

Internet Search Assistant based on the Random Neural Network

Thesis submitted for the Degree of
Doctor of Philosophy of the University of London
and
the Diploma of Imperial College

June 2018
Supervisor: Professor Erol Gelenbe

Guillermo Serrano Bermejo (Will Serrano)
g.serrano11@imperial.ac.uk
Intelligent Systems and Networks Group
Electrical and Electronic Engineering Department
Imperial College London

This work is on my own and else is appropriately referenced.

'The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any use reuse or redistribution, researches must make clear to others the licence terms of this work'

I would like to thank and express my admiration and gratitude to Professor Erol Gelenbe; his personal and academic support during my challenging research studies has made this PhD and Thesis real.

I feel privileged to have been to Professor Erol Gelenbe' PhD student; sharing some of his ideas has been very rewarding. Professor Erol Gelenbe has made an exceptional academic and industrial contribution to the Artificial Intelligence and Machine Learning research field.

In addition; I would like to express my gratitude to my viva examiners, search validators and Imperial College London.

This Thesis and research work is dedicated to its readers.

Blank Page

Table of Contents

Abstract.....	12
1 Introduction	13
1.1 Research Proposal	14
1.2 Related Work	15
1.3 Summary of Contributions.....	21
1.4 Summary of Publications	23
2 Web Search	24
2.1 Internet Assistants	24
2.2 Web Search Engines	26
2.3 Metasearch Engines.....	27
2.4 Web result clustering	30
2.5 Travel Services	33
2.6 Citation Analysis	36
3 Ranking	38
3.1 Ranking Algorithm.....	38
3.2 Relevance Metrics	43
3.3 Learning to Rank.....	48
4 Recommender Systems	50
4.1 Recommender System Types.....	50
4.2 Recommender System Relevance Metrics	51
4.3 Recommender System Model	53
5 The Random Neural Network.....	54
5.1 Neural Networks	54
5.2 Deep Learning	55
5.3 G-Networks	56
5.4 The Random Neural Network	58
5.5 The Deep Learning Cluster Random Neural Network	67
5.6 Random Neural Network Extensions	77
5.7 Random Neural Network Applications.....	79
6 Internet Search Assistant Model	87

6.1	Intelligent Search Assistant Model	87
6.2	Result Cost Function	88
6.3	User iteration.....	91
6.4	Dimension Learning	94
6.5	Gradient Descent Learning	96
6.6	Reinforcement Learning	97
7	Unsupervised Evaluation	99
7.1	Implementation	99
7.2	Spearman's Rank Correlation Coefficient	101
7.3	Google Search	101
7.4	Web Search Evaluation	104
7.5	Metasearch Evaluation	107
8	User Evaluation – First Iteration	111
8.1	Implementation	111
8.2	Quality Metric	113
8.1	Google Search – Result Cost Function	113
8.2	Web Search – Result Cost Function.....	115
8.3	Google Search - Fixed Query – Relevant Centre Point	116
8.4	Google Search – Open Query - Relevant Centre Point	117
9	User Evaluation – Learning algorithms.....	119
9.1	Quality Metric	119
9.2	Web Search Evaluation	120
9.3	Academic Database Evaluation.....	128
9.4	Recommender System Evaluation.....	141
10	User Evaluation – Deep Learning.....	158
10.1	Implementation	159
10.2	Evaluation	159
10.3	Experimental Results	161
11	Conclusions	168
12	References	171
	Appendix	190
	A ISA Screen shots	190

B Unsupervised Evaluation.....	191
C User Evaluation – First Iteration	192
Google Search – Result Cost Function	192
Web Search – Result Cost Function	194
D User Evaluation – Learning Algorithms	202
E User Evaluation – Deep Learning	239

List of Figures

Figure 1: Web search engine architecture	26
Figure 2: Metasearch services model	28
Figure 3: Metasearch engine architecture	29
Figure 4: Web Cluster Engine Architecture	31
Figure 5: Traditional travel services model.....	34
Figure 6: Online travel services model.....	35
Figure 7: Learning to Rank Model	49
Figure 8: Recommender system architecture	50
Figure 9: Common types of Recommender systems	51
Figure 10: Recommender system model.....	53
Figure 11: Artificial Neural Network – Recurrent and feed forward models	54
Figure 12: Artificial Neural Network – Deep Learning model	56
Figure 13: Random Neural Network: Principles.....	59
Figure 14: Random Neural Network: Model.....	60
Figure 15: Random Neural Network: Theorem.....	61
Figure 16: Random Neural Network: Gradient Descent learning.....	63
Figure 17: Random Neural Network: Gradient Descent iteration	64
Figure 18: Random Neural Network: Reinforcement Learning.....	66
Figure 19: Random Neural Network: Reinforcement iteration	67
Figure 20: Cluster Random Neural Network: Principles	69
Figure 21: Cluster Random Neural Network: Model	70
Figure 22: Single Cluster Random Neural Network: Theorem	70
Figure 23: Multiple Cluster Random Neural Network: Theorem	72
Figure 24: Cluster Random Neural Network: Gradient Descent learning	75
Figure 25: Cluster Random Neural Network: Gradient Descent iteration	76
Figure 26: The Random Neural Network with a Management Cluster.....	77
Figure 27: Internet Search Assistant Model.....	88
Figure 28: Intelligent Search Assistant User Iteration.....	93
Figure 29: Intelligent Search Assistant Dimension Learning	96
Figure 30: Intelligent Search Assistant Model – Gradient Descent Learning	97
Figure 31: Intelligent Search Assistant Model – Reinforcement Learning.....	98
Figure 32: ISA Client side implementation	100
Figure 33: ISA Client interface	100
Figure 34: Google Search evaluation.....	104
Figure 35: Web search evaluation – Average Values	106

Figure 36: Metasearch Evaluation – Average Values.....	110
Figure 37: ISA Server implementation	112
Figure 38: ISA Server interface	112
Figure 39: Google Search – Result Cost Function.....	114
Figure 40: Web Search – Result Cost Function	115
Figure 41: Google Search - Fixed Query – Relevant Center Point	116
Figure 42: Google Search – Open Query - Relevant Center Point	118
Figure 43: ISA Web Search interface.....	121
Figure 44: Web Search evaluation – Gradient Descent - Average Values.....	122
Figure 45: Web Search evaluation – Gradient Descent - Improvement	122
Figure 46: Web Search evaluation – Reinforcement Learning - Average Values	124
Figure 47: Web Search evaluation – Reinforcement Learning – Improvement.....	124
Figure 48: Web Search evaluation – Evaluation between learnings	125
Figure 49: Relevance Metric Evaluation – Gradient Descent	127
Figure 50: Relevance Metric Evaluation – Reinforcement Learning	127
Figure 51: ISA Academic Database interface.....	128
Figure 52: Database evaluation – Gradient Descent - Average Values	130
Figure 53: Database evaluation – Gradient Descent – Improvement	130
Figure 54: Database evaluation – Reinforcement Learning - Average Values	132
Figure 55: Database evaluation – Reinforcement Learning - Improvement.....	132
Figure 56: Database evaluation – Evaluation between learnings	133
Figure 57: Database Evaluation - Quality order by ISA.....	134
Figure 58: Database Evaluation - Improvement order by ISA.....	135
Figure 59: Database Evaluation - Quality order by Academic Database.....	136
Figure 60: Database Evaluation - Improvement order by Academic Database	136
Figure 61: Database Evaluation - Quality ISA and Online Academic Database	138
Figure 62: Database Evaluation - Improvement ISA and Academic Database	138
Figure 63: Relevance Metric Evaluation – Gradient Descent	140
Figure 64: Relevance Metric Evaluation –Reinforcement Learning	140
Figure 65: ISA Recommender - Film interface	142
Figure 66: Recommender Evaluation – Film Database	144
Figure 67: Recommender Evaluation - Improvement – Film Database	145
Figure 68: ISA Recommender – Trip Advisor interface.....	147
Figure 69: Recommender Evaluation – Trip Advisor Car Database	149
Figure 70: Recommender Evaluation - Improvement – Trip Advisor Car Database	149
Figure 71: Recommender Evaluation – Trip Advisor Hotel Database	152
Figure 72: Recommender Evaluation -Improvement – Trip Advisor Hotel Database.....	152

Figure 73: ISA Recommender – Amazon interface	154
Figure 74: Recommender Evaluation – Amazon Database.....	156
Figure 75: Recommender Evaluation - Improvement – Amazon Database.....	156
Figure 76: ISA Deep Learning Clusters model	158
Figure 77: ISA Deep Learning cluster interface.....	159
Figure 78: Deep Learning Cluster Evaluation.....	160
Figure 79: Management Cluster Evaluation	161
Figure 80: Deep Learning Cluster Evaluation – Average Results.....	163
Figure 81: Best Performing Cluster Evaluation – Average Results	165
Figure 82: Management Cluster Evaluation – Average Results.....	167
Figure 83: ISA Interface.....	190
Figure 84: ISA Result presentation	190
Figure 85: Web Search Evaluation	191
Figure 86: Meta Search Evaluation	191
Figure 87: Google Search Result Cost Function.....	193
Figure 88: Google Search Result Cost Function.....	194
Figure 89: Web Search Result Cost Function	196
Figure 90: Web Search Result Cost Function	199
Figure 91: Google Search Relevant Centre Point.....	200
Figure 92: Google Search Relevant Centre Point.....	201
Figure 93: Web Search Learning evaluation – Gradient Descent - Average.....	216
Figure 94: Web Search Learning evaluation – Reinforcement Learning - Average.....	217
Figure 95: Web Search Learning evaluation – Evaluation between learnings	217
Figure 96: Database Learning evaluation – Gradient Descent - Average	237
Figure 97: Database Learning evaluation – Reinforcement Learning - Average	237
Figure 98: Database Learning evaluation – Evaluation between learnings	238

List of Tables

Table 1: Google search evaluation – Master Result List.....	102
Table 2: Google search evaluation – Google first 10 results	103
Table 3: Google search evaluation – ISA first 10 results	103
Table 4: Web search evaluation.....	105
Table 5: Metasearch evaluation	108
Table 6: Google Search – Result Cost Function.....	114
Table 7: Web Search – Result Cost Function	115
Table 8: Google Search - Fixed Query – Relevant Center Point	116
Table 9: Google Search – Open Query - Relevant Centre Point	117
Table 10: Web Search evaluation – Gradient Descent - Average Values.....	121
Table 11: Web Search evaluation – Reinforcement Learning - Average Values.....	123
Table 12: Relevance Metric evaluation – Gradient Descent - Average Values.....	126
Table 13: Relevance Metric evaluation – Reinforcement Learning - Average Values.....	126
Table 14: Database evaluation – Gradient Descent - Average Values	129
Table 15: Database evaluation – Reinforcement Learning - Average Values	131
Table 16: Database Evaluation - Quality order by ISA	134
Table 17: Database Evaluation - Quality order by Academic Database.....	135
Table 18: Database Evaluation - ISA and Online Academic Database	137
Table 19: Relevance Metric evaluation – Gradient Descent Average	139
Table 20: Relevance Metric evaluation –Reinforcement Learning Average	139
Table 21: Recommender Evaluation – Film – Gradient Descent	143
Table 22: Recommender Evaluation – Film – Reinforcement Learning	143
Table 23: Recommender Evaluation – Trip Advisor Car – Gradient Descent.....	147
Table 24: Recommender Evaluation – Trip Advisor Car – Reinforcement Learning.....	148
Table 25: Recommender Evaluation – Trip Advisor Hotel – Gradient Descent	150
Table 26: Recommender Evaluation – Trip Advisor Hotel– Reinforcement Learning	151
Table 27: Recommender Evaluation – Amazon – Gradient Descent	154
Table 28: Recommender Evaluation – Amazon – Reinforcement Learning	155
Table 29: Deep Learning Cluster Evaluation – Average Results.....	162
Table 30: Best Performing Cluster Evaluation – Average Results.....	164
Table 31: Management Cluster Evaluation – Average Results	166

Abstract

Web users can not be guaranteed that the results provided by Web search engines or recommender systems are either exhaustive or relevant to their search needs. Businesses have the commercial interest to rank higher on results or recommendations to attract more customers while Web search engines and recommender systems make their profit based on their advertisements. This research analyses the result rank relevance provided by the different Web search engines, metasearch engines, academic databases and recommender systems.

We propose an Intelligent Search Assistant (ISA) that addresses these issues from the perspective of end-users acting as an interface between users and the different search engines; it emulates a Web Search Recommender System for general topic queries where the user explores the results provided. Our ISA sends the original query, retrieves the provided options from the Web and reorders the results.

The proposed mathematical model of our ISA divides a user query into a multidimensional term vector. Our ISA is based on the Random Neural Network with Deep Learning Clusters. The potential value of each neuron or cluster is calculated by applying our innovative cost function to each snippet and weighting its dimension terms with different relevance parameters.

Our ISA adapts to the perceived user interest learning user relevance on an iterative process where the user evaluates directly the listed results. Gradient Descent and Reinforcement Learning are used independently to update the Random Neural Network weights and we evaluate their performance based on the learning speed and result relevance.

Finally, we present a new relevance metric which combines relevance and rank. We use this metric to validate and assess the learning performance of our proposed algorithm against other search engines. In some situations, our ISA and its iterative learning outperforms other search engines and recommender systems.

1 Introduction

The need to search for specific information in the ever expanding Internet has led the development of Web search engines and Recommender systems [27]. Whereas their benefit is the provision of a direct connection between users and the information or products sought, any search outcome will be influenced by a commercial interest as well as by the users' own ambiguity in formulating their requests or queries. Sponsored search enables the economic revenue that is needed by Web search engines [15]; it is also vital for the survival of numerous Web businesses and the main source of income for free to use online services. Multiple payment options adapt to different advertiser targets while allowing a balanced risk share among the advertiser and the Web search engine for which pay-per-click method is the widest used model. Commercial applications of Recommender Systems range from product suggestion, sponsored search and targeted advertising.

The Internet has fundamentally changed the travel industry; it has enabled real time information and the direct purchase of services and products; Web users can buy directly flight tickets, hotels rooms and holiday packages. Travel industry supply charges have been decreased or eliminated because the Internet has provided a shorter value chain [43]; however services or products not displayed within higher order of Web Search Engines or Recommender systems lose tentative customers. A parallel situation is also found in academic and publications search where the Internet has permitted the open publication and accessibility of academic research; Open Access publication has generated economic, society and academic impact through their two Access levels: Green with copyright limitations and Gold with minor article processing charges with a business model based on a subscription fee. Authors are able to avoid the conventional method of the journal human evaluation [50] and upload their work on to their personal Websites. With the intention to expand the research contribution to a wider number of readers and be more cited [52], authors have the personal interest to show publications at high rank orders in academic search.

Ranking algorithms are critical in the presented examples as they decide on the result relevance and order therefore making data as transparent or no transparent to e-commerce customers and general Web users. Considering the Web search commercial model, businesses or authors are interested in distorting ranking algorithms by falsely enhancing the appearance of their publications or items whereas Web search engines or

Recommender systems are biased to pretend relevance on the rank they order results from explicit businesses or Web sites in exchange for a commission or payment. The main consequence for a Web user is that relevant products or results may be “hidden” or displayed at the very low order of the search list and unrelated products or results on higher order.

Artificial neural networks are models based on the brain within the central nervous system; they are usually presented as artificial nodes or "neurons" in different layers connected together via synapses. The learning properties of artificial neural networks have been applied to resolve extensive and diverse tasks that would have been difficult to solve by ordinary rules based programming; these applications include optimization [145,146] and image and video recognition [149,154]. Neural Networks have also been applied to Web Search and result rank and relevance [206,207].

1.1 Research Proposal

In order to address these challenges from an end-user perspective, this research proposes a neuro-computing approach to the design of an Intelligent Search Assistant (ISA) based on the Random Neural Network [135]. ISA acts as a Web Search Recommender System for general topic queries where the user explores the results provided rather than providing the better one off search results; ISA is an interface between an individual user's query and the different Web search engines or recommender systems that uses a learning approach to iteratively determine the best set of results that best match the learned characteristics of the end user's queries. ISA benefits the users by showing relevant set results on first positions rather than a best result surrounded of many other irrelevant results.

This research presents ISA which acquires a query from the user, retrieves results or snippets from one or more search engines or recommender systems and reorders them according to an innovative cost function that measures snippets' relevance. ISA represents queries and search outcomes as vectors with logical or numerical entries, where the different terms are weighted with relevance parameters. A Random Neural Network (RNN) with Deep Learning Clusters is designed, with one neuron or cluster of neurons for each dimension of the data acquired from the Web as a result of a query, and it is used to determine the relevance (from the end-user's perspective) of the acquired result. User relevance is learned iteratively from user feedback, using independently either Gradient Descent to reorder the results to the minimum distance to

the user Relevant Center Point or Reinforcement Learning that rewards the relevant dimensions and "punish" the less relevant ones.

Learning in our proposed method is not global user-specific learning, where user iterations are specific to the current search session. This thesis considers the term iteration as user search iterations rather than the algorithm iterations of the machine learning methods. This iterative approach is useful in search situations where the user wants to explore a broad topic; such as conducting literature reviews on Google scholar or searching for deals when planning travel, but less helpful if looking for the answer to a simple question.

The proposed algorithm has been validated with real searches instead of computer simulation or Web Search algorithm datasets. The results are shown as "proof of concept" rather than statistically significant with small confidence intervals; however, statistical significance to demonstrate estimation and hypothesis of the research evaluation is also reported.

1.2 Related Work

Artificial neural networks are representations of the brain, the principal component of the central nervous system. They are usually presented as artificial nodes or "neurons" in different layers connected together via synapses to create a structure that emulates a biological neural network. The synapses have values called weights which are updated during the learning algorithm calculations.

1.2.1 Neural Networks in Web Search

The capability of a neural network to learn recursively from several input figures to obtain the preferred output values has also been applied in the World Wide Web as a user interest adjustment method to provide relevant answers. Neural networks have modelled Web graphs to compute page ranking; a Graph Neural Network presented by Franco Scarselli et al [94] consists on nodes with labels to include features or properties of the Web page and edges to represent their relationships; a vector called state is associated to each node, it is modelled using a feed forward neural network that represents the reliance of a node with its neighbourhood. Another graph model developed by Michael Chau et al [95] assigns every node in the neural network to a Web page where the synapses that connect neurons denote the links that connect Web pages; the Web page material rank approximation uses the Web page heading, text material and

hyperlinks; the link score approximation applies the similarity between the quantity of phrases in the anchor text that are relevant pages. Both Web page material rank approximation and Web link rank approximation are included within the connection weights between the neurons.

A unsupervised neural network is presented in a Web search engine by Sergio Bermejo et al [96] where the k-means method is applied to cluster n Web results retrieved from one or more Web search engines into k groups where k is automatically estimated; once the results have been retrieved and feature vectors are extracted, the k means grouping algorithm calculates the clusters by training the unsupervised neural network. In addition, a neural network method proposed by Bo Shu et al [97] classifies the relevance and reorders the Web search results provided by a metasearch engine; the neural network is a three layer feed forward model where the input vector represents a keyword table created by extracting title and snippets words from all the results retrieved and the output layer consists of a unique node with a value of 1 if the Web page is relevant and 0 if irrelevant.

A neural network that ranks pages using the HTML properties of the Web documents was introduced by Justin Boyan et al [98] where words in the title have a stronger weight than in the body; it propagates the reward back through the hypertext graph reducing it at each step. A back propagation neural network defined by Shuo Wang et al [99] is applied to Web search engine optimization and personalization where its input nodes are assigned to an explicit measured Web user profile and a single output node represents the likelihood the Web user may regard Web page as relevant.

An agent learning method is applied to Web information retrieval by Yong S. Choi et al [100] where every agent uses different Web search engines and learns their suitability based on user's relevance response; a back propagation neural network is applied where the input and the output neurons are configured to characterize any training term vector set and relevance feedback for a given query.

1.2.2 Neural Networks in Learning to Rank Algorithms

Although there are numerous learning to rank methods we only analyse the ones based on Neural Networks. Learning to rank algorithms are categorized within three different methods according to their input and output representation:

The Pointwise method assigns every query couple of Web pages within a training set to a quantitative or ordinal value. It assumes an individual document as its only learning input. Pointwise is represented as a regression model where provided a unique query document couple, it predicts its rating.

The Pairwise method evaluates only the relative order between a pair of Web pages; it collects documents pairs from the training data on which a label that represents the respective order of the two documents is assigned to each document pair. Pairwise is approximated by a classification problem. Algorithms take document pairs as instances against a query where the optimization target is the identification of the best document pair preferences. RankNet was proposed by Christopher J. C. Burges et al [101], it is a pairwise model based on a neural network structure and Gradient Descent as a method to optimize the probabilistic ranking cost function; a set of sample pairs together with target likelihoods of one result is to be ranked higher than the other is given to the learning algorithm. RankNet is used by Matthew Richardson et al [102] to combine different static Web page attributes such as Web page content, domain or outlinks, anchor text or inlinks and popularity; it outperforms PageRank by selecting attributes that are detached from the link fabric of the Web where accuracy can be increased by using the regularity Web pages are visited. RankNet adjusts attribute weights to best meet pairwise user choices as presented by Eugene Agichtein et al [103] where the implicit feedback such as clickthrough and other user interactions is treated as vector of features which is later integrated directly into the ranking algorithm. SortNet was defined by Leonardo Rigutini et al [104]; it is a pairwise learning method with its associated priority function provided by a multi-layered neural network with a feed forward configuration; SortNet is trained in the learning phase with a dataset formed of pairs of documents where the associated score of the preference function is provided, SortNet is based on minimization of square error function between the network outputs and preferred targets on every unique couple of documents.

