Document Classification with Recommendation Architecture: Extensions for Feature Intensity Recognition and Column Labeling

Uditha Ratnayake[1], Tamás D. Gedeon[1], Nalin Wickramarachchi[2]

[1]School of Information Technology
Murdoch University
Murdoch WA 6150
Western Australia
{ratnayak|t.gedeon}@murdoch.edu.au

[2]Dept. of Electrical Engineering
University of Moratuwa
Moratuwa
Sri Lanka
wick@elect.mrt.ac.lk

Abstract

In this paper we present the adaptation of the Recommendation Architecture (RA) model for successfully classifying text documents. The RA is a connectionist model that attempts to simulate the human process of discovering and recognizing repeating patterns among objects. We propose to extend the RA model to effectively apply to the problem of text classification in three areas. We extend the model to use the word frequency information of the document vectors when recognizing patterns, which increase the sensitivity of the created columns (clusters). We also present a scheme to label the created columns with contributing features (words) by way of a word map. This word map represent the new patterns that the system has identified within the set of documents spanning many categories and assist a human user to assign meaning to a discovered patterns. Input to the RA is prepared in a pre-processing phase where an existing feature selection method is modified to give a priori guidance to the system about the major document categories (topics). A set of experiments is carried out with the TREC CD-5 containing news articles of the Foreign Broadcasting Information Services and LA Times. A detailed discussion of the experiment results are presented with comparison to early experiments.

Keywords feature selection, data mining, pattern discovery, pattern repetition, Recommendation Architecture, text classification

1 Introduction

Automated text classification and interpretation of the classified groups are very useful in content-based organization of large collections of text documents. Organized collections of data also facilitate data mining where it enables the user to find pieces of relevant information that that the user is not explicitly searching for as described by Kohonen et al [9].

The Recommendation Architecture theory of human cognition proposed by Coward [2,3] is a computational approach, which simulates the ability of the human brain in discovering patterns among objects. It is known that the learning in the human brain is carried out by associating new patterns with previous experiences and also that the new learning does not disrupt earlier learning. Once a pattern is learnt it leaves a large area sensitized for a pattern with some similarity to be recognized in relation to the existing ones as shown by Coward [1]. The Recommendation Architecture model is functionally separated into two subsystems called the clustering subsystem and the competitive subsystem. Here the clustering subsystem is a modular hierarchy, which functions by detection of functionally ambiguous repetition. The system gets built up to few columns depending on the input space. A high dimensional vector can be used as the input to the RA where as the output is a set of columns (clusters) representing groups of similar texts. As the system gets built up depending on the input space a large input space is compressed to a few outputs from a few columns. Columns are built when similar inputs are exposed to the system and are imprinted creating a path to the output. Once columns are built, the incoming inputs are matched with their first or sensory layer to see whether they have any similarity to the existing columns. If a totally new pattern arrives as input a new column can be created as described by Coward [3], unlike hierarchical feature maps or basic self-organizing maps having a fixed architecture which has to be defined a-priori according to Hotho et al [6]. The RA has performed well with statistically generated data as shown by Coward [4] and is being applied to real-world problems as described in Coward [5] and in Ratnayake and Gedeon [11,12].

Due to the large-scale nature and the uncertainty inherent, text classification is a challenging task. A

Proceedings of the 7th Australasian Document Computing Symposium,
Sydney, Australia, December 16, 2002.

major difficulty for text classification algorithms, especially the machine learning approaches, is the high dimensionality of the feature space. Many efforts were made to map the meaning of words to concepts and to cluster the concepts into themes by Iwayama et al [8] and Tufis et al [16]. The RA model has demonstrated that it can divide experience into ambiguous but roughly equal and largely orthogonal conditions and learn to use the indication of the presence of the conditions to determine appropriate similarity as shown by Coward [4], Coward and Gedeon [5], and Ratnayake and Gedeon [11,12]. The experience is heuristically divided up into input information conditions that repeat, and different combinations of conditions are heuristically associated with different behaviors. The RA model is able to handle high dimensional vectors as input as well as large data sets.

