies flooding Bollywood | 6 | 1 | 8 | 10 | 3 | 8 | 9 | 5 | 2 | 6 | 5 |
| Sessions judge to probe lock-up death | judge of sessions to investigate lock kill | | 10 | 10 | | | 9 | 9 | 10 | | 10 | |
| Microsoft launches program for developers | organizing of Microsoft program for programmers | 2 | 2 | 7 | 2 | 6 | 4 | 3 | 5 | 3 | 8 | |
| British forces to begin copter crash investigation | army of Britian to start helicopter crash probe | 1 | 3 | 4 | 3 | 1 | 5 | | 5 | 2 | 8 | |
| Bomb defused in North Block | defusing the bomb in North section | 5 | 5 | 8 | 5 | | 5 | | | 3 | 5 | |
| HBO to get into non-movie programmes | HBO starting to start non-movie shows | 6 | 6 | 8 | 6 | | 6 | | | 4 | 4 | |
| Yoga to the rescue of diabetics | diabetic patients will be rescued by yoga | 5 | 4 | 9 | 7 | | 3 | | | 5 | 5 | |
| US slowdown is real and hurting | slowdown of the united states is actual and painful | 4 | 4 | 4 | | | 2 | | | 3 | 1 | |
| GSLV launches India into elite space club | India enters elite space club because of GSLV | 5 | 5 | 1 | | | 1 | | | 2 | | |
| India must tread carefully in brave new biotech world | brave new biotech universe must be treaded by country carefully | 4 | 2 | | | | | | | 1 | | |
| **Total** | | **4.694444444** | **5.083333333** | **5.444444444** | **6.076923076** | **5.090909090** | **5.451612903** | **5.789473684** | **5.444444444** | **2.5** | **5.727272727** | **5.952380952** |

From the above chart we can infer that model 9 gives us the most accurate result of a query integrated with semantics and tf-idf weights. The average rank here of model 9 is coming out to be 2.5 whereas other models are giving way higher ranks than model 9.

## References:


1. Semantic Web Improved with IDF Feature of the TFIDF Algorithm

2. Focused Crawling Based Upon Tf-Idf Semantics and Hub Score Learning
3. Build your semantic document search engine with TF-IDF and Google-USE | by Zayed Rais | Analytics Vidhya | Medium
4. Rada, R., et al., Development and application of a metric on semantic nets. Systems, Man and Cybernetics, IEEE Transactions on, 1989.
5. Azuaje, F., H. Wang, and O. Bodenreider. Ontology-driven similarity approaches to supporting gene functional assessment. in Proceedings of the ISMB'2005 SIG meeting on Bio-ontologies. 2005.
6. Guisse, A., K. Khelif, and M. Collard. PatClust : une plateforme pour la classification sémantique des brevets. in Conférence d'Ingénierie des connaissances. 2009. Hammamet, Tunisie.
7. Comprehensive Guide To Approximate Nearest Neighbors Algorithms | by Eyal Trabelsi | Towards Data Science
8. A Data Scientist's Guide to Picking an Optimal Approximate Nearest-Neighbor Algorithm | by Braden Riggs | GSI Technology | Medium
9. Search | TensorFlow Hub (tfhub.dev)