The Listwise method takes ranked Web pages lists as instances to train ranking models by minimizing a cost function defined on a predicted list and a ground truth list; the objective of learning is to provide the best ranked list. Listwise learns directly document lists by treating ranked lists as learning instances instead of reducing ranking to regression or classification. ListNet was proposed by Zhe Cao et al [105]; it is a Listwise cost function that represents the dissimilarity between the rating list output generated by a ranking model and the rating list given as master reference; ListNet maps the relevance tag of a query Web page set to a factual value with the aim to define the

distribution based on the master reference. ListNet is built on a neural network with an associated Gradient Descent learning algorithm with a probability model that calculates the cost function of the Listwise approach; it transforms the ratings of the allocated documents into probability distributions by applying a rating function and implicit or explicit human assessments of the documents.

1.2.3 Neural Networks in Recommender Systems

Neural Networks have been also applied in Recommender Systems as a method to predict user ratings to different items or to cluster users or items into different categories.

A recommender system based on a collaborative filtering application using the k-separability method was proposed by Smita Krishna Patil et al [106]; it is built for every user on various stages: a collection of users is clustered into diverse categories based on their likeness applying Adaptive Resonance Theory, then the Singular Value Decomposition matrix is calculated using the k separability method based on a neural network with a feed forward configuration where the n input layer corresponds to the user ratings' matrix and the single m output the user model with $k = 2m+1$. An ART model is used by Cheng-Chih Chang et al [107] to cluster users into diverse categories where a vector that represents the user's attributes corresponds to the input neurons is and the applicable category to the output ones.

A Recommender application that joins Self Organizing Map with collaborative sorting was presented by Meehee Lee et al [108]; it applies the division of customers by demographic features in which customers that correspond to every division are grouped following to their item selection; the Collaborative filtering method is used on the group assigned to the user to recommend items. The SOM learns the item selection in every division where the input is the customer division and the output is the cluster type. A SOM that calculates the ratings between users was defined by M. K. Kavitha Devi et al [109] to complete a sparse scoring matrix by forecasting the rates of the unrated items where the SOM is used to identify the rating cluster.

There are different frameworks that combine collaborative sorting with neural networks. Implicit patterns among user profiles and relevant items are identified by a neural network as presented by Charalampos Vassiliou [110]; those patterns are used to improve collaborative filtering to personalize suggestions; the neural network algorithm

is a multiplayer feed forward model and it is trained on each user ratings vector. The neural network output corresponds to a pseudo user ratings vector that fills the unrated items to avoid the sparsity issue on recommender systems. The probable similarities among the scholars' historic registers and final grades is based on an Intelligent Recommender System structure was studied by Kanokwan Kongsakun et al [111] where a multi layered neural network with a feed forward configuration is applied with a supervised learning.

Any Machine Learning algorithm that includes a neural network with a feed forward configuration with n input nodes, two hidden nodes and a single output node learning process can be applied to represent collaborative filtering tasks as demonstrated by Daniel Billsus et al [112]; the presented algorithm is founded on the reduction of the dimensionality reduction applying the Singular Value Decomposition (SVD) of an preliminary user ranking matrix that excludes the necessity for customers to rank shared items with the aim of becoming forecasters for another customer preferences. The neural network is trained with a n singular vector and the average user rating; the output neuron represents the predicted user rating.

Neural Networks have been also implemented in film recommendation systems. A neural network in a feed forward configuration with a single hide layer is used as an organizer application that predicts if a certain program is relevant to a customer using its specification, contextual information and given evaluation was presented by Marko Krstic et al [113]; a TV program is represented by a 24 dimensional attribute vector however the neural network has five input nodes; three for transformed genre, one for type of day and the last one for time of day, a single hidden neuron and two output neurons: one for like and the other for dislike. A neural network that identifies which household follower provided a precise ranking to a movie at an exact time was proposed by Claudio Biancalana et al [114]; the input layer is formed of 68 neurons which correspond to different user and time features and the output layer consists of 3 neurons which represent the different classifiers. An "Interior Desire System" approach introduced by Pao-Hua Chou et al [115] considers that customers may have equivalent interest for specific products if they have close browsing patterns; the neural network classifies users with similar navigation patterns into groups with similar intention behavioural patterns based on a neural network with back propagation configuration and supervised learning.

1.2.4 Neural Networks in Deep Learning

Deep learning applies a neural network with various computing layers that perform several linear and nonlinear transformations to model general concepts in data. Deep learning is under a branch of computer learning that models representations of information. Deep learning is characterized as using a cascade of l-layers of nonlinear computing modules for attribute identification and conversion; each input of every sequential layer is based on the output from the preceding layer. Deep learning learns several layers of models that correlate to several levels of conceptualization; those levels generate a scale of notions where the higher the level, the more abstract concepts are learned.

Deep learning models have been used by Aliaksei Severyn et al [116] in learning to rank rating brief text pairs whose main components are phrases; the method is built using a convolutional neural network structure where the best characterization of text pair sets and a similarity function is learned with a supervised algorithm. The input is a sentence matrix with a convolutional feature map layer to extract patterns; a pooling layer is then added to aggregate the different features and reduce the representation. An attention based deep learning neural network was presented by Baiyang Wang et al [117]; it focuses on different aspects of the input data to include distinct features; the method incorporates different word order with variable weights that changed over the time for the queries and search results where a multi-layered neural network ranks results and provides a listwise learning to rank using a decoder mechanism. Deep Stacking Networks are used by Li Deng et al [118] for information retrieval with parallel and scalable learning; the design philosophy is based on basic modules of classifiers are first designed and then they are combined together to learn complex functions. The output of each Deep Stacking Network is linear whereas the hidden unit's output is sigmoidal nonlinear. Deep learning is also used in Recommender Systems. A deep feature representation was defined by Hao Wang et al [119]; it learns the content information and captures the likeness and implicit association among customers and items where Collaborative filtering is used in a Bayesian probabilistic framework for the rating matrix. A Deep learning approach presented by Ali Mamdouh Elkahky et al [120] assigns items and customers to a vector space model in which the similarity between customers and their favoured products is optimized; the model is extended to jointly learn features of items from different domains and user features according to their Web browsing history and search queries. The deep learning neural network maps two different high dimensional sparse features into low dimensional dense features within a joint semantic space.

1.2.5 User Feedback

User feedback can be explicit by directly asking the user to rank or evaluate the results or implicit by analysing the user behaviour. Diane Kelly et al [211] consider that relevance feedback is normally applied for query expansion during the short term modelling of a user's instantaneous information need and for user profiling during the long term modelling of a user's persistent interests and preferences. Steve Fox et al [212] analyse if there is a connection between explicit ratings of user satisfaction and implicit measures of user interest. In addition they assess the implicit measures that were most strongly correlated with user satisfaction. Thorsten Joachims et al [213] assess the reliability of implicit feedback generated from clickthrough data in Web Search; they concluded that clicks are informative but biased. Filip Radlinski [214] uses clickthrough data to learn ranked retrieval functions for Web search results observing that Web Search users frequently make a sequence of queries with a similar information need. Gawesh Jawaheer et al [215] expose that explicit and implicit feedback presents diverse properties of users' preferences; their combination in a user preference model provides a number of challenges however this can also overcome the issues associated complementing each other, with similar performances despite their different characteristics. Ryen White et al [216] examine the extent to which implicit feedback can act as a substitute for explicit feedback; they hypothesised that implicit and explicit feedback was interchangeable as sources of relevance information for relevance feedback. Douglas Oard [217] et al identify three types of implicit feedback based on examination, retention and reference and suggest two strategies for using implicit feedback to make recommendations based on rating estimation and predicted observations.

1.3 Summary of Contributions

- We have presented a mathematical model for our Intelligent Search Assistant based on the Random Neural Network with Deep Learning clusters. We have associated one neuron or clusters of neurons to each Web result dimension;
- We have proposed a Deep Learning Management Cluster that supervise the performance of other Deep Learning Clusters;
- We have included user feedback on an iterative process from which our ISA learns the perceived user's relevance using independently Gradient Descent or Reinforcement Learning. We have analysed the learning algorithms based on

relevance learning speed and optimum number of learning user iterations. Our ISA learns mostly on its first user iteration with a residual learning on its third user iteration;

- We have proposed an innovative cost function which it is used by our ISA to re-rank the snippets retrieved from different Web search engines, Metasearch engines and academic databases;
- We have developed our ISA in a java application with both client and server platforms;
- We have proposed a new quality definition which combines both relevance and rank. We have quantified relevance using the result order instead of a binary relevant or irrelevant metric. A relevant result on a top position scores more than another relevant result at the bottom of the list;
- We have validated our proposed ISA against other Web search applications and relevance metrics. On average, our ISA outperforms other search engines.

We describe Web Search, including Internet assistants, Web Search Engines, Metasearch Engines, Web Result clustering, Travel services and Citation analysis in Section 2 of this thesis. We present Ranking Algorithms, along with Relevance metrics and Learning to Rank in Section 3. We introduce Recommender systems in Section 4 with the different types, relevance metrics and model. We survey the Random Neural Network with Deep Learning clusters with its different expansions and applications in Section 5; in addition, Neural Networks, Deep Learning and G-Networks are also included. The mathematical model of our Intelligent Search Assistant based on a relational database with its associated result cost function and its dimension learning including independently Gradient Descendent and Reinforcement Learning is presented in Section 6. We have evaluated our ISA with an unsupervised evaluation in Section 7 and one iteration user evaluation to define the Relevant Center Point in Section 8. We have evaluated independently our proposed learning algorithm performance in Section 9 where ISA progressively refines the search results while interacting with the user and different Web search engines. We have validated our Deep Learning structure model and Deep Learning management cluster against other Web search engines in Section 10. Finally, our conclusions are presented in Section 11.

1.4 Summary of Publications

Conferences

- Will Serrano, Erol Gelenbe: An Intelligent Internet Search Assistant Based on the Random Neural Network. Artificial Intelligence Applications and Innovations. 141-153 (2016)
- Will Serrano: A Big Data Intelligent Search Assistant Based on the Random Neural Network. International Neural Network Society Conference on Big Data. 254-261 (2016)
- Will Serrano: The Random Neural Network Applied to an Intelligent Search Assistant. International Symposium on Computer and Information Sciences. 39-51 (2016)
- Will Serrano, Erol Gelenbe: Intelligent Search with Deep Learning Clusters. Intelligent Systems Conference. 254-261 (2017)
- Will Serrano, Erol Gelenbe: The Deep Learning Random Neural Network with a Management Cluster. International Conference on Intelligent Decision Technologies. 185-195 (2017)
- Will Serrano: The Random Neural Network and Web Search: Survey Paper. Surveys. Intelligent Systems Conference (2018)

Journals

- Will Serrano: Smart Internet Search with Random Neural Networks. European Review. 25, 2, 260-272 (2017)
- Will Serrano, Erol Gelenbe: The Random Neural Network in a Neurocomputing Application for Web Search. Neurocomputing (2019)
- Will Serrano, Erol Gelenbe: The Deep Learning Random Neural Network in Web Search. Neurocomputing (2018)

2 Web Search

With the development of the Internet, several applications and services have been proposed or developed to manage the increasingly greater volume of information and data accessible in the World Wide Web.

2.1 Internet Assistants

Internet assistants learn and adapt to variable user's interests in order to filter and recommend information as intelligent agents. These assistants normally define a user as a set of weighted terms which are either explicitly introduced or implicitly extracted from its Web browsing behaviour. Relevance algorithms are determined by a vector space model that models both query and answer as an ordered vector of weighed terms. Web results are the parsed fragments obtained from Web pages, documents or Web results retrieved from different sources. The user provides explicit or implicit feedback on the results considered relevant or interesting, this is then used to adapt the weights of the term set profile.

Intelligent Agents are defined by San Murugesan [1] as a self-contained independent software module or computer program that perform simple and structurally repetitive automated actions or tasks in representation of Web users while cooperating with other intelligent agents or humans. Their attributes are autonomy, cooperation with other agents and learning from interaction with the environment and the interface with users' preferences and behaviour. Oren Etzioni et al [2] propose Intelligent Agents behave in a manner analogous to a human agent with Autonomy, Adaptability and Mobility as desirable qualities; they have two ways to make the Web invisible to the user: by abstraction where the used technology and the resources accessed by the agent are user transparent and by distraction where the agent runs in parallel to the Web user to perform tedious and complex tasks faster than would be possible for a human alone.

Spider Agent is a metagenetic assistant presented by Nick Z. Zacharis et al [3] to whom the user provides a set of relevant documents where the N highest frequency keywords form a dictionary which is represented as a Nx3 matrix. The first column of the dictionary contains the keywords whereas the second column measures the entire amount of documents that contains the keywords, finally, the third column contains the combined frequency of the specific word over the overall documents. The metagenetic algorithm first creates a population of keyword sets from the dictionary based on three genetic operators: Crossover, Mutation and Inversion; then it creates a population of logic

operators sets (AND, OR, NOT) for each of the first populations. Spider Agent forms different queries by the combination of both populations and searches for relevant documents for each combination. The main concept is that different combinations of words in different queries may result in search engines providing additional different relevant results.

Syskill & Webert is defined by Michael J. Pazzani et al [4]; it helps users to select relevant Web pages on specific topics where each user has a set of profiles, one for each topic, and Web Pages are rated as relevant or irrelevant. Syskill & Webert transforms the source code of the Web Page based on Hyper Text Markup Language (HTML) into a binary feature vector which designates the presence of words using a learning algorithm based on a naive Bayesian classifier. Letizia was proposed by Henry Lieberman [5], it is a Web user interface agent that helps Web browsing. Letizia learns user behaviour and provides with additional interesting Web pages by exploring the current Web page links where the user interest assigned to a Web document is calculated as the reading time, the addition to favourites or the click of a shown link.

WebWatcher was presented by Thorsten Joachims et al [6], it is a Web tour guide agent that provides relevant Web links to the user while browsing the Web; it acts as a learning apprentice that observes and learns interest from its user actions when select relevant links. WebWatcher uses Reinforcement Learning where the reward is the frequency of each the searched terms within the Web Page recommending Web pages that maximize the reward path to users with similar queries.

Lifestyle Finder, defined by Bruce Krulwich [7], is an agent that generates user profiles with a large scale database of demographic data. Users and their interests are grouped by their input data according to their demographic information. Lifestyle Finder generalizes user profiles along with common patterns within the population; if the user data corresponds to more than one cluster, the demographic variables whose estimates are close with the entire corresponding groups generate a limited user profile. The demographic feature that best distinguishes the corresponding groups is utilized to ask the Web user for additional details where the final group of corresponding clusters is obtained after several user iterations.

2.2 Web Search Engines

Internet assistants have not been practically adopted by Internet users as an interface to reach relevant information; instead Web search engines are the preferred option as the portal between users and the Internet due their simplicity. Web search engines are software applications that search for information in the World Wide Web while retrieving data from Web sites and Online databases or Web directories. Web search engines have already crawled the Web, fetched its information and indexed it into databases so when the user types a query, relevant results are retrieved and presented promptly (Fig. 1).

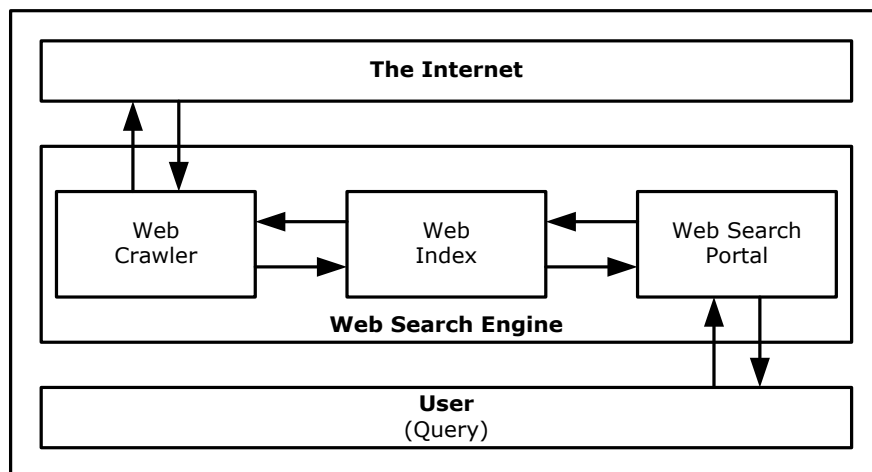


Figure 1: Web search engine architecture

The main issues of Web search engines are result overlap, rank relevance and adequate coverage for both sponsored and non-sponsored results as stated by Amanda Spink et al [8]. Web personalization builds user's interest profile by using their browsing behaviour and the content of the visited Web pages to increase result extraction and rank efficiency in Web search. A model presented by Alessandro Micarelli [9] represents a user needs and its search context is based on content and collaborative personalization, implicit and explicit feedback and contextual search. A user is modelled by Nicolaas Matthijs et al [10] as a set of terms and weights related to a register of clicked URLs with the amount of visits to each, and a set of previous searches and Web results visited; this model is applied to reorder the Web search results provided by a non-personalized Web search engine. Web queries are associated by Fang Liu et al [11] to one or more related Web page features and a group of documents is associated with each feature; a document is both associated with the feature and relevant to the query. These double profiles are merged to attribute a Web query with a group of features that define the user's search relevance; the algorithm then expands the query during the Web search by using the group of categories. Filip Radlinski et al [12] present different methods to improve

personalized web search based on the increment of the diversity of the top results where personalization comprises the re-ranking of the first N search results to the probable order desired by the user and query reformulations are implemented to include variety. The three diversity methods proposed are: the most frequent selects the queries that most frequently precede the user query; the maximum result variety chooses queries that have been frequently reformulated but distinct from the already chosen queries, and finally, the most satisfied selects queries that usually are not additionally reformulated but they have a minimum frequency.

Spatial variation based on information from Web search engine query records and geolocation methods was included in Web search queries by Lars Backstrom et al [13] to provide results focused on marketing and advertising, geographic information or local news; accurate locations are assigned to the IP addresses that issue the queries. The center of a topic is calculated by the physical areas of the users searching for it where a probabilistic model calculates the greatest probability figure for a geographic center and the geographical dispersion of the query importance. Aspects for a Web query are calculated by Fei Wu et al [14] as an effective tool to explore a general topic in the Web; each aspect is considered as a set of search terms which symbolizes different information requests relevant to the original search query. Aspects are independent to each other while having a high combined coverage; two sources of information are combined to expand the user search terms: query logs and mass collaboration knowledge databases such as Wikipedia.

2.3 Metasearch Engines

Metasearch engines were developed based on the concept that single Web search engines were not able to crawl and index the entire Web. While Web search engines are useful for finding specific information, like home pages, they may be less effective with a comprehensive search or wide queries due their result overlap and limited coverage area. Metasearch engines try to compensate their disadvantages by sending simultaneously user queries to different Web search engines, databases, Web directories and digital libraries and combining their results into a single ranked list (Fig. 2, Fig. 3). The main operational difference with Web search engines is that Metasearch engines do not index the Web.

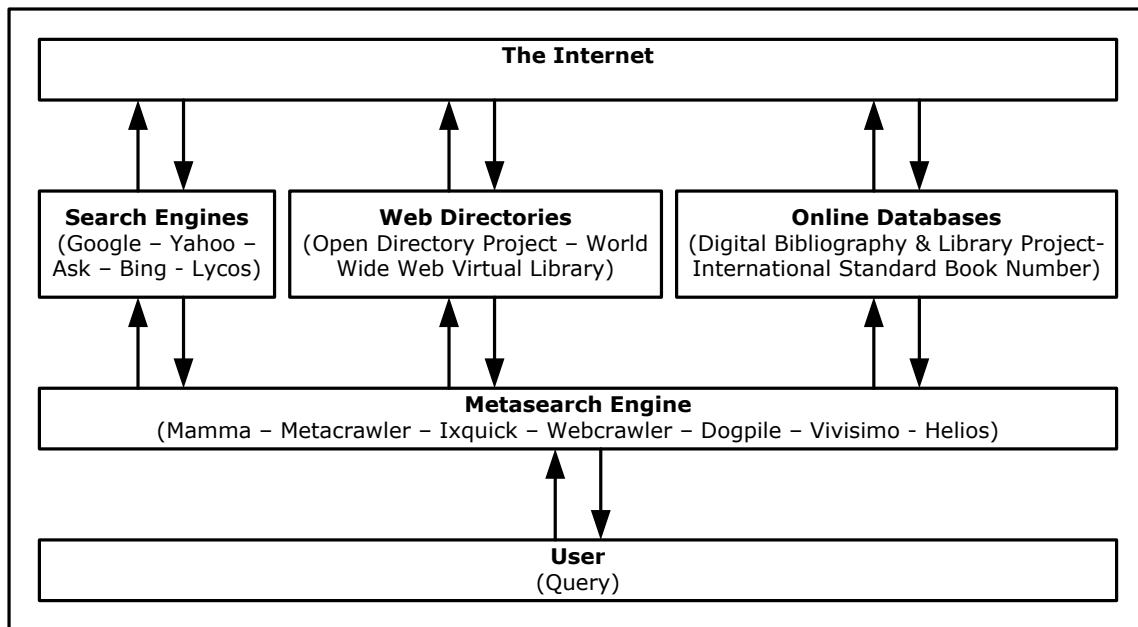


Figure 2: Metasearch services model

There are challenges for developing Metasearch engines as described by Weiyi Meng et al [16]: Different Web search engines are expected to provide relevant results which have to be selected and combined. Different parameters were considered by Manoj, M et al [17] when developing a Metasearch engine: functionality, working principles including querying, collection and fusion of results, architecture and underlying technology, growth, evolution and popularity. Numerous metasearch architectures were found by Hossein Jadidoleslami [18] such as Helios, Tadpole and Treemap with different query dispatcher, result merger and ranking configurations. Helios was presented by Antonio Gulli et al [19]; it is an open source metasearch engine that runs above different Web Search Engines where additional ones can be flexibly plugged into architecture. Helios retrieves, parses, merges, and reorders results given by the independent Web search engines.