The inherent features of the RA model make it a natural candidate for solving a problem like classification of text documents. In this paper we describe the successful adaptation of the RA model to classify a set of newspaper articles from TREC CD-5 corpus. We extend the clustering system of the RA model to use the frequency component of the information encoded in the document vectors indicating word occurrences. We also devise a method to automatically label the created columns (clusters) based on the features learned during the training process and also depict the co-occurrence of frequent features. This information will be used to make a map display, which shows the relationships among the features that contribute most in creation and maintaining a column. In situations where a document corpus is unclassified, being able to automatically discover document classes and to be able to label them meaningfully for human identification has wide applications in data mining. Labelling of output units is also done in SOM by Rauber [14] but as there are no clear cluster boundaries it requires human inspection to mark the separated groups.

We propose to pre-process the input to optimise the performance of the system in finding associative similarities among documents. A priori guidance can be given to the system when selecting features in favour of the inputs more likely to provide useful discrimination. Unguided input space presentation results in heuristic categories according to Ratnayake and Gedeon [11], which makes it very difficult to evaluate performance with standard criteria. We also encode the information about the normalized frequency of occurrence of features to the input vectors.

This paper is organized as follows: Section 2 describes the functional overview of the Recommendation Architecture, in Section 3 we describe the experiment conducted with the extensions done to the RA, Section 4 presents the discussion of the results, Section 5 is the conclusion and the future work planned.

2 Recommendation Architecture

The two key features of the RA model are that the functionality is not defined by design and the system components exchange partially ambiguous information. The system defines its own functionality depending on the given inputs, a set of basic actions and a few internal operational measures for success and failure conditions. Ability to modify its functionality heuristically enables learning in the RA.

The modules cannot use direct consequence information such as an output from a component as an input to another component because they may need to change the inputs they receive, thus changing the output without knowing the modules that use their outputs. Therefore it is difficult to maintain an unambiguous context for the information exchanged. In the RA, the information exchanged is partially ambiguous (but not meaningless) and the functional components detect ambiguous repetitions and generate corresponding recommendations.

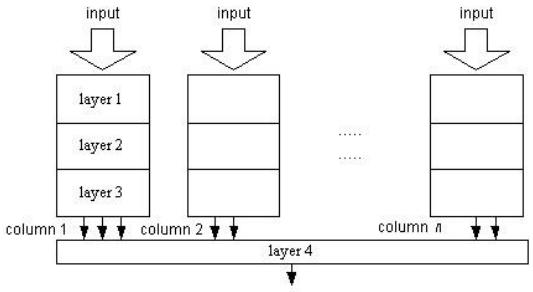

Fig.1 Overview of the 4 layers of the Recommendation Architecture

The Recommendation Architecture model by Coward [3] is a hierarchical architecture with uniform functionality at every layer. It tries to achieve an approximate equality among the functional components and attempts to minimize the information exchange between components.

The basic device that records information is a simple device. In a device, information is coded in the input connections and the threshold. Learning is carried out by gradual adjustment of thresholds and by addition of new connections. A set of devices makes one layer and a column consists of 3 layers. (Fig.1)

1. First layer (Alpha layer) selects the inputs from which information will be allowed to influence the column.
2. Second layer (Beta layer) recommends imprinting of additional repetitions in all layers.

3. Third level (Gamma layer) is the output identification layer.
4. Fourth layer is the competition or behavioural layer.

The context of information is compressed at higher layers. Each layer consists of two sets of devices called the regular section and the virgin section. Already imprinted devices reside in the regular section whereas the un-imprinted devices reside in the virgin section.

Complex repetitions are the combination of simple repeating patterns. Most simple repetitions are imprinted in devices, with devices being combined into layers, and layers into columns. Repetition of a combination occurs when a significant subset of its constituents repeat. This hierarchy of repetition represents the clustering function whereas the devices, layers and columns represent the functional components. An output from a device in the last layer indicates a programmed repetition that corresponds with an action recommendation.

The system operates in two phases. In the 'wake period' the system takes in the incoming patterns. In the 'sleep' period the system undergoes a fast re-run of the recent past experience and also synthesizes for the future. New columns are created with randomly initialised virgin devices. Inputs to the virgin devices of the first layer have a statistical bias towards the combinations that have frequently occurred when no other column has produced output. In a column that is already operating, inputs to virgin devices are randomly assigned with a statistical bias in favour of inputs that have recently caused a device to fire. The system activates at most one unused column per wake period, and only if that column has been pre-configured in a previous sleep period. Usually, an adequate number of virgin devices exist with appropriate inputs to support a path to output and if not, then more devices are configured during the next sleep period.