An extensive modular metasearch engine with automatic search engine discovery was proposed by Zonghuan Wu [20]; it incorporates a numerous number of autonomous search engines; it is based on three components: "automatic Web search engine recognition, automatic Web search engine interconnection and automatic Web search result retrieval". The solution crawls and fetches Web pages choosing the Web Search Engine ones, which once discovered, are connected by parsing the HTML source code, extracting the form parameters and attributes and sending the query to be searched. Finally, URLs or snippets provided by the different Web Search Engines are extracted and displayed to the Web user.

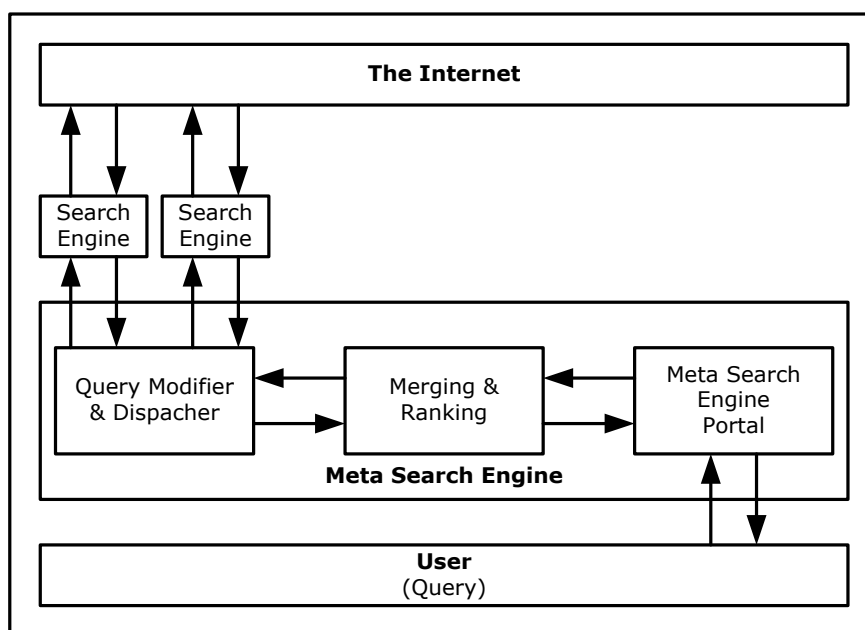


Figure 3: Metasearch engine architecture

There are several rank aggregation metasearch models to combine results in a way that optimizes their relevance position based on the local result ranks, titles, snippets and entire Web Pages. The main aggregation models were defined by Javed A. Aslam et al [21] as: Borda-fuse is founded on the Borda Count, a voting mechanism on which each voter orders a set of Web results; the Web result with the most points wins the election. The Bayes-fuse is built on a Bayesian model of the probability of Web result relevance. The difference between the probability of relevance and irrelevance respectively determines relevance of a Web Page based on the Bayes optimal decision rule. Rank aggregation was designed against spam, search engine commercial interest and coverage by Cynthia Dwork [22].

The use of the titles and associated snippets presented by Yiyao Lu et al [23] produce a higher success than parsing entire Web pages where the effective algorithm of a metasearch engine for Web result merging outperforms the best individual Web search engine. A formal approach to normalize scores for metasearch by balancing the distributions of scores of the first irrelevant documents was presented by R. Manmatha et al [24], it is achieved by using two different methods: the distribution of scores of all documents and the combination of an exponential and a Gaussian distribution to the ranks of the documents where the developed exponential distribution is used as an approximation of the irrelevant distribution.

Metasearch engines provide personalized results using different approaches as presented by Leonidas Akritidis et al [25]. The first method provides distinct results for users in separate geographical places where the area information is acquired for the user and Web page server. The second method has a ranking method where the user can establish the relevance of every Web search engine and adjusts its weight in the result ranking in order to obtain tailored information. The third method analyses Web domain structures of Web pages to enable the user the possibility to limit the amount of Web pages with close subdomains.

Results from different databases present in the Web were also merged and ranked by Weiyi Meng et al [26] with a database representative that is highly scalable to a large number of databases and representative for all local databases. The method assigns only a reduced but relevant number of databases for each query term where single term queries are assigned the right databases and multi-term queries are examined to extract dependencies between them in order to generate phrases with adjacent terms. Metasearch performance against Web search has been widely studied by B.T. Sampath Kumar et al [27] where the search capabilities of two metasearch engines, Metacrawler and Dogpile, and two Web search engines, Yahoo and Google, are compared.

2.4 Web result clustering

Web result clustering groups results into different topics or categories in addition to Web search engines that present a plain list of result to the user where similar topic results are scattered. This feature is valuable in general topic queries because the user gets the theme bigger picture of the created result clusters (Fig. 4). There are two different clustering methods; pre-clustering, as defined by Oren Zamir et al [28], calculates first the proximity between Web pages and then it assigns the labels to the defined groups whereas post-clustering, as proposed by Stanislaw Osinski et al [29], discovers first the cluster labels and then it assigns Web pages to them. Web clustering engines shall provide specific additional features, as established by Claudio Carpineto et al [30], in order to be successful: fast subtopic retrieval, topic exploration and reduction of browsing for information. The main challenges of Web clustering are overlapping clusters, shown by Daniel Crabtree et al [31], precise labels as presented by Filippo Geraci [32], undefined cluster number, as experienced by Hua-Jun Zeng et al [33], and computational efficiency.

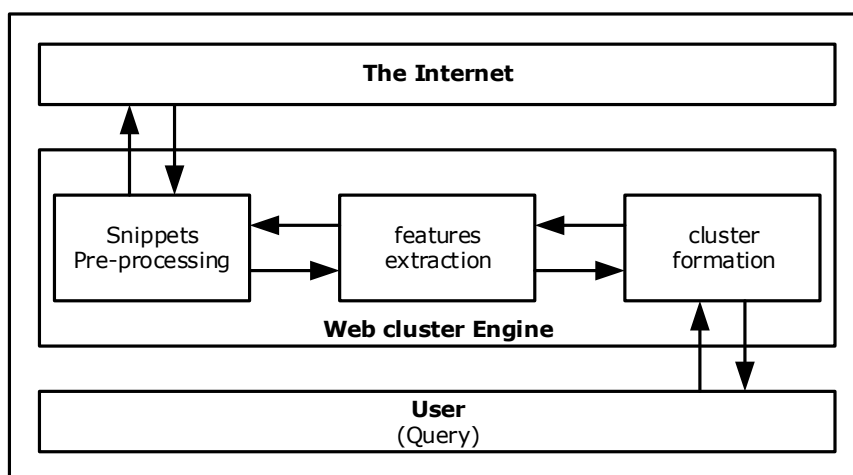


Figure 4: Web Cluster Engine Architecture

Suffix Tree Clustering (STC) was presented by Oren Zamir et al [28], it is an incremental time method which generates clusters using shared phrases between documents. STC considers a Web page as an ordered sequence of words and uses the proximity between them to identify sets of documents with common phrases between them in order to generate clusters. STC has three stages: document formatting, base clusters identification and its final combination into clusters.

Lingo Algorithm, proposed by Stanislaw Osinski [29], finds first clusters utilizing the Vector Space Model to create a "TxD term document matrix where T is the number of unique terms and D is the number of documents". A Singular Value Decomposition method is used to discover the relevant matrix orthogonal basis where orthogonal vectors correspond to the cluster labels. Once clusters are defined; documents are assigned to them.

A cluster scoring function and selection algorithm, defined by Daniel Crabtree [31], overcomes the overlapping cluster issue; both methods are merged with Suffix Tree Clustering to create a new clustering algorithm called Extended Suffix Tree Clustering (ESTC) which decreases the amount of clusters and defines the most useful ones. This cluster scoring function relies on the quantity of different documents in the cluster where the selection algorithm is based on the most effective collection of clusters that have marginal overlay and maximum coverage; this makes the most different clusters where most of the documents are covered in them.

A cluster labelling strategy, established by Filippo Geraci et al [32], combines intra-cluster and inter-cluster term extraction where snippets are clustered by mapping them into a

vector space based on the metric k -center assigned with a distance function metric. To achieve an optimum compromise between defining and distinctive labels, prospect terms are selected for each cluster applying an improved model of the information Gain measurement. Cluster labels are defined by Hua-Jun Zeng et al [33] as a supervised salient phrase ranking method based on five properties: "sentence frequency / inverted document frequency, sentence length, intra cluster document similarity, cluster overlap entropy and sentence independency" where a supervised regression model is used to extract possible cluster labels and human validators are asked to rank them.

Cluster diversification is a strategy used in ambiguous or broad topic queries that takes into consideration its contained terms and other associated words with comparable significance. A method proposed by Jiyin He et al [34] clusters the top ranked Web pages and those clusters are ordered following to their importance against the original query however diversification is limited to documents that are assigned to top ranked clusters because potentially they contain a greater number of relevant documents. Tolerance classes approximate concepts in Web pages to enhance the snippets concept vector as defined by Chi Lang Ngo et al [35] where a set of Web pages with similar enhanced terms are clustered together. A technique that learns relevant concepts of similar topics from previous search logs and generates cluster labels was presented by Xuanhui Wang [36], it clusters Web search results based on these as the labels can be better than those calculated from the current terms of search result.

Cluster hierarchy groups data over a variety of scales by creating a cluster tree. A hierarchy of snippets, defined by Zhao Li [37], is based on phrases where the snippets are assigned to them; the method extracts all salient phrases to build a document index assigning a cluster per salient phrase; then it merges similar clusters by selecting the phrase with highest number of indexing documents as the new cluster label; finally, it assigns the snippets whose indexing phrases belong to the same cluster where the remaining snippets are assigned to neighbours based on their k -nearest distance.

Web search results are automatically grouped through a Semantic, Hierarchical, Online Clustering (SHOC) in Dell Zhang et al [38] experiments; a Web page is considered as a string of characters where a cluster label is defined as a meaningful substring which is both specific and significant. The latent semantic of documents is calculated through the analysis of the associations between terms and Web pages. Terms assigned with the same Web page should be close in semantic space; same as the Web pages assigned with the same terms. Densely allocated terms or Web pages are close to each other in

semantic space, therefore they should be assigned to the same cluster. Snake is a clustering algorithm developed by Paolo Ferragina et al [39], it is based on two information databases; the anchor text and link database assigns each Web document to the terms contained in the Web document itself and the words included in the anchor text related to every link in the document, the semantic database ranks a group of prospect phrases and chooses the most significant ones as labels. Finally, a Hierarchical Organizer combines numerous base clusters into few super clusters. A query clustering approach was presented by Ricardo A. et al [40], it calculates the relevance of Web pages using previous choices from earlier users, the method applies a clustering process where collections of semantically similar queries are identified and the similarity between a couple of queries is provided by the percentage of shared words in the clicked URL within the Web results.

2.5 Travel Services

Originally, travel service providers, such as airlines or hotels, used global distribution systems to combine their offered services to travel agencies (Fig. 5). Global distribution systems required high investments due to their technical complexity and physical dimensions as they were mainframe based infrastructure; this generated the monopoly of a few companies that charged a high rate to the travel services providers in order to offer their services to the travel agencies as described by Athina Sismanidou et al [41]. With this traditional model, customers could purchase travel provider services directly at their offices or through a travel agent. This scenario has now been changed by two factors presented by Nelson F. Granados et al [42]: the Internet has enabled e-commerce; the direct accessibility of customers to travel services providers' information on real time with the availability of online purchase and higher computational servers and software applications can implement the same services as the global distribution systems did, at a lower cost.

As a consequence, Hannes Werthner [43] described the new players that have entered to this scenario; Software database applications such as ITA Software, or G2 SwitchWorks use high computational server applications and search algorithms to process the data obtained from travel services providers; they are the replacement of the global distribution systems.

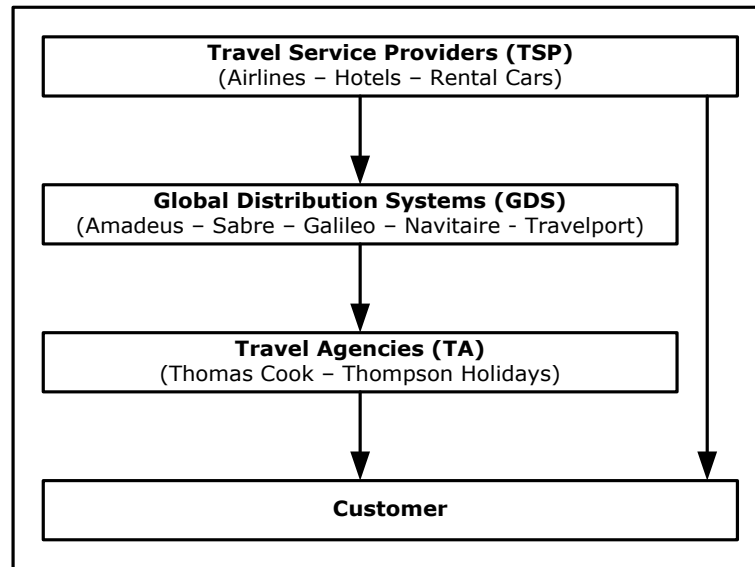


Figure 5: Traditional travel services model

Online travel agents, such as Expedia or TripAdvisor, are traditional travel agents that have automatized the global distribution systems interface; customers can interact with them directly through the Internet and buy the final chosen product. Metasearch engines like Cheapflights or Skyscanner among many others use the Internet to search customers' travel preferences within travel services providers and online travel agents however customers can not purchase products directly; they are mainly referred to the supplier Web site (Fig. 6).

Other players that have an active part in this sector are Google and other Web Search Engines as shown by Bernard J. Jansen et al [44]; they provide search facilities that connect directly travel service providers with consumers bypassing global distribution systems, software distribution systems, Metasearch engines and online travel agents. This allows the direct product purchase that can reduce distribution costs due to a shorter value chain.

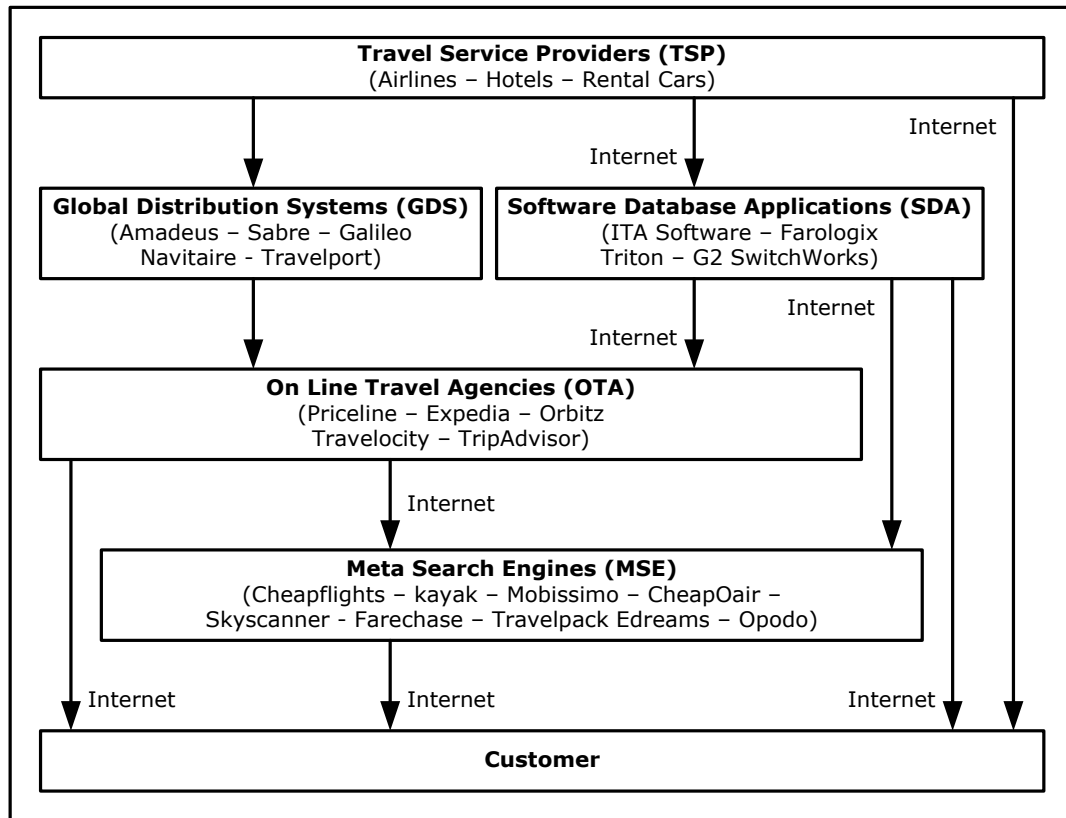


Figure 6: Online travel services model

Travel ranking systems, as described by Anindya Ghose et al [45], also recommend products and provide the greatest value for the consumer's budget with a crucial concept where products that offer a higher benefit are ranked in higher positions.

With the integration of the customer and the travel service provider through the Internet, different software applications have been developed to provide extra information or to guide through the purchase process based on user interactions as presented by Zheng Xiang et al [46]. Travel related Web interfaces help users to make the process of planning their trip more entertaining and engaging while influencing users' perceptions and decisions as described by Bernhard Kruepl et al [47]. There are different information search patterns and strategies proposed by Nicole Mitsche [48] within specific tourism domain web search queries based on search time, IP address, language, city and key terms keywords.

2.6 Citation Analysis

Citation analysis consists on the access and referral of research information including articles, papers and publications. Its importance is based on the measurement of the value or relevance of personal academic work; it is becoming more respected since its evaluation impacts on other decisions such as promotion, scholarships and research funds. The Journal Impact Factor is calculated by Lokman I. Meho et al [49] as “the number of citations received in the current year to articles published in the two preceding years divided by the total number of articles published in the same two years”. It is based on the concept that high impact journals will only try to publish very relevant work therefore they will be used for outstanding academics. These traditional measurements are peer review based on human judgment however it is time consuming and assessors may be biased by personal judgments such as the quality of journal or the status of university instead of competent and impartial reviews.

The Web has provided the possibility of other alternative citation measures making change to the citation monopoly because of the accessibility it provides. New online platforms presented by Judit Bar-Ilan [50] such as Scopus, Web of Science and Google Scholar provide other measures that may represent more accurately the relevance of an academic contribution. Bibliometric indicators proposed by Judit Bar-Ilan [51] evaluate the quality of the publication based on empirical values; mostly number of citations or number of downloads. They are not only quicker to obtain but they also provide a wider coverage because they also retrieve results from relevant academic work published on personal or academic Web pages and open journals based on conference papers, book chapters or theses. In addition, academics are personally interested to artificially influence ranking methods by “optimizing” the appearance of their research publications to show them in higher ranks with the purpose to extend their reader audience in order to get cited more as exposed by Joeran Beel et al [52].

The h-index defined by Jorge E. Hirsch [53] is applied to qualify the contribution of individual scientific research work; “An academic has an h-index of h if the x publications have individually received at least h citations and the other (x-h) publications have up to h citations”. The h-index has become popular among researchers because it solves the disadvantages of the journal impact factors and it has also a good correlation with the citation analysis database. It is simple and fast to calculate utilizing databases like Google Scholar, Scopus or Web of Science however it suffers some disadvantages [49]; it does not differentiate negative citations from positive ones so both are equally taken into

account, it does not consider the entire amount of citations an author has collected and it disadvantages a reduced but highly-cited paper set very strongly.

Google Scholar is a free Web academic search engine that registers the metadata or the complete text of academic articles from different databases and journals. The result relevance is calculated by a ranking algorithm that combines the term occurrence, author, publications and citations weights following the work of Jöran Beel et al [54]. Google Scholar has received both good and bad critics since it started in 2004 as Dirk Lewandowski [55] stated; the lack of transparency in the relevance assessment of the information due its automatic process of retrieving, indexing and storing information has been criticized; however the more transparent its evaluation algorithms, the easier it becomes to influence its metrics. Google Scholar has been widely analysed and studied by the research community against other online databases by William H. Walters [56] and Anne-Wil Harzing et al [57].

A hybrid recommender system based on content-based and collaborative-based relevance was presented by Bela Gipp [58]; it expands the method of the traditional keyword or key term search by adding citation examination, author examination, source examination, implicit ratings and explicit marks. Two ranking methods are applied: “in-text citation frequency examination” evaluates the occurrence in which a publication is “cited within the citing document” and “in-text citation distance examination” determines the similarity between references within a publication by evaluating the word separation.

3 Ranking

Ranking is the core process of many information retrieval applications including Web search engines and Recommender systems for two main reasons: the user evaluation of quality or performance from a customer point of view and sponsored adverts based on ranking from a commercial perspective. There are several methods and techniques for ranking Web pages and products that exploit different algorithms built on the analysis of the Web page content, the anchor text of the links between pages or the network structure of the World Wide Web hyperlinked environment.

3.1 Ranking Algorithm

Web search engine evaluation comprises the measurement of the quality and services they provider to users; it is also used to compare their different search capabilities. Automatic evaluation analyses the different presented URLs whereas human evaluation is based on the measurement of result relevance; while the latest one is more precise; it takes more effort therefore more expensive as it is normally done by surveys. Other methods that combine both automatic and user evaluation are applications that monitor user activities such as click-throughs or time spent of each Web site. "Dynamic ranking or query-dependent ranking" improves the order of the results returned to the user based on a query while "static ranking or query-independent" calculates the relevance of the Web pages to different topics.

There are different methods to assess Web pages relevance.

3.1.1 HTML Properties

The first method focuses on the HTML properties of the Web page including the type of text, section, size, position, anchor text and URL. Web spammers base their strategy mostly in the design of the HTML properties of the Web page to get an artificial better ranking from Web search engines.

3.1.2 TF-IDF

The second method is founded on the inspection of the Web page text itself and how the words in the query relate within it. TF-IDF or "Term Frequency and Inverse Document Frequency" applies the vector space model with a scoring function where the importance of a term increases if it is repeated often in a document however its weight decreases if the same term appears multiple times on numerous documents as it is considered less descriptive.

$$TF-IDF(t,d,D)=TF(t,d) \times IDF(t,D)=[1+\log f_{t,d}] \times \left[\log \left(1+\frac{N}{d_t} \right) \right] \quad (1)$$

where t is the keyword, d is the Web page, $f_{t,d}$ the amount of occurrences that keyword t is contained within the Web page d , N is the entire amount of Web pages and d_t is the quantity of Web pages that include the keyword t .

A probabilistic TF-IDF retrieval model proposed by Ho Chung Wu et al [59] emulates the decision making of a human brain for two categories of relevance: “local relevance” is first applied to a particular Web page location or section whereas the “wide relevance” spans to cover the whole Web page; the method then merges the “local relevance” decisions for each position as the Web page “wide relevance” decision. Document Transformation adapts the Web page vector close to the query vector by adding or removing terms; an example to long term incremental learning presented by Charles Kemp [60] is based on a search log that stores information about click thoughts Web users have browsed among the Web result pages from a Web Search engine and Web Pages selected by the user. Content-Time-based ranking is a Web search time focus ranking algorithm proposed by Peiquan Jin [61] that includes both “text relevance” and “time relevance” of Web pages. “Time relevance” is the interval computed between the query and the content or update time; where every term in a Web page is assigned to an explicit content time interval, finally, the time focus TF-IDF value of each keyword is calculated to obtain the overall rank.

3.1.3 Link Analysis

The third method is the link analysis which it is founded on the relation between Web pages and their Web graph; link analysis associates Web pages to neural network nodes and links to neural network edges.

In-Degree was defined by Taher H. Haveliwala [62], it was the first link analysis ranking algorithm; it ranks pages according to their popularity which it is calculated by the sum of links that point to the page.

$$Degree(p_i) = \sum_{p_j \in M(p_i)} p_j \quad (2)$$

where $p_1, p_2, \dots, p_N$ correspond to the Web pages to be ranked and $M(p_i)$ defines the collection of Web documents pointing to p_i .

Page Rank was proposed by Sergey Brin et al [63]; it is founded on the concept that a Web document has a top "Page Rank" if there are many Web pages with reduced "Page Rank" or a few Web pages with elevated "Page Rank" that link to it. Page rank algorithm provides a likelihood distribution that represents the probability that a Web user will reach to the specific Web page arbitrarily following the links.

$$\text{PageRank}(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{\text{PageRank}(p_j)}{L(p_j)} \quad (3)$$

where $p_1, p_2, \dots, p_N$ correspond to the Web pages to be ranked, d is the dumping factor, $M(p_i)$ defines the collection of Web documents pointing to p_i , $L(p_j)$ is the amount of external connections from page p_j and N the entire quantity of Web pages.

HITS algorithm or Hyperlink Induced Topic Search algorithm was presented by Jon M. Kleinberg [64], it assigns relevance founded on the authority concept and the relation among a group of relevant "authoritative Web pages for a topic" and the group of "hub Web pages that link to many related authorities" in the Web link arrangement. The hyperlink structure among Web pages is analysed to formulate the notion of authority; the developer of Web document p , when inserts a hyperlink to Web document q , provides some relevance or authority on q . Hubs and authorities have a reciprocally strengthening association: "a relevant hub is a Web page that links to many relevant authorities; a relevant authority is a Web page which is linked by numerous relevant hubs".

The authority value of a Web document is the addition of every hub value of Web documents linking to it:

$$\text{auth}(p) = \sum_{i=1}^n \text{hub}(i) \quad (4)$$

where p the relevant authority Web document, i is a Web document linked to p and n is the entire quantity of Web documents linked to p .

The hub value of a Web document is the addition of total number of authorities' values of Web documents linking to it:

$$\text{hub}(p) = \sum_{i=1}^n \text{auth}(i) \quad (5)$$

where p is the relevant authority Web document, i is a Web document that p links to and n is the entire quantity of Web documents p connects to.

SALSA algorithm or “Stochastic Approach for Link-Structure Analysis” was developed by Ronny Lempel [65]; it combines Page Rank and HITS by taking a random walk alternating between hubs and authorities. Two random walks are assigned two scores per Web Page; it is based on the in link and out link that represent the proportion of authority and hub respectively.

The hub array is expressed in the below sum:

$$h_{i,j} = \sum_{\{k | (i_h, k_a), (j_h, k_a) \in G\}} \frac{1}{\deg(i_h)} \times \frac{1}{\deg(k_a)} \quad (6)$$

The authority array is expressed in the below sum:

$$a_{i,j} = \sum_{\{k | (k_h, i_a), (k_h, j_a) \in G\}} \frac{1}{\deg(i_a)} \times \frac{1}{\deg(k_h)} \quad (7)$$

where Web document h links to mutually documents i and j and $\deg$ is the amount of hyperlinks connecting to a page.

QS page-rank is a query-sensitive algorithm proposed by Tao Wen-Xue [66] that combines both global scope and local relevance. Global Web Page importance is calculated by using general algorithms such as Page Rank; Local query sensitive relevance is calculated by a voting system that measures the relation between the top Web pages retrieved.

The priority of a Query Sensitiveness rank (QS) is defined where document 1 is ordered on front of document 2:

$$(QS_1 > QS_2) \ || \ (QS_1 == QS_2 \ \&\& \ PR_1 > PR_2) \quad (8)$$

The priority of a Global Importance Rank (PR) is defined where document 1 is ordered on front of document 2:

$$(PR_1 > PR_2) \ || \ (PR_1 == PR_2 \ \&\& \ QS_1 > QS_2) \quad (9)$$

where QS_1 and QS_2 are the Query sensitive values and PR_1 and PR_2 are the Global importance values for document 1 and document 2 respectively.

DistanceRank was defined by Ali Mohammad Zareh Bidoki [67], it is an iterative algorithm built on Reinforcement Learning; distance between pages is counted as punishment where distance is established as the amount of links among two Web Pages. The objective of the method is the reduction of the punishment or distance therefore a Web document with reduced “distance” has a greater rank. The main concept is that as correlated Web pages are connected between them; the distanced built method finds relevant Web pages quicker.

$$\text{distance}_n[j] = (1-\alpha) \cdot \text{distance}_{n-1}[j] + \alpha \cdot \min_i (\log(O[i]) + \text{distance}_{n-1}[i]) \quad (10)$$

where i is the collection of Web documents linking to j and $O[i]$ is the amount of outbound hyperlinks in Web document i and α is the learning rate of the user.

DING is a Dataset Ranking algorithm presented by Renaud Delbru [68] that calculates dataset ranks; it uses the connections among them to combine the resulting scores with internal semantic based ranking strategies. DING ranks in three stages: first, a dataset rank is calculated by an external dataset hyperlink measurement on the higher layer, then for each dataset, local ranks are calculated with hyperlink measurements between the inner entities and finally the relevance of each dataset entity is the combination of both external and internal ranks.

The internal rank score of entity e is calculated as:

$$r(e) = w_{\sigma,i,j} = \text{LF}(L_{\sigma,i,j}) \times \text{IDF}(\sigma) = \frac{|L_{\sigma,i,j}|}{\sum_{L_{r,i,k}} |L_{r,i,k}|} \times \log \frac{N}{1 + \text{frequency}(\sigma)} \quad (11)$$

where $L_{\sigma,i,j}$ is a linkset (σ is term, i is the source and j is the target), N represents the amount of datasets and $\text{freq}(\sigma)$ defines the rate of appearance of the term σ in the group of datasets.

The dataset rank score is calculated as:

$$r^k(D_j) = \alpha \sum_{L_{\sigma,i,j}} r^{k-1}(D_i) w_{\sigma,i,j} + (1-\alpha) \frac{|E_{D_j}|}{\sum_{D \in G} |E_D|} \quad (12)$$

where D is a dataset and E_{D_j} is the set of nodes within D_j

The final global score of entity e for dataset D is:

$$r_g(e) = r(e) \cdot r(D) \quad (13)$$

where $r(e)$ and $r(D)$ are the internal rank entities and dataset rank score respectively

ExpertRank was proposed by Jian Jiao [69]; it is an online community and discussion group expert ranking algorithm that integrates a vector space method to calculate the expert subject importance and a method similar to Page Rank algorithm to evaluate topic authority. Expert subject relevance score is calculated as a likeness of the candidate description and the provided request where the candidate description is generated by combining all the comments in the topic conversation group and the authority rating is computed according to the user interaction graph.

$$\text{Expert}(c,q)=i(\text{RE}(c,q),\text{AU}(c)) \quad (14)$$

where c is the candidate, q is the query, i is the integration method, $\text{RE}(c,q)$ represents the relevance rating among the query q and candidate c 's knowledge and $\text{AU}(c)$ represents the authority rating.

3.2 Relevance Metrics

Performance can be described as the display of relevant, authoritative, updated results on first positions. Retrieval performance is mostly evaluated on two parameters: precision is the percentage of applicable results within the provided Web document list and recall is the percentage of the entire applicable results covered within the provided Web document list. Due to the intrinsic extent of the Web, where it is difficult to measure the quantity of all Web pages and the fact that users only browse within the first 20 results; precision is widely used as a performance measurement. In addition, another factor to consider is that Web search engines are different and therefore they perform differently as exposed by Mildrid Ljosland [70]; some may perform better than others for different queries and metrics.

There are different parameters to assess the performance of recommender systems:

- Precision evaluates the applicability of the first N results of the rating result set in relation to a query:

$$P@N = \frac{\text{Number of relevant documents in top N results}}{N} \quad (15)$$

- Recall represents the applicability of the first N results of the rating result set in relation to the complete result set:

$$R@N = \frac{\text{Number of relevant documents in top N results}}{S} \quad (16)$$

where S is the result set.

- The Mean Absolute Error (MAE) evaluates the error between user predictions and user ratings:

$$MAE = \frac{\sum_{d=1}^N |P_{ui} - R_{ui}|}{N} \quad (17)$$

- The Root Squared Mean Error (RSME) measures the differences between predictions and user ratings:

$$RME = \frac{\sqrt{\sum_{d=1}^N (P_{ui} - R_{ui})^2}}{N} \quad (18)$$

where P_{ui} is the forecasted ranking for u on product i, R_{ui} is the current rank for u on product i and N the entire amount of products ranked by the user u.

- F-score combines Precision and Recall in an evenly weighted metric:

$$F_{\text{measure}} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (19)$$

- The Average Precision calculates the mean figures of $P@n$ over the number n of retrieved Web pages:

$$AP = \frac{\sum_{n=1}^N P@n * \text{rel}(n)}{\text{Number of relevant documents for this query}}$$

(20)

where N is the total amount of retrieved Web pages in the result list, and $rel(n)$ represents that the n -th Web page is either applicable or not to the query with values 1 or 0.

- The Mean Average Precision measures the balance of average precision within a collection of Q Web searches:

$$MAP = \frac{\sum_{q=1}^Q AP(q)}{Q} \quad (21)$$

- The Normalized Discounted Cumulative Gain uses a rated relevance metric to measure the relevance of a result that uses its rank within the Web page set:

$$NDCG@N = Z_N \sum_{j=1}^N \frac{2^{R(j)} - 1}{\log_2(1+j)} \quad (22)$$

where $R(j)$ is the rated applicability of the result at position j within the Web page set, Z_N is the standardization parameter that assures the optimum $NDCG@N$ matches to 1. For binary ranking then $R(j)$ is fixed to 1 when the result at position j is relevant to the Web search and is fixed to 0 when the Web page is irrelevant.

- TREC average precision includes rank position on the performance evaluation; it is defined at a cutoff N :

$$TSAP@N = \frac{\sum_{i=1}^N r_i}{N} \left\{ \begin{array}{ll} r_i = 1/i & \text{if the } i_{th} \text{ ranked result is relevant} \\ r_i = 0 & \text{if the } i_{th} \text{ ranked result is not relevant or duplicate} \end{array} \right\} \quad (23)$$

- The Reciprocal Rank of a Web search is defined as the multiplicative inverse applied to the rank of the first relevant result:

$$RR = \frac{1}{rank_i} \quad (24)$$

where $rank_i$ is the order of the first relevant result.

- The Mean Reciprocal Rank corresponds to averaged value of the RR (Reciprocal Rank) over a collection of Q searches:

$$MRR = \frac{\sum_{q=1}^Q RR(q)}{Q} \quad (25)$$

where $RR(q)$ is the Reciprocal Rank associated to the search q and Q is the total collection of searches.

- The Expected Reciprocal Rank is based on a cascaded model which penalizes documents which are shown below very relevant documents:

$$ERR = \sum_{r=1}^N \frac{1}{r} \prod_{i=1}^{r-1} (1 - R_i) \cdot R_r \quad (26)$$

where N is the amount of Web pages in the ranking, r is the position at which the Web user stops its search, R measures the probability of a click which can be interpreted as the relevance of the snippet.

Different research analyses have studied the efficiency of Web search engines by Michael D. Gordon et al [71] and the level of overlap among the Web pages presented by the search engines. Two new measurements to assess Web search engine performance are proposed by Liwen Vaughan [72]: the relevance of result rating by a Web search engine is assessed by the association among Web and human rating and the capability to retrieve applicable Web pages as the percentage of first ranked within the retrieved Web pages. Four relevance metrics are applied by Longzhuang Li [73] to calculate the quality of Web search engines; three of them are adapted from the Information Retrieval: Vector Space method, Okapi likeness value and Coverage concentration rating and a new method which is developed from the human interaction to take human expectations into account based on a raw score and a similarity score.

The Text REtrieval Conference (TREC), as presented by Amit Singhal et al [74], is supported by the "National Institute of Standards and Technology (NIST)" and "U.S. Department of Defence" to sponsor studies in the information and data retrieval research group; TREC provides the framework for extensive evaluation applied to different text retrieval methods. Effectiveness of TREC algorithms against Web search engines has been compared by Amit Singhal et al [74] where tested algorithms were developed with Okapi or Smart based on the term weighting methods (TF-IDF). Web search engines are

evaluated against a series of relevance metrics by David Hawking et al [75] and [76]: precision, TREC based mean reciprocal rank (MRR) and the TREC style average precision (TSAP) obtained from human evaluations for the top 20 Web results are displayed.

Web search evaluation methods were assigned into eight different categories by Rashid Ali et al [77]: assessment founded on relevance, assessment founded on ranking, assessment founded on user satisfaction, assessment founded on size and coverage of the Web, assessment founded on dynamics of Search results, assessment founded on few relevant and known items, assessment based on specific topic or domain and automatic assessment.

Different Automatic Web search engine evaluation methods have been proposed by Fazli Can et al [78] and Rashid Ali et al [79] to reduce human intervention. The algorithm first submits a query to the different Web search engines where the top 200 Web pages of every Web search engine are retrieved; then it ranks the pages in relation to their likeness to the Web user information requirements; the model is built on the vector space method for query-document matching and ranking using TF-IDF. After ranking; the method lists as relevant the Web pages that are in the first 20 documents retrieved by each Web search engine with the top 50 ranked results. An automatic C++ application presented by Jinbiao Hou [80] compares and analyses Web result sets of several Web search engines; the research analyses the URL coverage and the URL rank where recall and accuracy are used as relevance metrics however there is not description regarding the method that decides URL relevance. A technique for evaluating Web search engines robotically presented by Abdur Chowdhury et al [81] is founded in the way they order already rated Web search results where a search query is transmitted to different Web search engines and the rank of retrieved Web results from is measured against the document that has already been paired with that query.

Web search engine evaluation has been examined from a webometric perspective by Mike Thelwall [82] using estimates of success values, amount of URL provided, amount of domains retrieved and number of sites returned where evaluations are made to assess the retrieval of the most precise and comprehensive Web results from every single Web engine.

3.3 Learning to Rank

Learning to rank is defined as the implementation of semi-supervised or supervised computer learning techniques to generate a ranking model by combining different documents features obtained from training data in an information retrieval system. The training data contains sets of elements with a specified position represented by a numerical score, order or a binary judgment for each element. The purpose of a ranking method is to order new result lists in order to produce rankings similar to the order within the training data. Query document sets are normally defined by quantitative vectors which components are denoted features, factors or ranking signals. The main disadvantages of the learning to rank approach are that the optimization of relevance is based on evaluation measurements without addressing the decision of which measures to use; in addition, learning to rank does not take into consideration that user relevance judgment changes over the time.

The model of learning to rank consists on:

- a set Q of M queries, $Q = \{q_1, q_2, \dots, q_M\}$
- a set D of N Web pages per query q_M , $D = \{d_{11}, d_{12}, \dots, d_{NM}\}$
- a set X of M feature vectors per each query q_M and Web page d_N pairs, $X = \{x_{11}, x_{12}, \dots, x_{NM}\}$
- a set Y of M relevance decisions per each query q_M and Web page d_N pairs, $Y = \{y_{11}, y_{12}, \dots, y_{NM}\}$

The set of query-Web page pairs $\{q_M, d_{NM}\}$ has associated a set of feature vectors $\{x_{NM}\}$ that represents specific ranking parameters describing the match between them. The set of relevance judgments $\{y_{NM}\}$ can be a binary decision, order or score. Each feature x_{NM} and the corresponding score y_{NM} form an instance. The input to the learning algorithm is the set feature vectors $\{x_{NM}\}$ that represents the pair $\{q_M, d_{NM}\}$ and the desired output corresponds to set of relevance judgement $\{y_{NM}\}$. The ranking function $f(x)$ is acquired by the learning method during the training process on which relevance assessments are optimized with the performance measurement or cost function (Fig. 7).

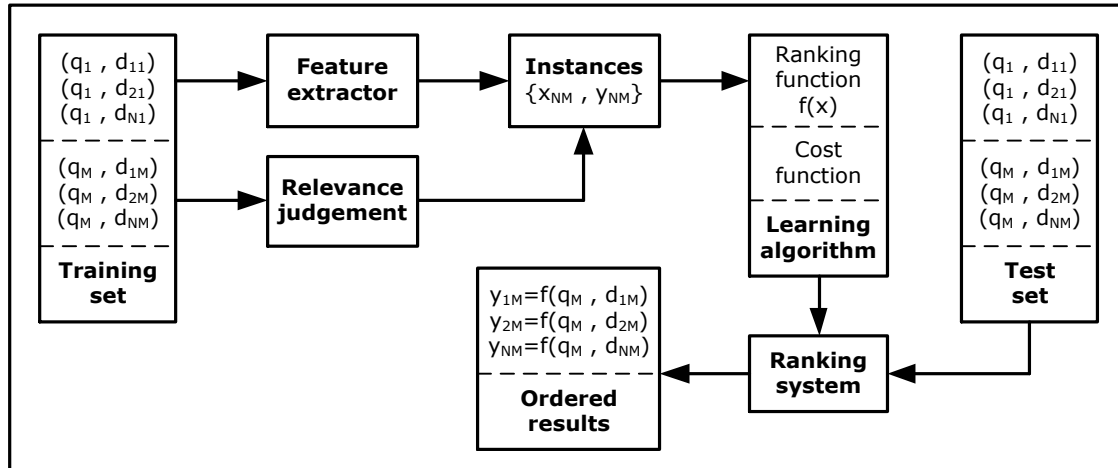


Figure 7: Learning to Rank Model

RankSVM learning to rank algorithm learns retrieval functions using clickthrough data for training based on a Support Vector Machine (SVM) approach as presented by Thorsten Joachims [83]. It uses a mapping function to correlate a search query with the features of each of the possible results where each data pair is projected into a feature space combined with the corresponding click-through data that measures relevance. Distinct ranking methods are developed for several classes of contexts by Biao Xiang et al [84] based on RankSVM that integrates the ranking principles into a new ranking model that encodes the context details as features. It learns a Support Vector Machine model for a binary categorization on the selection between a couple of Web pages where the evaluation is based on human conclusions and indirect user click.