The number of columns created after a few wake and sleep periods does not have any relation to the number of cognitive categories of objects. Because of the use of ambiguous information, strictly separated learning and operational phases are not necessary. After a few wake-sleep periods the system continue to learn while outputs are being generated in response to early experiences. The system becomes stable as the variation in input diminishes. If a totally different set of inputs were presented a new column will be added automatically. If column outputs should be different for similar inputs then more repetition information would be taken in through additional inputs. The additional inputs will aid the system to better identify the differences in inputs.

3 Experiment

We built a reference implementation of the clustering subsystem of the RA model in C++. The model was realized as a set of multi-dimensional dynamic linked lists. As the system runs, a long series of documents are presented to the clustering system to organize its experiences into a hierarchy of repetitions. These repeating patterns, which do not necessarily correlate exactly with cognitive categories, create a few columns identifying similarities among the data. Output of a binary signal indicates the presence of the simplest complexity repetitions while a combination of binary signals indicates the presence of more complex repetitions. A few system parameters were fine tuned to get the system stable after a few hours of processing in a 1GHz desktop computer with 256 MB RAM.

The data set consists of a set of 20,000 news articles from the Foreign Broadcasting Services (FBIS) and the LA Times of the TREC CD-5 corpus [17]. Though the articles are judged for relevance to 50 topics in the TREC relevance judgements, only 10 topics have more than 100 articles each. The system requires several similar inputs to discover and imprint a pattern. Therefore we use only those 10 categories with at least minimum 100 documents in each category for our experiments. The documents were selected from the ten topic categories: 401, 412, 415, 422, 424, 425, 426, 434, 436 and 450. These topics as given in the TREC relevance judgment information document are: 401 – foreign minorities in Germany, 412 – airport security, 415 – drugs and the golden triangle, 422 – art, stolen, forged, 424 – suicides, 425 – counterfeiting money, 426 – dogs, law enforcement, 434 – economy in Estonia, 436 – railway accidents, 450 – King Hussein and peace.

3.1 Feature Selection and Preprocessing

Firstly a feature selection is done to select a set of features representative of the 10 topics. We extended the Two-Step feature selection method by Stricker et al [15] to be suitable for variable length document groups. In the Two-Step feature selection method, the words are selected if they are in the top half of each document when ordered descending by the frequency ratio. We saw that when selecting a set of words for each topic using this method a few categories were left with very few words. Mainly topic 450 was left with very few words that contribute very little to the content of the documents. Therefore we extend the Two-Step feature selection method to use a threshold based selection scheme. The words were selected if their frequency ratio was above the given threshold even if they are not in the top half of frequencies for each document.

As in the original algorithm, using those selected words, frequency for each word was calculated for

each topic but we used a simple method to eliminate the duplicates among topics instead of using the Grand-Schmidt method. The top 300 words from each frequency list were used with the duplicates across the topics deleted, and the top 125 were selected. These 125 words are of higher frequency for one topic but will be of lower frequency for other topics if they exist in other topics. Thus a feature set of 1250 words was selected using 125 words from each topic to represent the 10 topics. It is also known that a few documents in the data set were categorized as relevant to more than one topic by the TREC classification. A stop list was not used as the Two-Step algorithm automatically discards the most rare and most common words of the corpus. Stemming was not done here but will be considered for future experiments.

To represent each document an integer vector was formed by counting the frequency of occurrence of each feature (word) in the document. In the document vector, the index denotes the feature and the content indicates the frequency of occurrence of the feature. Document vectors were then normalized to base 5 (maximum number of a feature frequency set to 5) to reduce the discrepancy in the size of the documents.

Finally, the input vectors were prepared by expanding each normalized vector with its index. If the content of a particular index in the normalized vector was greater than 1, then multiple number of copies of that index was written as the input vector. These input vectors were then presented to the clustering subsystem of the RA.

The training set comprises of a set of 1500 document vectors consisting of 150 vectors from each topic. A few vectors from a few topics are duplicated once to get the minimum of 150 for that topic. The test set comprises of a new set of 1000 (500 unique vectors duplicated once to make 1000) document vectors, which has not been used for training or feature selection.