A machine learning ranking model defined by Anlei Dong et al [85] routinely perceives and reacts to recent related queries where recent ranking takes into account freshness and relevance; the system is divided into two approaches: a high accuracy recent related query detector and a specific recent related sorter trained to represent the characteristics and data distribution applicable for recent ordering.

Random Forests is a low computational cost alternative to point-wise ranking approach algorithm presented by Ananth Mohan [86], it is founded on the machine learning method "Gradient Boosted Regression Trees". The learning algorithm of Random Forests is applied multiple times to different subsets and the results are averaged whereas Gradient Boosted Regression Trees sequentially adds small trees at each iteration instead of training many full trees. The combination of both two algorithms first learns a ranking function with Random Forests and later uses it as initialization for Gradient Boosted Regression Trees.

4 Recommender Systems

Recommender systems predict users' interest of different items or products providing a set of suggested items through the use of tailored relevance algorithms. Recommender systems consist on a database where user ratings and item description are stored and updated iteratively. A user interface interacts with the user where a profiler extracts user properties with explicit and implicit feedback; different suggestions and their order are computed by the recommender ranking algorithm (Fig. 8). Due to their filtering properties, they are widely used within e-commerce as presented by Lee Wei-Po et al [87] as they also support e-commerce customers to find rare products they might not have discovered by themselves.

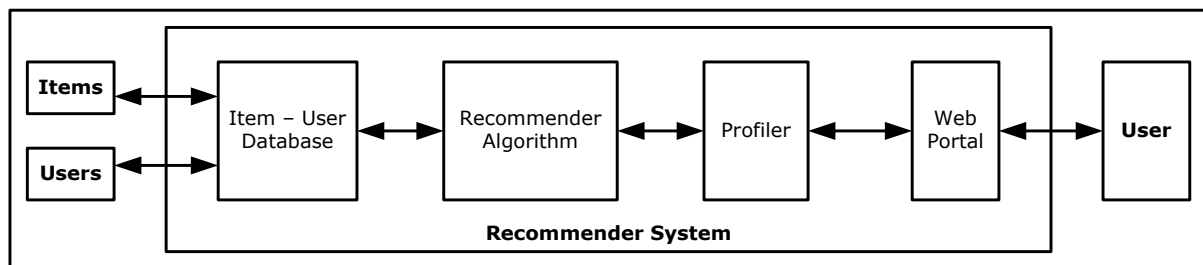


Figure 8: Recommender system architecture

4.1 Recommender System Types

There are two main different categories of recommender systems (Fig. 9) as described by Dhoha Almazro et al [88]. Content-based recommender systems are built on a description of the product and a profile of the customer's wishes where different properties of the items and users are used to identify products with similar features without including other user's ratings. They suffer from some disadvantages as presented by Gediminas Adomavicius et al [89] such as their inability to recommend completely different items that the user may also consider relevant and a customer needs to mark several products before getting relevant recommendations. Collaborative recommender systems are founded on the customer's previous marks to other products and the consideration of other decisions made by similar users; they made suggestions based on a high correlation between users or items. Although collaborative recommendation reduces the issues from the Content-based solution; it has other drawbacks such as it needs a large number of rating data to calculate accurate correlations and predictions; it also ignores on its calculations new added users or items. Hybrid Recommender Systems take a combination of both approaches.

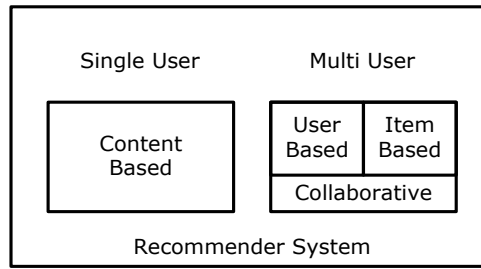


Figure 9: Common types of Recommender systems

In a multi-criteria ranking recommenders, users give ratings on several characteristics of an item as a vector of ratings as defined by Gediminas Adomavicius et al [90] whereas cross domain recommender systems suggest items from multiple sources with item or user based formulas founded on locality collaborative filtering as presented by Paolo Cremonesi et al [91]; they operate by first modelling the traditional likeness association as a directly connected graph and then exploring the entire potential paths that link users or items to discover new cross domain associations.

There are several relevance metrics for assessment of recommender systems in various e-commerce business frameworks presented by Gunnar Schröder et al [92]; accuracy evaluation metrics are allocated into three major categories: predictive based on the error between estimated and true user ratings, classification based on the successful decision making and rank based on the correct order.

Social network information is inserted to recommender systems by Xiwang Yang et al [93] as an additional input to improve accuracy; Collaborative Filtering based social recommender systems are classified into two categories: matrix factorization approach where user to user social information is integrated with user item feedback history and neighbourhood based social approaches based on social network graphs.

4.2 Recommender System Relevance Metrics

The user based recommender generally uses the Pearson's relationship similarity:

$$\text{similarity}(u,v) = \frac{\sum_{i \in M} (r_{u,i} - \bar{r}_u) * (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in M} (r_{u,i} - \bar{r}_u)^2} * \sqrt{\sum_{i \in M} (r_{v,i} - \bar{r}_v)^2}} \quad (1)$$

which the prediction on the user-based is defined below:

$$\text{prediction}(u,i) = \bar{r}_u + \frac{\sum_{v \in N(u)} \text{similarity}(u,v) * (r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u)} |\text{similarity}(u,v)|} \quad (2)$$

where $N(u)$ defines a set of neighbours of u , $\text{similarity}(u,v)$ represents the correlation between user u and v , $r_{u,i}$ is the mark of customer u to product i , $\bar{r}_u$ is the average rating of customer u and M is the total quantity of items.

The item based recommender systems generally use the Cosine Similarity:

$$\text{similarity}(u,v) = \frac{\sum_{d=1}^N i_d * p_d}{\sqrt{\sum_{d=1}^N i_d^2} * \sqrt{\sum_{d=1}^N p_d^2}} \quad (3)$$

although the Euclidean distance similarity can also be used:

$$\text{similarity}(i,p) = \frac{1}{\sqrt{\sum_{d=1}^N (i_d - p_d)^2}} \quad (4)$$

and the Manhattan distance similarity:

$$\text{similarity}(i,p) = \frac{1}{\sum_{d=1}^N |i_d - p_d|} \quad (5)$$

which the prediction on the item-based is defined below:

$$\text{prediction}(u,i) = \frac{\sum_{q \in N(i)} \text{similarity}(i,q) * r_{u,q}}{\sum_{q \in N(i)} \text{similarity}(i,q)} \quad (6)$$

where $N(i)$ is a collection of neighbours of i and $\text{similarity}(i,p)$ is the correlation between item i and p , i_d the relevance of item i to user d and N the overall quantity of users.

The linear prediction on the content-based is defined below:

$$\text{prediction}(u,i) = \langle w, x \rangle = \sum_{q=1}^{N+M} w_q x_q \quad (7)$$

where x are the instance vectors and w the matrix weights.

4.3 Recommender System Model

The model of a Recommender system (Fig. 10) consists on:

- A set Q of N users, $Q = \{u_1, u_2, \dots, u_N\}$
- A set I of M items, $I = \{i_1, i_2, \dots, i_M\}$
- A rating matrix R , $R = [r_{ij}]$ where $i \in Q$ and $j \in I$
- A set X of N feature vectors, $X = \{x_1, x_2, \dots, x_N\}$
- A weight matrix W , $W = [w_{ij}]$ where $i \in N$ and $j \in N+M$

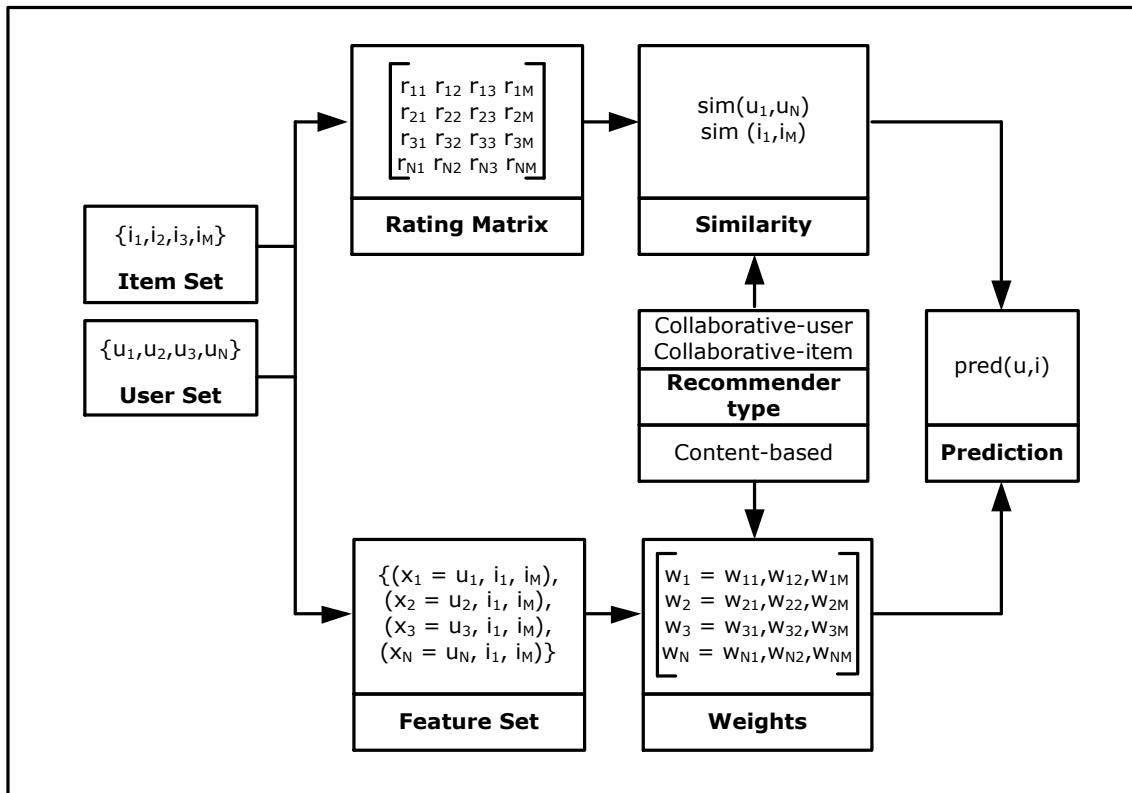


Figure 10: Recommender system model

The set of user-items $\{u_N, i_M\}$ has an associated set of feature vector $\{x_N\}$ that represents users and the different items assigned to them in the Content model. The relevance judgement $\text{pred}(u, i)$ can be a binary decision, order or score based on the similarity between users or items in the Collaborative model and weights in the Content model.

5 The Random Neural Network

5.1 Neural Networks

Artificial neural networks are models inspired by the brain central nervous system. They are usually presented as artificial nodes or "neurons" in different layers connected together via synapses to form a structure which mimics a biological neural network. The synapses have values called "weights" which value is updated during the learning algorithm calculations. There are two main models to represent a neural network (Fig. 11); the feed forward model where connections between the neurons follow only the forward direction and the recurrent model where connections between neurons form a direct cycle.

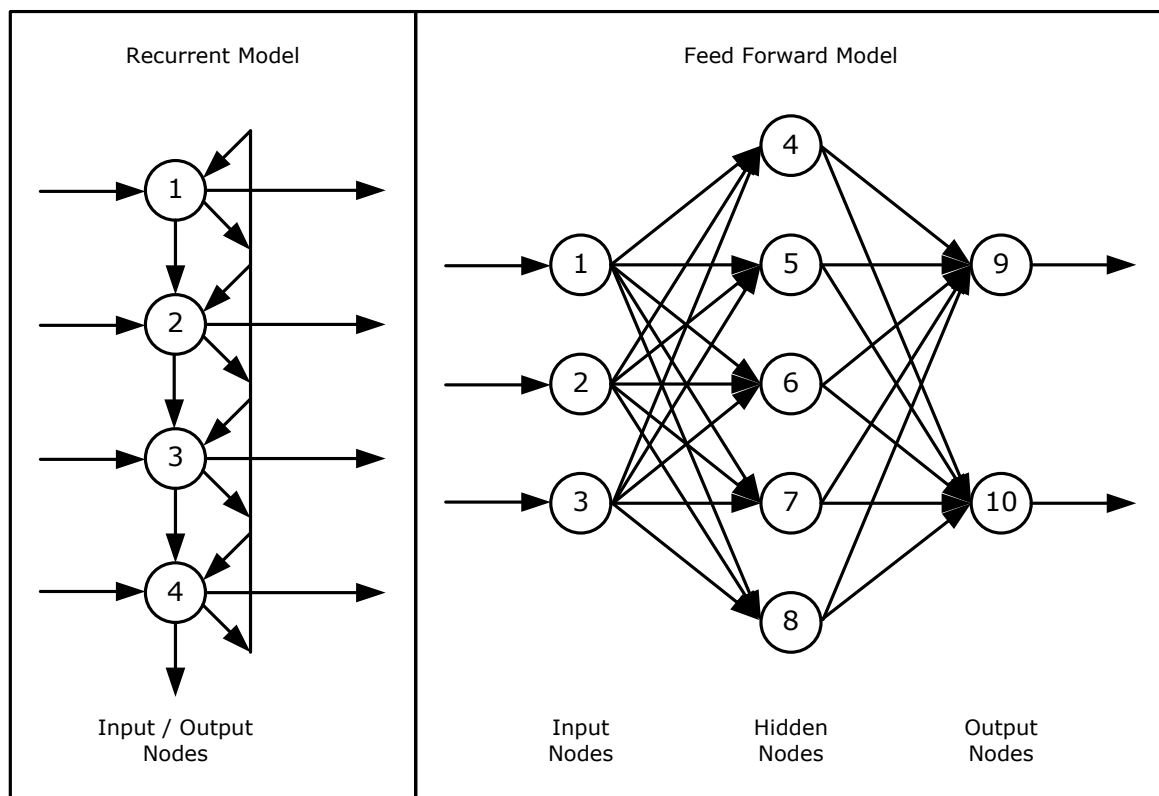


Figure 11: Artificial Neural Network – Recurrent and feed forward models

Artificial neural networks are normally defined by three parameters: the interconnection layout between different layers of neurons; the learning process for updating the weights of the interconnections and the activation function that converts a neuron's weighted input to its output activation.

Artificial neural networks have been used to solve a wide variety of tasks that would have been difficult to resolve by ordinary rules based programming, including computer vision, speech and pattern recognition. The learning capabilities of neural networks are the properties that have attracted the most interest; learning uses a set of observations to find which network weighs solve a specific task in some optimal sense. The cost function is an important concept in learning, as it measures how far away a particular solution is from an optimal solution.

There are three major learning methods: supervised learning, unsupervised learning and reinforcement learning.

Supervised learning finds a function, f , that from a given input - output pair (x, y) matches them. The cost function, C , is related to the mismatch between the artificial neural network approximation, $f(x)$ and the output, y . A commonly used cost is the mean-squared error, which tries to minimize the average squared error between the network's output, $f(x)$, and the target value y over the example pair.

An input x is given in unsupervised learning and the cost function, C , to be minimized depends on the task that is tried to be modelled and prior assumptions like implicit properties of the model, its parameters, or the observed variables.

The input x is usually not given in Reinforcement Learning; it is generated by an agent's interactions with the environment. At each point in time t , the agent performs an action and the environment generates an observation and an instantaneous cost C , according to some dynamics. The aim is to discover a rule for selecting actions that minimizes some long-term cost.

5.2 Deep Learning

Deep Learning models high level abstractions in data by using a network with multiple processing layers; it is composed of multiple linear and non-linear transformations. Deep learning is a branch of machine learning based on modelling representations of data.

Deep learning is characterized as using a cascade of l -layers of non-linear processing units for feature extraction and transformation (Fig. 12); each successive layer uses the output from the previous layer as input. Deep learning learns multiple layers of

representations that correspond to different levels of abstractions; those levels form a hierarchy of concepts where the higher the level, the more abstract concepts are learned.

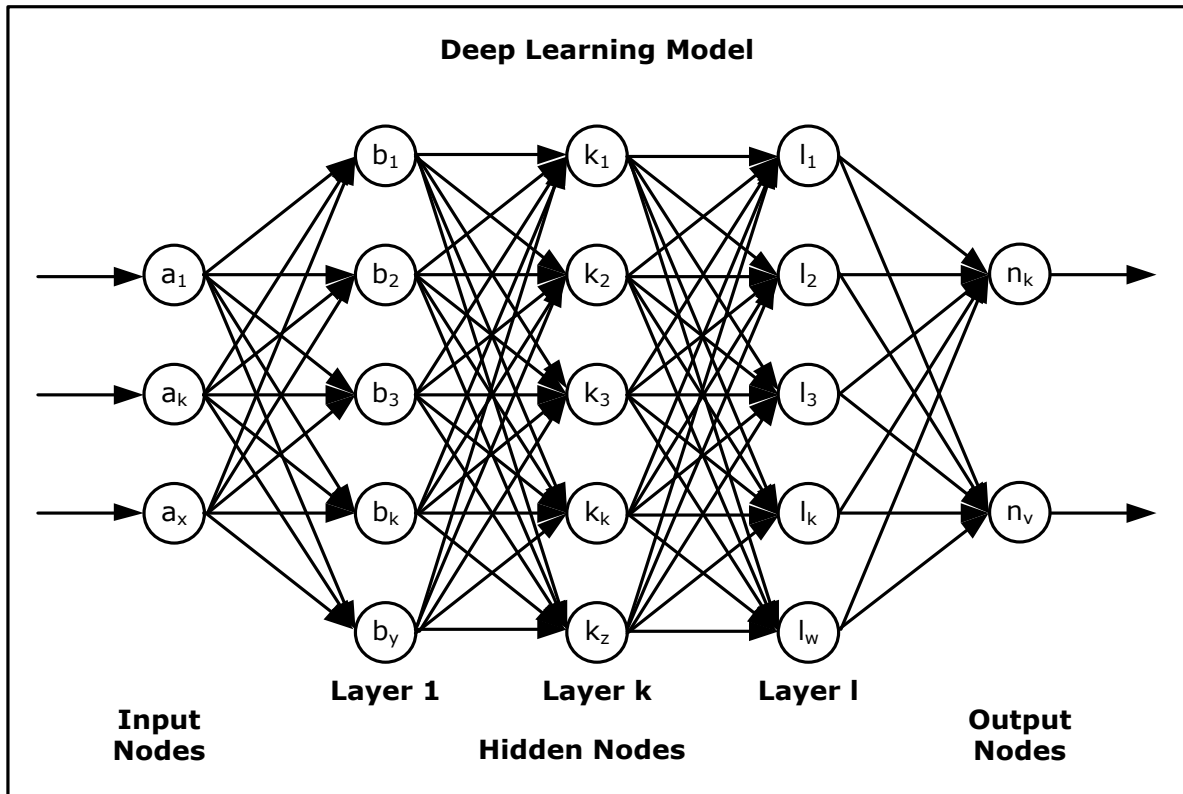


Figure 12: Artificial Neural Network – Deep Learning model

There are various deep learning architectures such as deep neural networks, convolutional deep neural networks, deep belief networks and recurrent neural networks. This research is based on a deep neural network with multiple hidden layers of processing units between the input and the output layers. Deep learning suffers two major issues; it does not generalize well due the additional layers of abstraction and it has a major computational learning time.

5.3 G-Networks

G-Networks or Gelenbe Network was presented by Erol Gelenbe [121] and Erol Gelenbe et al [122]; it is a model for neural networks as well as queueing systems with specific control functions, such as traffic routing or destruction. A product form solution to Jackson's theorem that requires the solution of a system of non-linear equations for the traffic flows was presented by Erol Gelenbe [123], the solution exists for the stationary distribution of G-networks as demonstrated by Erol Gelenbe et al [124].

A G-Network is an open network of G-queues with several types of customers:

- positive customers arrive from other queues or arrive externally as Poisson arrivals, and conform to normal service and routing rules as in conventional network models [123];
- negative customers arrive from another queue, or arrive externally as Poisson arrivals, and remove customers in a non-empty queue [123], representing the need to remove traffic when the network is congested, including the removal of "batches" of customers [125];
- "triggers" arrive from other queues or from outside the network, they displace customers and move them to other queues [126];
- "resets" arrive from other queues or from outside the network and they set the empty queue to a random length whose distribution is identical to the stationary distribution at that queue [127,128]. A reset which arrives to a non-empty queue has no effect.

The G-queues that form the G-network must have the following properties:

- each G-queue has one server, that serves at rate $\mu(i)$;
- external arrivals of positive customers, triggers or resets form Poisson processes of rate $\Lambda(i)$, for positive customers, while triggers and resets, including negative customers, form a Poisson process of rate $\lambda(i)$;
- a positive customer acts as usual when it arrives to a queue and increases the queue length by 1;
- a negative customer reduces the length of the queue by some random number (if there is at least one positive customer present at the queue) when it arrives to a queue. A trigger moves a customer probabilistically to another queue and a reset sets the state of the queue to its steady-state if the queue is empty when the reset arrives. Triggers, negative customers and resets disappear after they have taken their action;
- a customer moves from queue i to queue j when a service is completed as a positive customer with probability $p^+(i,j)$ as a trigger or reset with probability $p^-(i,j)$ and departs the network with probability $d(i)$.

G- Networks were extended to multiple classes of positive and negative customers by Jean-Michel Fourneau et al [129] and Erol Gelenbe et al [130]. A positive customer class is characterized by the routing probabilities and the service rate parameter at each

service center while negative customers of different classes may have different "customer destruction" capabilities.

G-networks have been used in a wide range of applications by Erol Gelenbe [131,132], such as single server presented by Erol Gelenbe et al [133] and resource allocation in multimedia systems as proposed by Erol Gelenbe et al [134].

5.4 The Random Neural Network

The Random Neural Network was presented by Erol Gelenbe [135]; it is a spiked recurrent stochastic model. The main analytical properties are the "product form" and the existence of a unique network steady state solution. The Random Neural Network model represents more closely how signals are transmitted in many biological neural networks where they travel as spikes rather than as fixed analogue signals.

5.4.1 Principles

The Random Neural Network consists on n -neurons (Fig. 13). The state of the n neuron network at time t is represented by the vector of non-negative integers $k(t) = [k_1(t), \dots, k_n(t)]$ where $k_i(t)$ is the potential of neuron i at time t . Neurons interact with each other by interchanging signals in the form of spikes of unit amplitude:

- A positive spike is interpreted as excitation signal because it increases by one unit the potential of the receiving neuron;
- A negative spike is interpreted as inhibition signal decreasing by one unit the potential of the receiving neuron or has no effect if the potential is already zero.

Each neuron accumulates signals and it will fire if its potential is positive. Firing will occur at random and spikes will be sent out at rate $r(i)$ with independent, identically and exponentially distributed inter-spike intervals:

- Positive spikes will go out to neuron j with probability $p^+(i,j)$ as excitatory signals;
- Negative spikes with probability $p^-(i,j)$ as inhibitory signals.

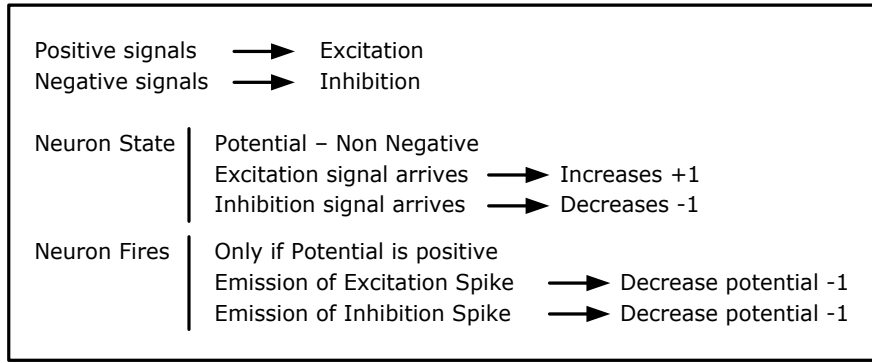


Figure 13: Random Neural Network: Principles

5.4.2 Model

A neuron may send spikes out of the network with probability $d(i)$. We have:

$$d(i) + \sum_{j=1}^n [p^+(i,j) + p^-(i,j)] = 1 \quad \text{for } 1 \leq i \leq n \quad (1)$$

Neuron potential decreases by one unit when the neuron fires either an excitatory spike or an inhibitory spike (Fig. 14). External (or exogenous) excitatory or inhibitory signals to neuron i will arrive at rates $\Lambda(i)$, $\lambda(i)$ respectively by stationary Poisson processes. The Random Neural Network weight parameters $w^+(j,i)$ and $w^-(j,i)$ are the non-negative rate of excitatory and inhibitory spike emission respectively from neuron i to neuron j :

$$\begin{aligned} w^+(j,i) &= r(i)p^+(i,j) \geq 0; \\ w^-(j,i) &= r(i)p^-(i,j) \geq 0. \end{aligned} \quad (2)$$

Information in this model is transmitted by the rate or frequency at which spikes travel. Each neuron i , if it is excited, behaves as a frequency modulator emitting spikes at rate $w(i,j) = w^+(i,j) + w^-(i,j)$ to neuron j . Spikes will be emitted at exponentially distributed random intervals. Each neuron acts as a non-linear frequency demodulator transforming the incoming excitatory and inhibitory spikes into potential.

This network model has a product form solution; the network's stationary probability distribution can be represented as the product of the marginal probabilities of the state of each neuron as demonstrated by Erol Gelenbe [136].

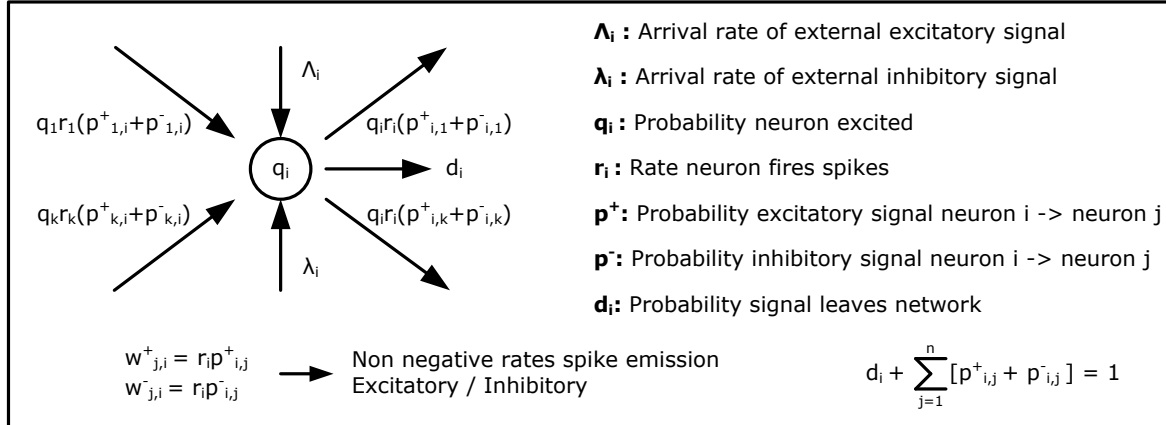


Figure 14: Random Neural Network: Model

5.4.3 Theorem

Let's define the probability distribution of the network state as $p(k, t) = \text{Prob}[k(t) = k]$ and the marginal probability a neuron i is excited at time t as $q_i(t) = \text{Prob}[k_i(t) > 0]$ (Fig. 15). The stationary probability distribution $p(k) = \lim_{t \rightarrow \infty} p(k, t)$ and $q_i = \lim_{t \rightarrow \infty} q_i(t)$ where $k(t)$ is a continuous time Markov chain that satisfies Chapman-Kolmogorov equations.

Let's define:

$$q_i = \frac{\lambda^+(i)}{r(i) + \lambda^-(i)} \quad r(i) = \sum_{j=1}^n [w^+(i, j) + w^-(i, j)] \quad \text{for } 1 \leq i \leq n \quad (3)$$

where the $\lambda^+(i)$, $\lambda^-(i)$ for $i=1, \dots, n$ satisfy the system of nonlinear simultaneous equations:

$$\begin{aligned} \lambda^+(i) &= \sum_{j=1}^n [q_j r(j) p^+(j, i)] + \Lambda(i) \\ \lambda^-(i) &= \sum_{j=1}^n [q_j r(j) p^-(j, i)] + \Lambda(i) \end{aligned} \quad (4)$$

If a nonnegative solution $\{\lambda^+(i), \lambda^-(i)\}$ exists to the equations 1 and 2 that meets $q_i < 1$ then:

$$p(k) = \prod_{i=1}^n [1 - q_i] q_i^{k_i} \quad (5)$$

The network will be stable if a value $q_i < 1$ can be found. The average potential at a neuron i is $q_i/[1-q_i]$ and the rate of emission of spikes from neuron i in steady state is $q_i r(i)$. If we have $\lambda^+(i) > [r(i) + \lambda^-(i)]$ for any neuron means that the neuron is unstable or saturated; this implies that it is constantly excited in steady state and its rate of excitatory and inhibitory spike emission $r(i)$ to another neuron j will be $r(i)p^+(i,j)$ and $r(i)p^-(i,j)$ respectively.

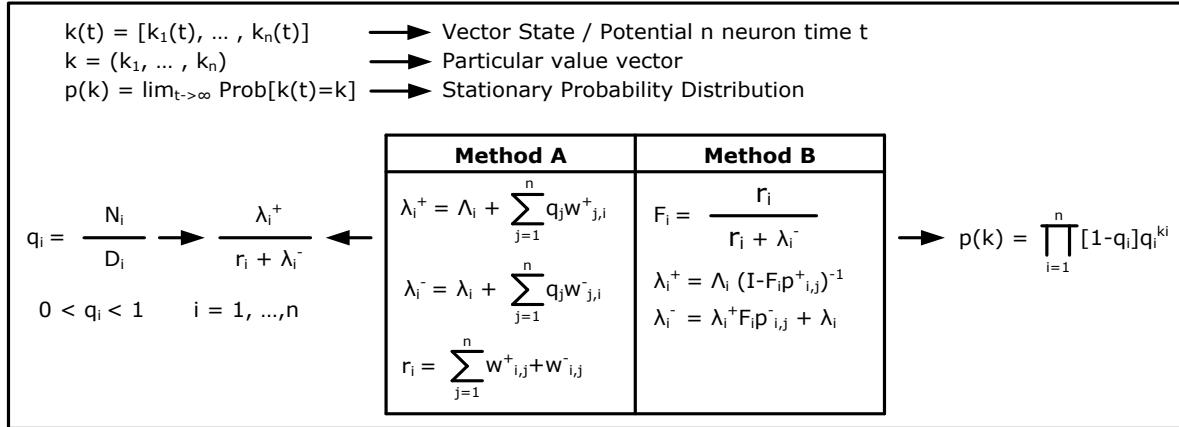


Figure 15: Random Neural Network: Theorem

5.4.4 Learning Algorithm: Gradient Descent

Neural networks are capable to learn iteratively through examples or training sets. The two main methods are supervised learning based on an input with the desired output and Reinforcement Learning based on the environment reactions from user actions.

The Random Neural Network learning algorithm proposed by Erol Gelenbe [137] is based on gradient descent of a quadratic error function (Fig. 16). The backpropagation model requires the solution of n linear and n nonlinear equations each time the n neuron network learns a new input and output pair.

Gradient Descent learning algorithm optimizes the network weight parameters w in order to learn a set of k input-output pairs (i,y) where successive inputs are denoted $i = \{i_1, \dots, i_k\}$ and the successive desired outputs are represented $y = \{y_1, \dots, y_k\}$.

Each input vector $i_k = (\Lambda_k, \lambda_k)$ is the pair of the excitatory and inhibitory signals entering each neuron: $\Lambda_k = [\Lambda_k(1), \dots, \Lambda_k(n)]$, $\lambda_k = [\lambda_k(1), \dots, \lambda_k(n)]$. Each output vector $y_k = (y_{1k}, \dots, y_{nk})$, $y_{ik} \in [0,1]$ is composed the desired values of each neuron.

The desired output vectors are approximated by minimizing the cost function E_k :

$$E_k = \frac{1}{2} \sum_{i=1}^n a_i (q_i - y_{ik})^2 \quad a_i \geq 0 \quad (6)$$

Each of the n neurons of the network is considered as an output neuron. The function of the variable a_i is to remove neurons from the network output. The network learns both $n \times n$ weight matrices $W_k^+ = \{w_k^+(i, j)\}$ and $W_k^- = \{w_k^-(i, j)\}$ by calculating new values of the network parameters for each input $i_k = (\Lambda_k, \lambda_k)$ using Gradient Descent.

The rule for weight update can take the generic form:

$$w_k(u, v) = w_{k-1}(u, v) - \eta \sum_{i=1}^n a_i (q_{ik} - y_{ik}) \left[\frac{\partial q_i}{\partial w(u, v)} \right]_k \quad (7)$$

where η is the learning rate and the term $w(u, v)$ denotes either $w^+(u, v)$ or $w^-(u, v)$

To evaluate the partial derivatives we define the vector $q = (q_1, \dots, q_n)$ and the $n \times n$ matrix:

$$W = \frac{w^+(i, j) - w^-(i, j) q_j}{r(i) + \lambda^-(i)} \quad \text{for } i, j = 1, \dots, n \quad (8)$$

We also define the n -vectors:

$$\gamma^+(u, v) = [\gamma_1^+(u, v), \dots, \gamma_n^+(u, v)] \quad (9)$$

and

$$\gamma^-(u, v) = [\gamma_1^-(u, v), \dots, \gamma_n^-(u, v)] \quad (10)$$

where:

$$\gamma_i^+(u, v) = \begin{cases} \frac{-1}{r(i) + \lambda^-(i)} & \text{if } u=i, v \neq i \\ \frac{+1}{r(i) + \lambda^-(i)} & \text{if } u \neq i, v=i \\ 0 & \text{for all other values of } (u, v) \end{cases} \quad \gamma_i^-(u, v) = \begin{cases} \frac{-(1+q_i)}{r(i) + \lambda^-(i)} & \text{if } u=i, v=i \\ \frac{-1}{r(i) + \lambda^-(i)} & \text{if } u=i, v \neq i \\ \frac{-q_i}{r(i) + \lambda^-(i)} & \text{if } u \neq i, v=i \\ 0 & \text{for all other values of } (u, v) \end{cases}$$

The vector equations can be written:

$$\frac{\partial q}{\partial w^+(u,v)} = \frac{\partial q}{\partial w^+(u,v)} W + \gamma^+(u,v) q_u$$

$$\frac{\partial q}{\partial w^-(u,v)} = \frac{\partial q}{\partial w^-(u,v)} W + \gamma^-(u,v) q_u$$

or equivalently:

$$\frac{\partial q}{\partial w^+(u,v)} = \gamma^+(u,v) q_u [I - W]^{-1}$$

$$\frac{\partial q}{\partial w^-(u,v)} = \gamma^-(u,v) q_u [I - W]^{-1}$$

(11)

where I denotes the n x n identity matrix.

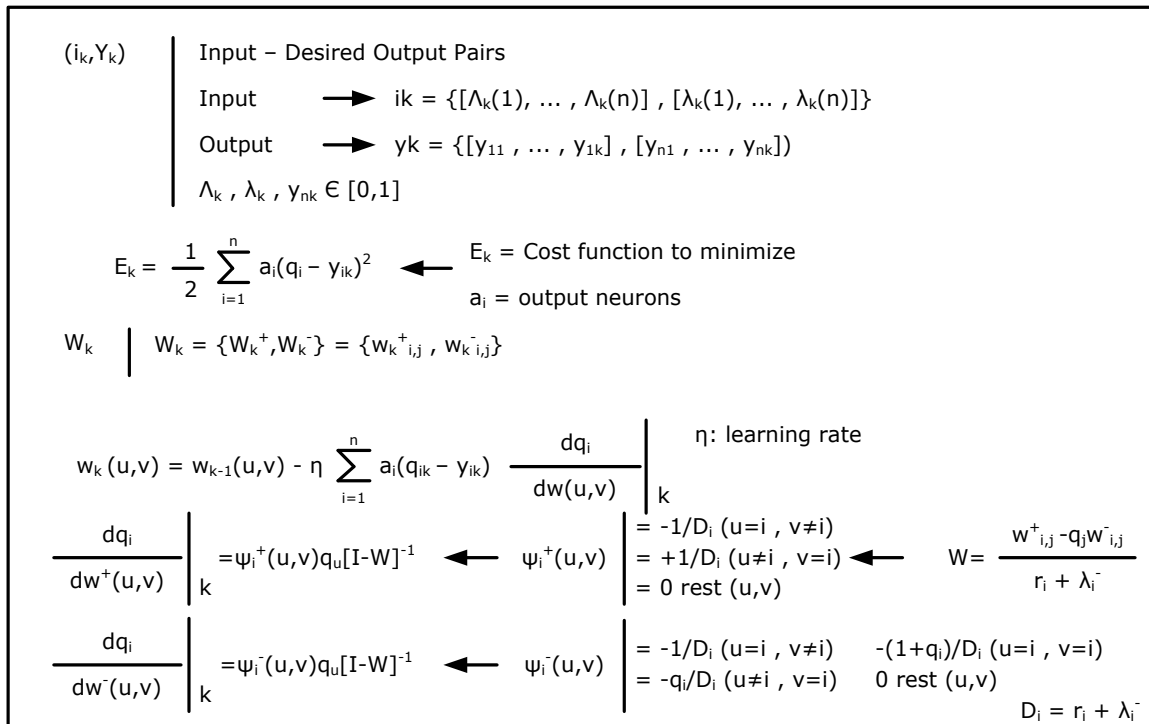


Figure 16: Random Neural Network: Gradient Descent learning

The complete learning algorithm can be specified in the following steps. We need first to appropriately initiate the weight matrices W_0^+ and W_0^- (Fig. 17). The network weight initialization method is based on the random initialization. A value for η needs to be selected and for each successive value of k the following steps apply:

- 1- Set the input values to $i_k = (\Lambda_k, \lambda_k)$.
- 2- Solve the system of non-linear equations 3 and 4.
- 3- Solve the linear equations 11 with the results of 4.

- 4- Update the weight matrices W_k^+ and W_k^- following the equation 4 using the results of 4 and 6.
- 5- Evaluate the cost function E_k according to equation 6 using the results of 7.

We iterate this learning algorithm until the value of the cost function from the network weight matrices is smaller than some predetermined value.

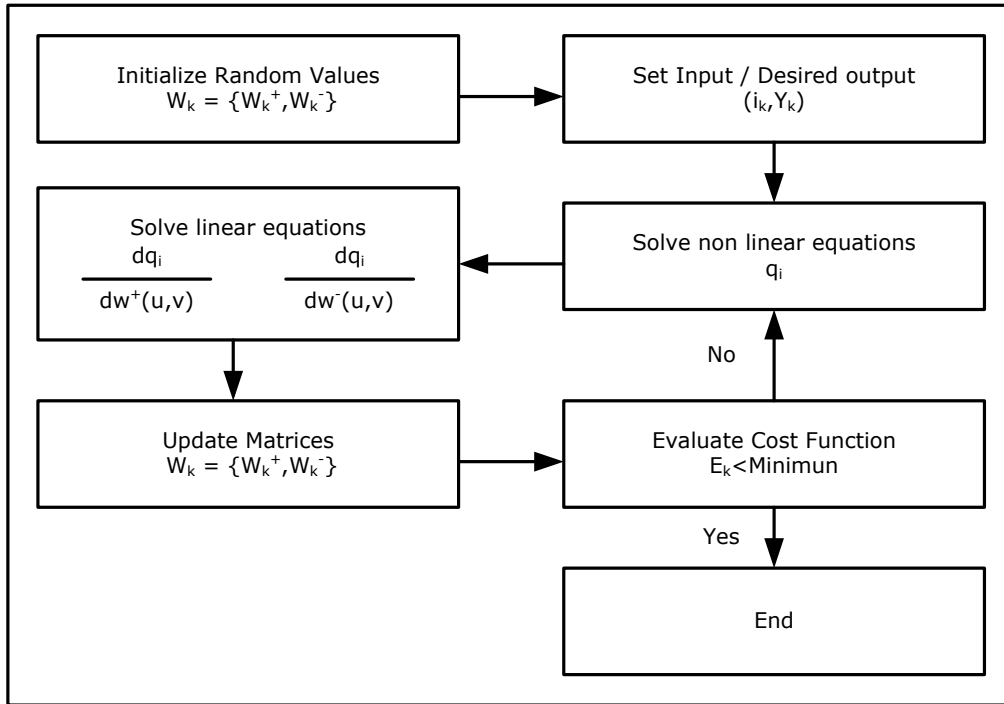


Figure 17: Random Neural Network: Gradient Descent iteration

5.4.5 Learning Algorithm: Reinforcement Learning

The Reinforcement Learning algorithm optimizes the network weight parameters w based on the external interaction with the environment after a selected search a_d from a cascaded set of decisions at each time step $d=0,1,2 \dots m$, where m is the total number of searches (Fig. 18).

Let's define a reward R that the algorithm needs to optimize based on the estimation that a result will be relevant to a user high level query. Successive values of the R are denoted by R_l , $l=1,2,..m$ of searches where m is the total number. The computation of the decision threshold will be:

$$T_l = \alpha T_{l-1} + (1-\alpha)R_l \quad (12)$$

Where α is a constant which weights the relevance of previous searches.

The amount of reinforcement is calculated according to the action sequence a_d . The objective of Reinforcement Learning is to find a strategy for selecting the action sequence that maximizes the expected reward and or minimizes the expected punishment.

Let the n neurons be numbered $1, \dots, i, \dots, n$ where each result i is assigned to a neuron i . Decisions in the Reinforcement Learning Algorithm with the RNN are based by selecting the result j for which the corresponding neuron is the most excited (largest value of q_j).

Let's suppose after the action a_i we have chosen the result j which corresponds to neuron j and we have measured the reward R_i . If the reward R_i is larger than the weighted relevance of previous searches T_{i-1} we reward by increasing significantly the excitatory weights going to that neuron and make a small increase of the inhibitory weights leading to other neurons.