3.2 Extending the Recommendation Architecture for Feature Intensity Recognition

The basic device of the RA model is sensitive to absence or presence of a particular feature, but not to the intensity of existence of a feature. We extend the basic device to be sensitive to the intensity of the input. To support intensity as a feature attribute, the system must be able to discriminate between the function of information recognition and information recording. As noted in the earlier section, the input was preprocessed to include multiple occurrences of a feature to indicate the intensity of its occurrence. Information is recorded or imprinted by means of converting a virgin device to a regular device. The algorithm was modified in a way that, though it counts multiple occurrences of a feature for device threshold calculation, the imprinting of a feature in a device was limited to one. This allows the system to recognize the intensity of a feature as an attribute as opposed to treating the multiple occurrences as distinctly different features.

The clustering system was modified to directly accept integer vectors indicating the word occurrence frequency. The time required for processing is drastically reduced if that is done in the pre-processing stage, and only the index numbers indicating the existence and the frequency of words are given as done in the current experiment.

3.3 Extending the Recommendation Architecture for Automatic Column Labeling

As the system is presented with input experiences, the clustering subsystem organizes itself into sets of columns identifying similarities among the data. Each column will identify a particular pattern prevalent in input, which is independent of pre-existing classifications. Though a document is judged as relevant to one or more TREC topics it may have other strong characteristics in common with a sub set of documents belonging to a different topic. It is possible that the system may discover those patterns and group those documents together. By labelling the columns we can make a judgement about the actual topics the system is using to cluster the documents.

Each column is labelled by way of assigning it a word map. The map consists of single words and word pairs. A collection of single words helps to understand the grouping of the documents, and word-pairs helps to understand the context of each word. For example, if the three words, car, traffic, stolen is present in a label, knowing car-stolen pair is more frequently occurring that car-traffic suggests that the topic may be more relevant to car stealing than car traffic.

The system keeps a record of the normalized frequency of each feature that contributed to firing a device in all layers. We extend the algorithm to use this data to label each new column. The most frequently occurring 20 features in an input vector to the alpha (first) layer of a specific column are extracted as the column label. We also extended the algorithm to keep a record of the feature pairs that occur together when a device is fired. The feature and feature pair lists are then used as the map that describes each of the columns.

4 Discussion

The input vectors are presented to the system in series of runs with alternating ‘sleep’ and ‘wake’

periods. Within each 'wake' period 100 vectors were presented, representing 10 documents from each group to ensure variety of input. The vectors were interleaved to avoid consecutive inputs from the same category. The system ran for a total of 126 'wake' periods and 126 'sleep' periods with the training data set. When the data is being presented, the system starts imprinting columns for repeating input patterns. As the system gains sufficient experience (number of presentations), gamma level outputs (Level 3) can be seen from the particular columns. They represent an identified pattern in the data set.

The system created 10 stable columns. From the 10 columns, 8 produce output mainly from one topic and two columns produce output from multiple topics (Table 1).

Precision for each column is calculated as:

$$\text{Precision} = \frac{\text{Total number of documents correctly acknowledged by the column}}{\text{Total number of documents acknowledged by the column}}$$

Column No.	Major document topics discovered	Precision as a %	
		Training set	Test set
1	450	83.7	81.5
2	436	63.8	52.5
3	401	80.7	71.4
4	415	96.9	89.5
5	422	83.3	81.4
6	412	80.4	57.6
7	425	66.3	45.7
8	401,425,434	81.0	56.7
9	424	78.6	60.0
10	401,425,426	82.3	53.3
Average Precision (%)		79.7	65.0

Table 1. Precision of each column regarding to the major document category identified. (Experiment 2)

It is interesting to note that the system was able to cluster documents quite similar to the clustering done by TREC relevance judgement scores.

Table 2 shows the labels assigned by the system to each column. Label words are stemmed to remove additional forms, as they do not carry significantly important additional information. It is also interesting to note that the words selected by the system to describe a column largely resemble the TREC topic.

It can be seen that some articles have content which are common across multiple TREC topics. The system identified these documents together as a single group. For example, topics 401 (foreign minorities in Germany), 425 (counterfeiting money) and 434 (economy in Estonia) produce the majority of the output from column 8.