If the reward R_i is not better than the weighted relevance of previous searches we punish by increasing moderately all excitatory weights leading to all neurons, except for the previous selected result and increasing significantly the inhibitory weights leading to the previous selected result. This algorithm is shown below:

if $T_{i-1} \leq R_i$

$$\begin{aligned} w^+(i, j) &\leftarrow w^+(i, j) + R_i \\ w^-(i, k) &\leftarrow w^-(i, k) + \frac{R_i}{n-1} \text{ if } k \neq j \end{aligned} \quad (13)$$

else

$$\begin{aligned} w^+(i, k) &\leftarrow w^+(i, k) + \frac{R_i}{n-1} \text{ if } k \neq j \\ w^-(i, j) &\leftarrow w^-(i, j) + R_i \end{aligned} \quad (14)$$

The state of the neural network is determined by the relative size of the weights of the RNN, rather than the actual values. We re-normalize all the weights:

$$\begin{aligned} r_i^* &= \sum_{m=1}^n [w^+(i, m) + w^-(i, m)] \\ w^+(i, j) &\leftarrow w^+(i, j) \frac{r_i}{r_i^*} \end{aligned}$$

$$w^-(i, j) \leftarrow w^-(i, j) \frac{r_i}{r_i^*} \quad (15)$$

The probabilities q_i are calculated by solving iteratively the non-linear equations (3) and (4). The decision of which result will be the most relevant to the user high level query corresponds to the neuron which has the largest probability.

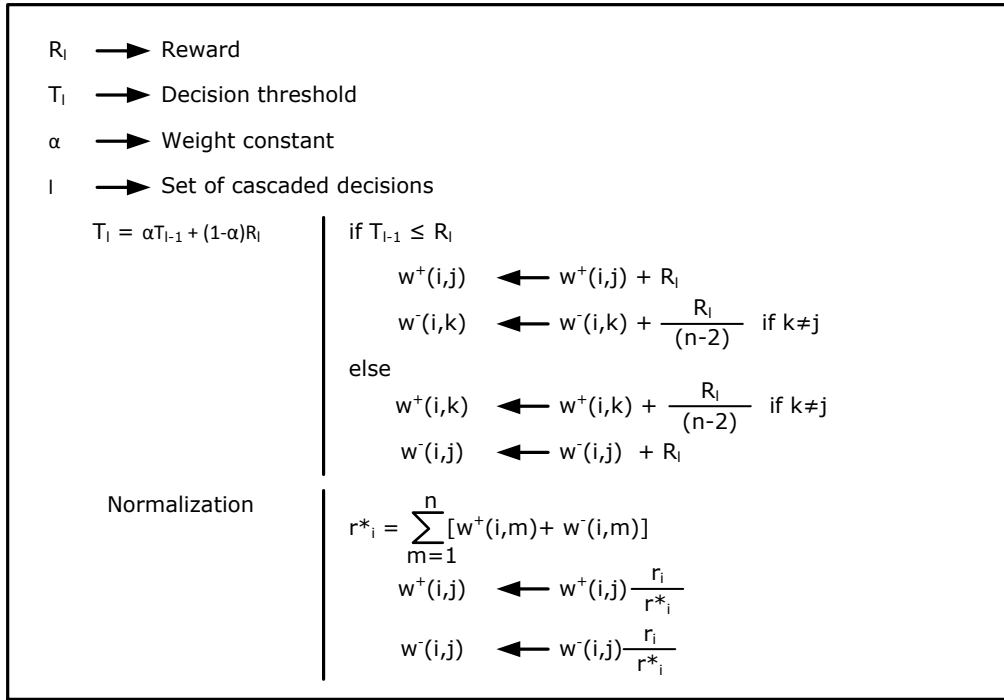


Figure 18: Random Neural Network: Reinforcement Learning

This iterative process is a m-stages cascaded decision learning where at each step the user needs to take an action or decision that it will change the environment changing the state of the network from state $x_i(m-1)$ to a new state $x_j(m)$. When the action sequence is completed a reinforcement $R_n(a_n)$ is produced as a result of the interaction with the user (Fig. 19).

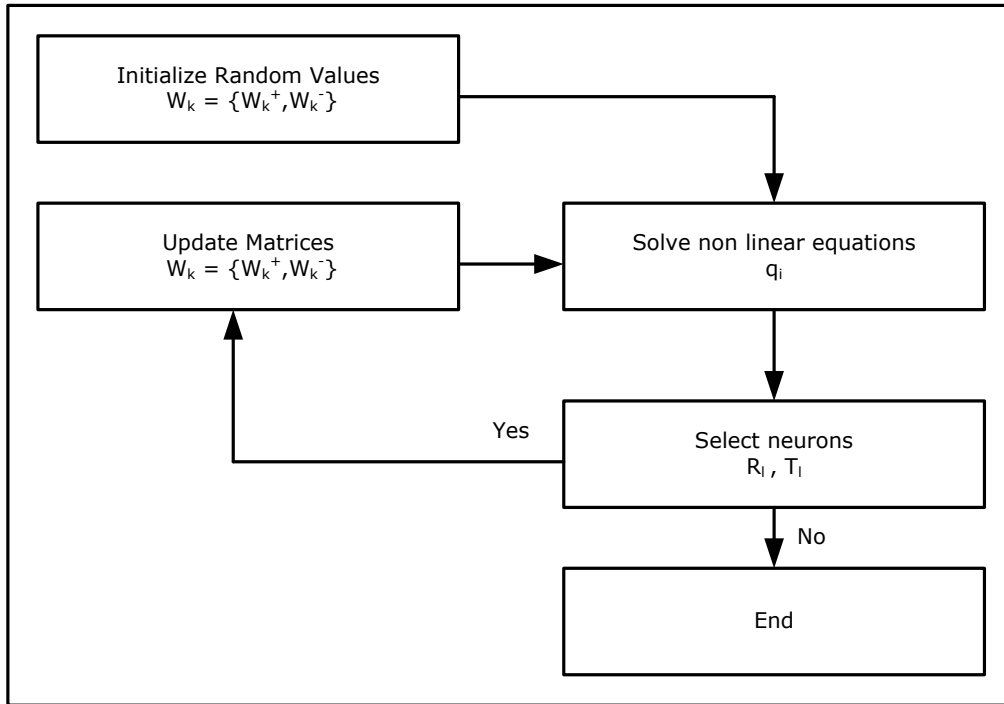


Figure 19: Random Neural Network: Reinforcement iteration

5.5 The Deep Learning Cluster Random Neural Network

5.5.1 Mathematical Model

The Random Neural Network is composed of M neurons each of which receives excitatory (positive) and inhibitory (negative) spike signals from external sources which may be sensory sources or other neurons as described by Erol Gelenbe et al [145]. These spike signals occur following independent Poisson processes of rates $\lambda^+(m)$ for the excitatory spike signal and $\lambda^-(m)$ for the inhibitory spike signal respectively, to cell $m \in \{1, \dots, M\}$ as defined by Erol Gelenbe et al [145].

In this model (Fig. 20), each neuron is represented at time $t \geq 0$ by its internal state $k_m(t)$ which is a non-negative integer. If $k_m(t) \geq 0$, then the arrival of a negative spike to neuron m at time t results in the reduction of the internal state by one unit: $k_m(t^+) = k_m(t) - 1$. The arrival of a negative spike to a cell has no effect if $k_m(t) = 0$. On the other hand, the arrival of an excitatory spike always increases the neuron's internal state by 1; $k_m(t^+) = k_m(t) + 1$.

If $k_m(t) > 0$, then the neuron m is defined as “excited”, and it may “fire” a spike with probability $r_m \Delta t$ in the interval $[t, t + \Delta t]$ where $r_m > 0$ is its “firing rate”, so that r_m is the average firing delay of the excited m neuron.

Neurons in this model can interact in the following manner at time $t \geq 0$. If neuron i is excited ($k_i(t) > 0$) then when neuron i fires its internal state drops by 1 ($k_m(t^+) = k_m(t) - 1$) and:

- It can send a positive or excitatory spike to neuron j with probability $p^+(i, j)$ resulting in $k_j(t^+) = k_j(t) + 1$;
- Or it can send a negative or inhibitory spike to neuron j with probability $p^-(i, j)$ resulting in $k_j(t^+) = k_j(t) - 1$ if $k_j(t) > 0$, else $k_j(t^+) = 0$ if $k_j(t) = 0$;
- Or it can trigger neuron j with probability $p(i, j)$ resulting in $k_j(t^+) = k_j(t) - 1$ if $k_j(t) > 0$, else $k_j(t^+) = 0$ if $k_j(t) = 0$ and one of two may happen. Either:
 - (A) with probability $Q(j, m)$ we have $k_m(t^+) = k_m(t) + 1$;
 - (B) or with probability $n(j, m)$ the trigger moves on to the neuron m and then with probability $Q(m, l)$ the sequence (A) or (B) is repeated.

Note that:

$$\sum_{j=1}^M [p(i, j) + p^+(i, j) + p^-(i, j)] = 1 - d(i) \quad \text{for } 1 \leq i \leq M \quad (16)$$

$$\sum_{m=1}^M [Q(j, m) + n(j, m)] = 1 \quad \text{for } 1 \leq j \leq M \quad (17)$$

where $d(i)$ is the probability that when the neuron i fires, the corresponding spike or trigger is lost or it leaves the network.

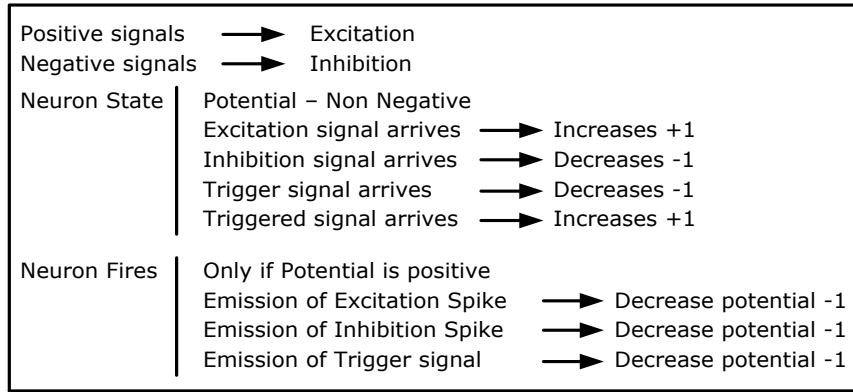


Figure 20: Cluster Random Neural Network: Principles

5.5.2 Neuron interaction

The model proposes $z(m) = (i_1, \dots, i_l)$ as any ordered sequence of distinct numbers $i_j \in S$; $i_j \neq m$; and $1 \leq l \leq M-1$ (Fig. 21). It defines $q_m = \lim_{t \rightarrow \infty} \text{Prob}[k_m(t) > 0]$ as the probability that the neuron m is excited. It is given by the following expression:

$$q_m = \frac{\lambda^+(m)}{r(m) + \lambda^-(m)} \quad (18)$$

$$\lambda^+(m) = \Lambda(m) + \sum_{j=1, j \neq m}^M [q_j r(j) p^+(j, m)] + \sum_{\text{all } z(m)} r_l(i) * \prod_{j=1, \dots, l-1} [q_j(i) p(i_j, i_{j+1}) Q(i_{j+1}, m)] \quad (19)$$

$$\lambda^-(m) = \lambda(m) + \sum_{j=1, j \neq m}^M [q_j r(j) p^-(j, m)] + \sum_{\text{all } z(m)} r_l(i) * \prod_{j=1, \dots, l-1} [q_j(i) p(i_j, i_{j+1}) p(i_{j+1}, m)] \quad (20)$$

To simplify the notations; we will define $w^+(j, i) = r(i) p^+(j, i)$ and $w^-(j, i) = r(i) p^-(j, i) \geq 0$ respectively.

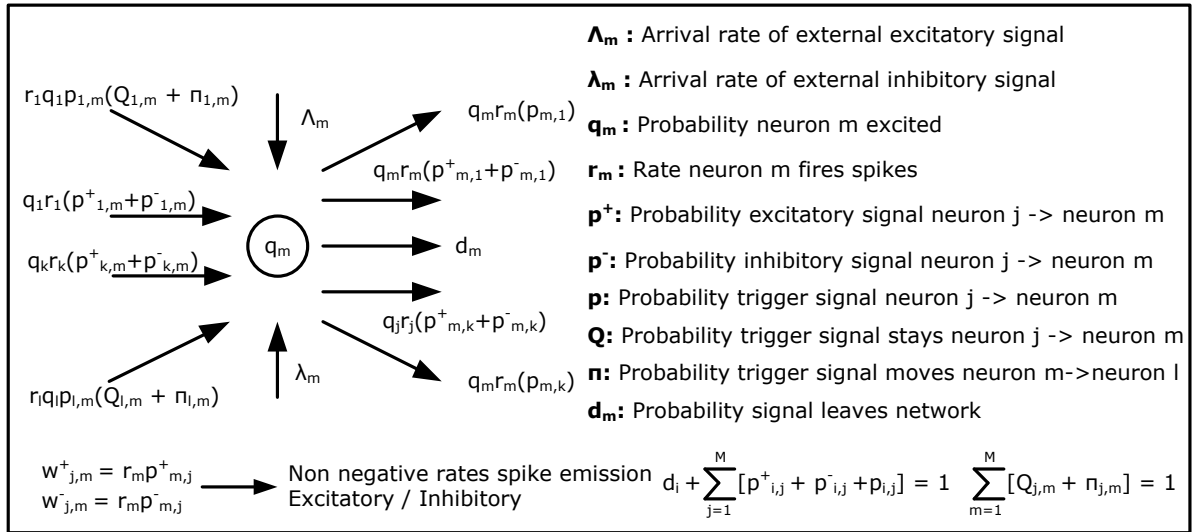


Figure 21: Cluster Random Neural Network: Model

5.5.3 Clusters of neurons

The model considers a special network $M(n)$ that contains n identically connected neurons, each which has a firing rate r and external inhibitory and excitatory signals Λ and λ respectively (Fig. 22). The state of each cell is denoted by q , and it receives an inhibitory input from the state of some cell u which does not belong to $M(n)$. Thus for any cell $i \in M(n)$ we have an inhibitory weight $w^-(u) \equiv w^-(u,i) > 0$ from u to i .

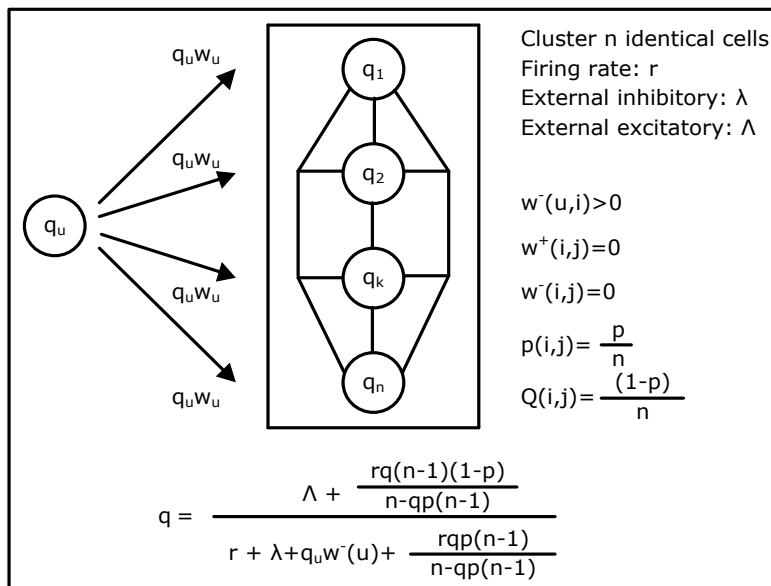


Figure 22: Single Cluster Random Neural Network: Theorem

For any $i, j \in M(n)$ we have $w^+(i, j) = w^-(i, j) = 0$, but all whenever one of the cells fires, it triggers the firing of the other cells with the following values:

$$p(i, j) = \frac{p}{n} \quad ; \quad Q(i, j) = \frac{1-p}{n}$$

As a result we have:

$$q = \frac{\Lambda + rq(n-1) \sum_{l=0}^{\infty} \left[\frac{qp(n-1)}{n} \right]^l \frac{1-p}{n}}{r + \lambda + q_u w^-(u) + rq(n-1) \sum_{l=0}^{\infty} \left[\frac{qp(n-1)}{n} \right]^l \frac{p}{n}} \quad (21)$$

which reduces to:

$$q = \frac{\Lambda + \frac{rq(n-1)(1-p)}{n-qp(n-1)}}{r + \lambda + q_u w^-(u) + \frac{rqp(n-1)}{n-qp(n-1)}} \quad (22)$$

Which is a second degree polynomial in q :

$$0 = q^2 p(n-1) [\lambda + q_u w^-(u)] - q [np(\Lambda + r) + n(\lambda + q_u w^-(u)) - p(\Lambda + r) + r] + n\Lambda \quad (23)$$

Hence it can be easily solved for its positive roots which are less than one, which are the only ones of interest since q is a probability.

5.5.4 The Random Neural Network with multiple clusters

The Deep Learning Architecture presented by Erol Gelenbe et al [145] is composed of C clusters $M(n)$ each with n hidden neurons (Fig. 23). For the c -th such cluster, $c = 1, \dots, C$, the state of each of its identical cells is denoted by q_c . In addition, there are U input cells which do not belong to these C clusters, and the state of the u -th cell $u=1, \dots, U$ is denoted by $\overline{q}_u$. The cluster network has U input cells and C clusters.

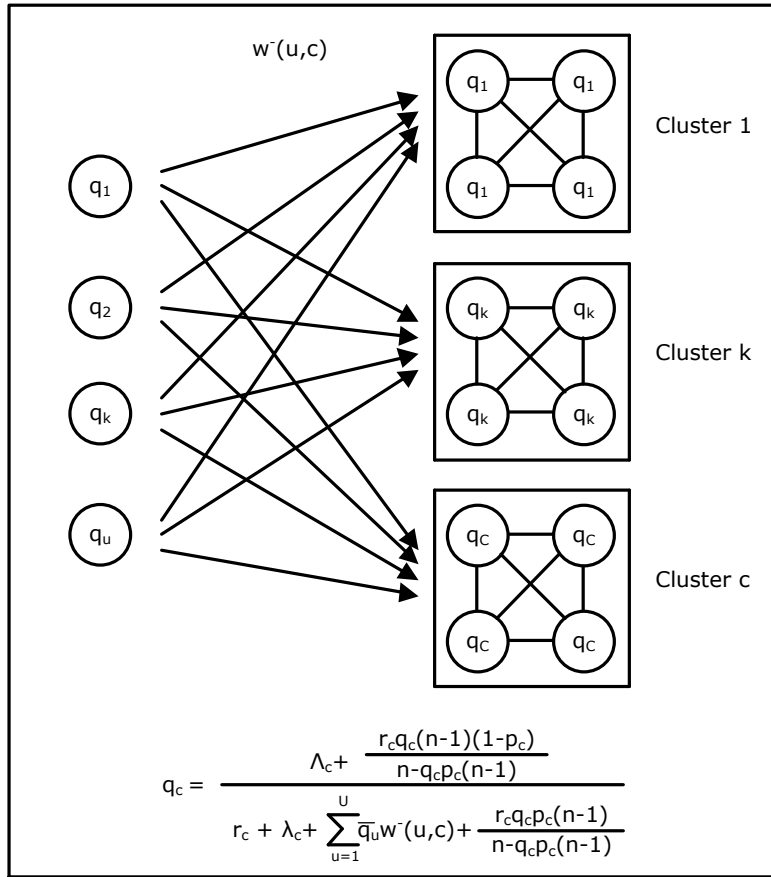


Figure 23: Multiple Cluster Random Neural Network: Theorem

Each hidden cell in the clusters c , with $c \in \{1, \dots, C\}$ receives an inhibitory input from each of the U input cell. Thus, for each cell in the c -th cluster, we have inhibitory weights $w^-(u, c) > 0$ from the u -th input cell to each cell in the c -th cluster; the u -th input cell will have a total inhibitory “exit” weight, or total inhibitory firing rate $\bar{r}_u$ to all the clusters which is of value:

$$\bar{r}_u = n \sum_{c=1}^C w^-(u, c)$$

(24)

then, from (22) and (24), we have:

$$q_c = \frac{\Lambda_c + \frac{r_c q_c (n-1)(1-p_c)}{n - q_c p_c (n-1)}}{r_c + \lambda_c + \sum_{u=1}^U \bar{q}_u w^-(u, c) + \frac{r_c q_c p_c (n-1)}{n - q_c p_c (n-1)}}$$

(25)

yielding a second degree polynomial for each of the q_c :

$$0 = q_c^2 p_c (n - 1) \left[\lambda_c + \sum_{u=1}^U \bar{q}_u w^-(u, c) \right] - q_c [np(\Lambda_c + r_c) + n(\lambda_c - \sum_{u=1}^U \bar{q}_u w^-(u, c)) - p(\Lambda_c + r_c) + r_c] + n\Lambda_c$$

(26)

its positive root is then:

$$q_c = \frac{b_c - \sqrt{b_c^2 - 4a_c d_c}}{2a_c}$$

where:

$$a_c = p_c (n - 1) \left[\lambda_c + \sum_{u=1}^U \bar{q}_u w^-(u, c) \right]$$

$$b_c = -[np(\Lambda_c + r_c) + n(\lambda_c - \sum_{u=1}^U \bar{q}_u w^-(u, c)) - p(\Lambda_c + r_c) + r_c]$$

$$d_c = n\Lambda_c$$

5.5.5 Deep learning clusters

The learning model of the Deep Learning Clusters was proposed by Erol Gelenbe et al [145]. It defines:

- I , a U -dimensional vector $I \in [0,1]^U$ that represents the input state $\bar{q}_u$ for the cell u ;
- $w^-(u,c)$ is the $U \times C$ matrix of weights from the U input cells to the cells in each of the C clusters;
- Y , a C -dimensional vector $Y \in [0,1]^C$ that represents the cell state q_c for the cluster c .