Column	Previously assigned TREC topic
	Automatically extracted words for columns label
1	450 - King Hussein and peace
	Jordan, Israel, Amman, Arab, Washington, Palestinian, order, military, secretary, role, administration, American
2	436 - railway accidents
	train, cars, freight, crossing, travelling, federal, transportation, hit, tons, front, stop, hospital
3	401 – foreign minorities in Germany
	German, federal, Bonn, violence, party, future, democratic, Europe, military, extremist, social, border, Klaus
4	415 – drugs and golden triangle
	Khun, Burma, sa, opium, Asia, Thailand, Bangkok, Shan, army, narcotics, order, Rangoon
5	422 - art, stolen, forged
	art, stolen, artist, gallery, painting, dealer, museum, arrested, Dutch, German, French, collection
6	412 – airport security
	airlines, federal, airport, aviation, flight, air, American, administration, Washington, bomb, screening, Scotland
7	425 – counterfeiting money
	counterfeit, order, social, party, crime, arrested, legal, Russia, economy, printing, notes, German
8	401 – foreign minorities in Germany, 425 – counterfeiting money, 434 - economy in Estonia
	goods, future, efforts, social, federal, Estonian, cooperation, Germany, products, order, party, financial, equipment, industrial, population
9	424 – suicides
	judge, kevorkian, prosecutor, michigan, ruled, jack, jail, deputies, motel, doctor, arrested, medical
10	401 – foreign minorities in Germany, 425 – counterfeiting money, 426 – dogs, law enforcement
	car, officers, German, arrested, federal, dog, search, military, American , incident, party, stolen, suspects, crime, robbery

Table 2. TREC assigned topic/topics for the major group discovered by each column and labels assigned to the columns by the system.

The words describing the column 8 mainly give the idea of social and economic situations whereas column 10 producing output from 401 (foreign minorities in Germany), 425 (counterfeiting money) and 426 (dogs, law enforcement) gives the idea of crime, robbery and arrests. Though both topics 401 and 425 get combined with another topic, in both cases the words describing the columns clearly show the difference between them.

We calculate the frequency that two words come together as input to a device when it fires. Table 4 shows a part of the word list for Column 1 which corresponds to TREC topic 401 (King Hussein and peace). For all the clusters a similar list is produced which depicts the context for a feature in Table 3. For example, it can be seen that the word 'role' occurs in the context of a country/region as oppose to military, administration or secretary which are the other frequent words in the label map.

Word 1	Word 2	Frequency
role	jordan	125
role	israel	110
role	amman	109
role	region	104
role	washington	103
role	future	90
role	efforts	88
role	israeli	84
role	arab	81
role	negotiations	75
role	palestinian	62
palestinian	jordan	156
palestinian	israel	131
palestinian	arab	130
palestinian	amman	127
palestinian	washington	110
palestinian	israeli	102
palestinian	negotiations	78
palestinian	region	71
palestinian	efforts	65
palestinian	future	65
…	…	…

Table 4: Frequently occurring feature pairs (a section of column 1)

In the following section we present a comparison of the results achieved prior to extending the system to feature intensity recognition.

Table 3 presents the results of an earlier experiment (Experiment 1) done by Ratnayake and Gedeon [12] with the same data set for training and testing and with the same feature set. Here the system was working with binary document vectors that indicated only the presence or absence of a feature. The current experiment (Experiment 2) produced 10 columns with 8 columns uniquely identifying one TREC topic whereas the earlier one produced only 8 columns with 6 uniquely identify the TREC topics.

Column No.	Major document topics discovered	Precision as a %	
		Training set	Test set
1	412	67.0	36.4
2	401	71.1	62.8
3	415	84.7	76.5
4	424,425,426	84.4	75.0
5	422	75.0	50.0
6	450	75.8	68.1
7	436	72.6	54.5
8	424,425	75.6	61.8
Average Precision (%)		75.77	60.50

Table 3. Precision of each column regarding to the major document category identified. (Experiment 1)

The feature intensity recognition extension also resulted in improvement in average systems Precision from 60.5% (Experiment 1) to 65.0% (Experiment 2) for the test data set. It is also interesting to note the improvement in worst-case Precision in Experiment 1 topic 412 from 36.4% to 57.6% in Experiment 2.

5 Conclusion and Future Work

Text classification is a process of uncovering the associative similarities between various documents. The inherent features of the RA model in pattern synthesis and recognition makes it suitable for solving this kind of real-world problem. Instead of mapping words to concepts before presenting to the system, here the system discovers its own higher-level similarities.

In this paper we present a method for modelling the input space of RA to apply it for a real word text classification problem. We also present enhancements to the RA model for feature intensity recognition and an approach to automatically label the document groups for successfully applying it to pattern discovery in documents.

Finding the descriptive words for the columns and their relationship to one another is very informative in interpreting the results. Especially where more than one predefined category produce output from the same column, the characteristics of the column become self-explanatory. In future, a graphical notation will be developed to depict the relationship among the most relevant and determining features in the input in forming a group of their own.

References

[1] L.A. Coward, *Pattern Thinking*, Praeger, New York, 1990.

[2] L.A. Coward, The Pattern Extraction Hierarchy Architecture: A Connectionist Alternative to the von Neumann Architecture, *Mira, J., Morenzo-Diaz, R., and Cabestany, J., (eds.) Biological and Artificial Computation: from Neuroscience to Technology*, Springer, Berlin, pp. 634-43, 1997.

[3] L.A Coward, A Functional Architecture Approach to Neural Systems, *International Journal of Systems Research and Information Systems*, pp. 69-120, 2000.

[4] L.A. Coward, The Recommendation Architecture: Lessons from Large-Scale Electronic Systems Applied to Cognition, *Journal of Cognitive Systems Research* Vol.2, No. 2, pp. 115-156, 2001.

[5] L.A Coward, T.D. Gedeon, W.D. Kenworthy, Application of the Recommendation Architecture to Telecommunications Network Management, *International Journal of Neural Systems*, Vol. 11, No. 4, pp. 323-327, 2001.

[6] M. Dittenbach, D. Merkl and A. Rauber, "The Growing Hierarchical Self-Organizing Map", *Int'l Joint Conference on Neural Networks (IJCNN'2000, Como, Italy*, pp. 24-27, 2000.

[7] A. Hotho, S. Staab, and A Manche, "Ontology-based Text Clustering", *IJCAI-01 Workshop on Text Learning: Beyond Supervision*, Seattle, Washington, 2001.

[8] M. Iwayama, T. Tokunaga, "Cluster-based text categorization: a comparison of category search strategies", *Proceedings of the 18th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, Seattle, Washington, US, pp. 273 - 280, 1995.

[9] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, V. Paatero, A. Saarela, Self Organization of a Massive Document Collection, *IEEE Transactions on Neural Networks, Special Issue on Data Mining and Knowledge Discovery*, Vol 11, No 3, pp. 574-585, 2000.

[10] D. Merkl and A. Rauber, "Document Classification with Unsupervised Neural Networks", *Soft Computing in Information Retrieval: Techniques and Applications, F. Crestani and G. Pasi (Eds.*), Physica Verlag, Heidelberg, Germany, pp. 102-121, 2000.

[11] U. Ratnayake, T.D. Gedeon, Application of the Recommendation Architecture Model for Document Classification, *Proceedings of the 2nd WSEAS Intl. Conference on Scientific Computation and Soft Computing*, Crete, 2002.

[12] U. Ratnayake, T.D. Gedeon, Extending The Recommendation Architecture Model For Effective Text Classification, *The Sixth Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, Canberra, Australia, 2002 (in-press)

[13] A. Rauber, E. Schweighofer, D. Merkl, "Text Classification and Labeling of Document Clusters with Self-Organizing Maps", *journal of the Austrian Society for Artificial Intelligence (ÖGAI)*, vol. 13:3, pp. 17-23, 2000.

[14] A. Rauber, LabelSOM:On the Labeling of Self-Organizing Maps, *Int'l Joint Conference on Neural Networks (IJCNN'99). Proceedings*, Vol. 5, pp. 3527-32, 1999.

[15] M. Stricker, F. Vichot, G. Dreyfus, F. Wolinski, Two-Step Feature Selection and Neural Classification for the TREC-8 Routing, *Proceedings of the 8th Text Retrieval Conference (TREC 8)*, 1999.

[16] D. Tufis, C. Popescu, and R. Rosu, Automatic Classification of Documents by Random Sampling, *Publish House Proceedings of the Romanian Academy Series A*, Vol 1, No. 2, pp. 117-127, 2000.

[17] Text REtrieval Conference (TREC), Data collection, http://trec.nist.gov/data/docs_eng.html