Let us now define the activation function of the c -th cluster as:

$$\zeta(x_c) = \frac{b_c}{2a_c} - \frac{\sqrt{b_c^2 - 4a_c d_c}}{2a_c} \quad (27)$$

where:

$$x_c = \sum_{u=1}^U \bar{q}_u w^-(u, c) \quad (28)$$

we have:

$$y_c = \zeta(x_c)$$

Gradient Descent learning algorithm optimizes the network weight parameters $w^-(u, c)$ from a set of input-output pairs (i_u, y_c) :

- the input vector $I = (i_1, i_2, \dots, i_u)$ where i_u is the input state $\bar{q}_u$ for cell u ;
- the output vector $Y = (y_1, y_2, \dots, y_c)$ where y_c is the cell state q_c for the cluster c .

The desired output vector is approximated by minimizing the cost function E_c :

$$E_c = \frac{1}{2} (q_c - y_c)^2 \quad (29)$$

The network learns the $U \times C$ weight matrix $w^-(u, c)$ by calculating new values of the network parameters for the input X and output Y using Gradient Descent (Fig. 24). The rule for weight update can take the generic form:

$$w_k^-(u, c) = w_{k-1}^-(u, c) - \eta (q_c - y_c) \frac{dq_c}{dx_c} \quad (30)$$

where η is the learning rate and k the iteration number

We calculate the derivative as:

$$\frac{dq_c}{dx_c} = \frac{n}{2a_c} + \frac{p(n-1)b_c}{a_c^2} + \frac{nb_c + 2p(n-1)d_c}{2a_c\sqrt{b_c^2 - 4a_cd_c}}$$

(31)

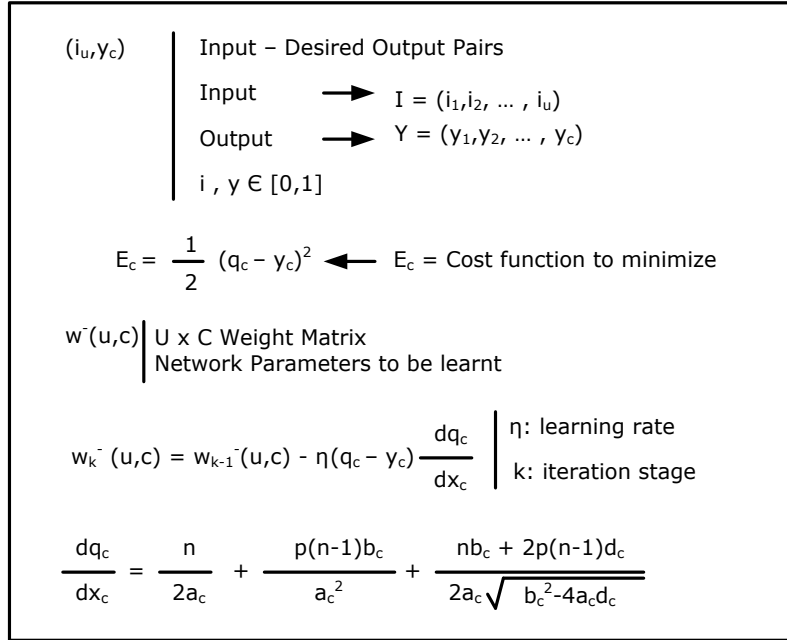


Figure 24: Cluster Random Neural Network: Gradient Descent learning

The complete learning algorithm can be specified in the following steps. We need first to appropriately initiate the weight matrix $w^-(u, c)$ with a random initialization (Fig. 25). A value for η needs to be selected:

- 1- Set the input values to $I = (i_1, i_2, \dots, i_u)$
- 2- Calculate q_c
- 3- Calculate derivative dq_c/dx_c
- 4- Update the weight matrices $w^-(u, c)$ following the equation 30 using the results of 28 and 31
- 5- Evaluate the cost function E_c according to equation 29 using the results of 30

We iterate this learning algorithm until the value of the cost function from the network weight matrices is smaller than some predetermined value.

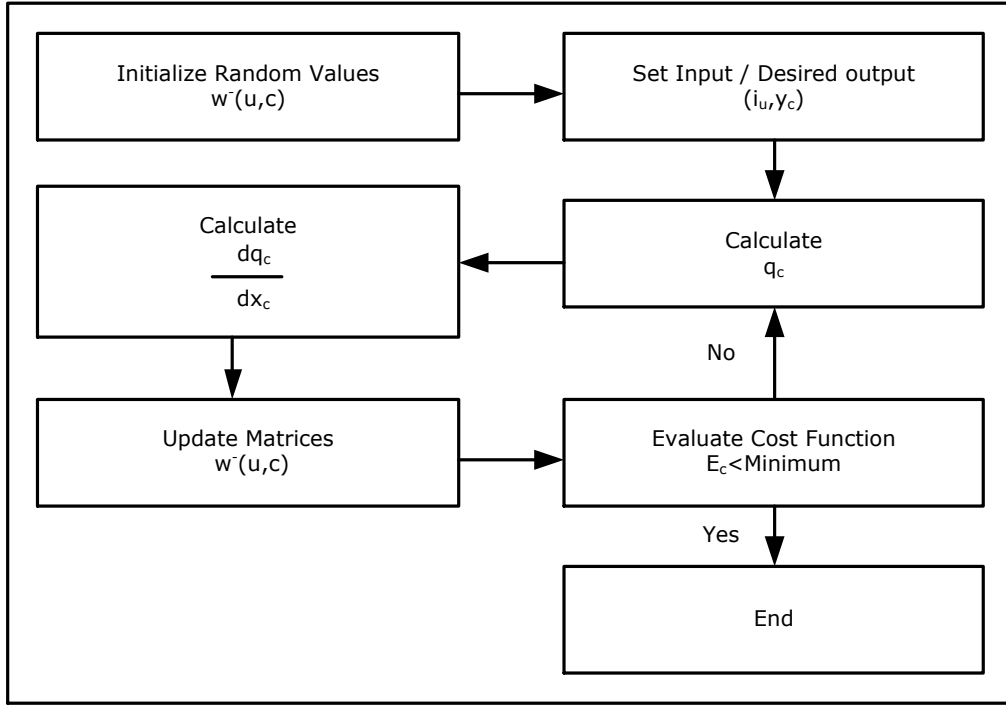


Figure 25: Cluster Random Neural Network: Gradient Descent iteration

5.5.6 Management cluster in the Random Neural Network

We propose the Management cluster model (Fig. 26) in this section:

- I_{mc} , a C-dimensional vector $I_{mc} \in [0,1]^C$ that represents the input state $\bar{q}_c$ for the cluster c ;
- $w(c)$ is the C-dimensional vector of weights from the C input clusters to the cells in the Management Cluster mc ;
- Y_{mc} , a scalar $Y_{mc} \in [0,1]$, the cell state q_{mc} for the Management Cluster mc .

Let us now define the activation function of the management cluster mc as:

$$\zeta(x_{mc}) = \frac{[np(\Lambda_{mc} + r_{mc}) + n(\lambda_{mc} + x_{mc}) - p(\Lambda_{mc} + r_{mc}) + r_{mc}]}{2p_{mc}(n-1)[\lambda_{mc} + x_{mc}]} - \frac{\sqrt{[np(\Lambda_{mc} + r_{mc}) + n(\lambda_{mc} + x_{mc}) - p(\Lambda_{mc} + r_{mc}) + r_{mc}]^2 - 4p_{mc}(n-1)[\lambda_{mc} + x_{mc}]n\Lambda_{mc}}}{2p_{mc}(n-1)[\lambda_{mc} + x_{mc}]} \quad (32)$$

where:

$$x_{mc} = \sum_{c=1}^C \bar{q}_c w(c) \quad (33)$$

we have:

$$y_{mc} = \zeta(x_{mc})$$

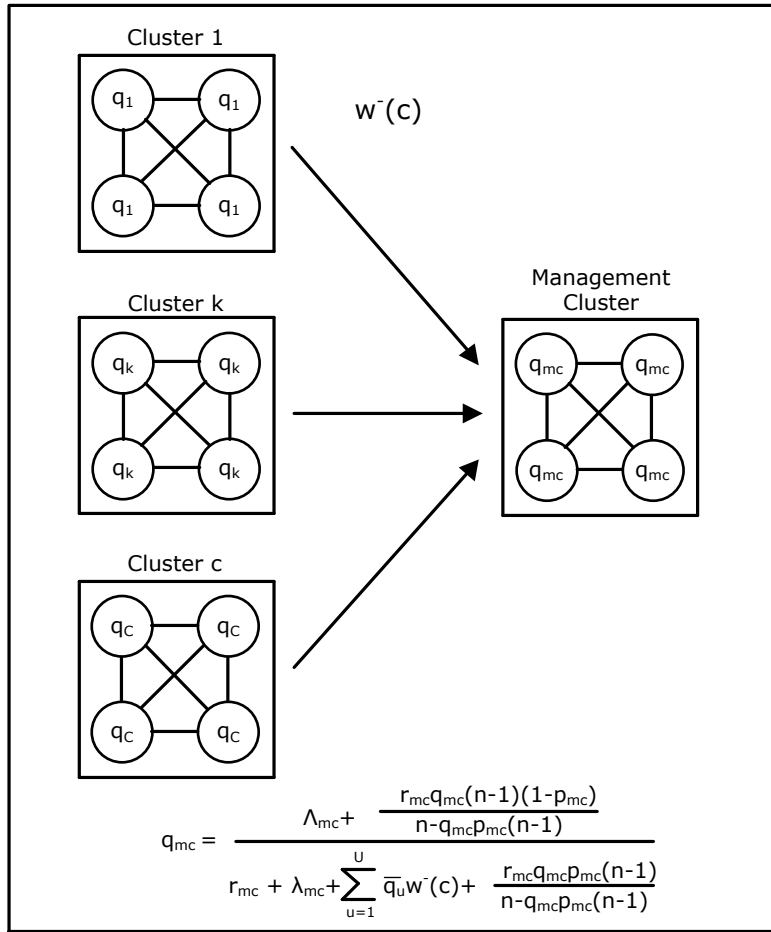


Figure 26: The Random Neural Network with a Management Cluster

The input state $\bar{q}_c$ for cell c represents the result relevance from each learning cluster; $w(c)$ is the C -dimensional vector of weights that represents the learning quality of each learning cluster c ; y_{mc} is the final result relevance assigned by the Management Cluster.

5.6 Random Neural Network Extensions

The Random Neural Network model has been extended in various aspects:

The Random Neural Network was extended to multiple classes of signals by Erol Gelenbe et al [138] where each distinct stream is a class of signals in the form of spikes. A spike that represents a signal class when leaves a neuron may be interpreted at the receiving neuron as an excitatory or inhibitory spike of the same class or different class. Firing rates of spikes from some class are proportional to the excitation level of the internal state in that particular class at the emitting neuron. Its learning algorithm defined by Erol

Gelenbe et al [139] applies to both recurrent and feedforward models based on the gradient descent optimization of a cost function. It requires the solution of a system of nC linear and nC nonlinear equations with a $O([nC]^3)$ complexity for the recurrent and $O([nC]^2)$ complexity for the feedforward models.

The Bipolar RNN model presented by Erol Gelenbe et al [140] has positive and negative nodes and symmetrical behaviour of positive and negative signals circulating through the network. Positive nodes accumulate and emit only positive signals whereas negative nodes accumulate and emit only negative signals. Positive signals cancel negative signals at each negative node and vice versa. Connections between different nodes can be positive or negative; a signal leaving a node can move to another node as a signal of same or opposite sign. The Bipolar RNN presents auto associative memory capabilities and universal function approximation properties as defined by Erol Gelenbe et al [141]. The feed forward Bipolar model presented by Erol Gelenbe et al [142] with s hidden layers ($s+2$ in total) can uniformly approximate continuous functions of s variables.

The RNN with synchronised interactions was defined by Erol Gelenbe et al [143], it includes the possibility of neurons acting together on other neurons, synchronised firing between neurons, where one neuron may trigger firing in another one, and cascades of such triggered firings; the model describes the triggering of firing between two neurons and triggered firing by cascades of neurons, including feedback loops in the cascades to include lengthy bursts of firing. Synchronous interactions between two neurons that jointly excite a third neuron as described by Erol Gelenbe et al [144] are sufficient to create synchronous firing by large ensembles of neurons. An n -neuron recurrent network that has both conventional excitatory-inhibitory interactions and synchronous interactions has an $O(n^3)$ gradient descent learning algorithm.

Deep learning with the Random Neural Network presented by Erol Gelenbe et al [145] is based on soma to soma interactions between natural neuronal cells. Clusters are formed of densely packed cells where the firing pattern of one cell immediately excites and provokes firing by neighbouring cells through dense soma-to-soma interactions based on the mathematical properties of the G-Networks and the Random Neural Network; the properties lead to a transfer function that can be exploited for large arrays of cells. The main advantage of the Deep Learning Clusters model is the increment of learning speed whereas the main disadvantage is that neurons do not achieve very low potential values.

5.7 Random Neural Network Applications

The Random Neural Network has been used on different applications.

5.7.1 Optimization

The RNN has been applied to obtain an approximate solution to the solution of several NP-hard optimization problems including the minimum vertex as proposed by Erol Gelenbe et al [146] to find the node cover of smallest possible size; the travelling salesman problem presented by Erol Gelenbe et al [147] to find the shortest closed path in a set of n cities with the constraint that all cities must be visited only once and task to processors assignment for distributed systems as defined by Jose Aguilar [148] with consideration to clustering transactions and dynamic load balancing. Multicast routing was optimized by using the RNN by Erol Gelenbe et al [149], the network is modelled as a weighted, undirected graph where the problem is finding a minimal Steiner tree for the graph given a set of destinations by the best available heuristics: the minimum spanning tree heuristic and the average distance heuristic.

5.7.2 Image Processing and Video Compression

The generation of various artificial image textures with different features is performed by a RNN following Volkan Atalay et al [150] which associates a neuron $n(i,j)$ to each pixel or picture element $p(i,j)$ and each neuron is connected up to eight neighbors. In addition, the RNN was used for texture modelling and synthesis by Erol Gelenbe et al [151] where it learns the weights of a recurrent network directly from the texture image which are then used to generate a synthetic texture that imitates the original one. The RNN extracts morphometric information from Magnetic Resonance Imaging scans of the Human Brain as presented by Erol Gelenbe et al [152] where images are divided into regions characterized by its detailed granular properties to be learned and identified by different trained recurrent RNNs. The RNN is also used to deduce a feature's vertical cross-section from two-dimensional top-down scanning electron microscopy images of the feature surface of a semi-conductor as shown by Erol Gelenbe et al [153]. Image content classification methods, systems and computer programs that repeatedly scan images assigning an array of image with at least one random neural network is patented solution by Erol Gelenbe et al [154] where each scan corresponds to one of multiple texture patterns and each corresponding texture pattern is compared against each of multiple image portions for each of the multiple scans.

The RNN achieves real time adaptive compression of video sources with motion detection following Erol Gelenbe et al [155] which maintains a target quality of the decomposed image specified by the user where the RNN acts as an auto encoder. A set of RNNs that compresses at different compression levels was proposed by Christopher Eric Cramer et al [156] in addition of a simple motion detection and temporal subsampling of frames and decompression where missing frames are then numerically interpolated using approximations. The deliberately drop of frames to reduce network resources presented by Christopher Cramer et al [157] can be compensated by interpolating frames using the RNN with a function approximation.

5.7.3 Cognitive Packet Networks

The Cognitive Packet Networks (CPN), defined by Erol Gelenbe et al [158], has intelligent and learning capabilities where routing and flow control are based on adaptive finite state machines and Random Neural Networks as presented by Erol Gelenbe et al [159]. CPN routing decisions are taken by the packets, rather than in the nodes and protocols. Cognitive packets are assigned goals before entering the network and follow them adaptively while leaning from their own measurements about the network and from the experience of other packets with whom they exchange information. The design of CPN architecture proposed by Erol Gelenbe et al [160,161] has been implemented with a QoS based routing algorithm in test bed for best and worst case performance to demonstrate the capacity of the CPN to adapt to changes in traffic load and link failures.

The CPN has also been tested to transmit Voice by Lan Wang et al [162] delivering better QoS metrics (Delay, jitter and packet desequencing) against Voice over IP. Voice over CPN is an extension of the CPN routing algorithm to support the needs of voice packet delivery presented by Lan Wang et al [163] in the presence of other background traffic flows with the same or different QoS requirements; the implementation is based on Reinforcement Learning to dynamically seek paths that meet the quality requirements of voice communications. Real Time over CPN presented by Lan Wang et al [164] uses QoS goals that match the needs of real-time packet delivery in the presence multiple QoS classes (delay, loss and jitter) for multiple traffic flows simultaneously.

The CPN has been applied in a framework at Internet Service Provider level by Erol Gelenbe et al [165] to optimize QoS requirements. The drift parameter is defined by Antoine Desmet et al [166] as the probability that packets will be forwarded according to the RNN's advice rather than random; Some CPN packets are routed at random to enable the discovery of new routes and to avoid the saturation of the network weights where the

CPN rapidly establishes a backbone of high quality paths and then explores other network areas.

Energy efficiency has also been considered in Ad Hoc Cognitive Packet Networks (AHCPN) by Erol Gelenbe et al [167] where the use of unicast messages for route searching is preferred rather than broadcast; QoS metrics include the energy stored in the nodes. In addition, as presented by Ricardo Lent et al [168], to reduce energy consumption and decrease the interference communications range, payload and acknowledgement packets are transmitted with an adjusted transmission power level. The AHCPN protocol is used in an infrastructure-less indoor emergency response system by Huibo Bi et al [169] to prolong the life time of smart phones and reduce time latency; it adaptively searches optimal communication routes between portable devices and the egress node while providing access to a cloud server. The CPN was patented by Erol Gelenbe [170] as a method of packet switching in a data packet communication environment having a plurality of digital packet transmission stations with interconnected paths.

5.7.4 Self Aware Networks

Self aware networks presented by Erol Gelenbe et al [171,172] observe their own performance using internal checks and measurement methods to make effective autonomous use of these observations for self-management where the pursued Goal is a combination of QoS metrics Delay and Loss. Self aware networks through self monitoring, measurement and intelligent adaptive behaviour simultaneously use a selection of wired and wireless communication networks following Erol Gelenbe [173] to offer different levels of quality-of-service (QoS), including reliability, security and cost; likewise, the communication infrastructure is shared among different networks and users where the resources available fluctuates over time. Nodes can join and leave the self aware network autonomously and discover paths to meet QoS and communication requirements in largely unknown networks as proposed by Erol Gelenbe [174]; nodes monitor and discover the status of other nodes, links, and paths, including traffic level and congestion, so as to update their own relevant information about the paths they need to use, based on criteria specific to their own need.

5.7.5 Software Defined Networks

Software Defined Networks enable network management and allow efficient network configuration to increase network performance and monitoring. An example of Software Defined Network (SDN) is the Cognitive Packet Network as defined by Erol Gelenbe [175]

which it is also a self-aware computer network (SAN). As a SDN, the CPN modifies its behaviour to adaptively accomplish its objectives by observing its own internal performance and the interfaces to the external networks; objectives include discovering services for users, improving Quality of Service (QoS), reduce own energy consumption, compensate for components which fail or malfunction, detect and react to intrusions, and defend itself against attacks.

A software define network platform is developed by Erol Gelenbe et al [176] with a bilateral QoS differentiation between pairs of communicating nodes. Each edge node or user node is a source and a destination at the same time, managing uplink user originated traffic and downlink traffic sent back in response to the uplink; this method allows the communication between end user nodes to alternate dynamically between two roles while offering best effort QoS. Traffic volume asymmetry between the received and sent data is used to trigger changes in the QoS where the lower traffic rate requires short delay QoS and the higher traffic rate requires loss reduction.

A Cognitive Routing Engine (CRE) presented by Frédéric François et al [177] is able to find near-optimal paths for a user-specified QoS while using a very small monitoring overhead and reduced response time for the Software Defined Network controllers and switches. A logically centralized CRE finds the optimal overlay paths when the public Internet is used as the communication means between the overlay nodes from different clouds as described by Frédéric François et al [178]. In addition; the CRE is able to do asymmetric path optimization where the forward path is different from the reverse path for a given data centre pair in order to further improve QoS.

A big data and machine learning for the real-time management of Internet scale quality-of-service (QoS) route optimisation with an overlay network presented by Olivier Brun et al [179] exploits the existing IP protocol with path variations offered by an overlay network. It uses an adaptive overlay system that creates a limited modification of IP routes resulting in lower packet forwarding delays and loss rates.

5.7.6 Smart routing

QoS goals were externally set by network users in Pu Su et [180] and then they were explicitly assigned by a self-aware network to control its own behaviour. Genetic algorithms generate and maintain routes from previously discovered paths as presented in Erol Gelenbe et al [181] by matching their fitness with respect to an established QoS

while combining relevant paths. An analogy between genotypes and networks paths is described by Erol Gelenbe et al [182], it uses the source node of each connection to discover new routes using the crossover operation where each path is considered as the encoding of a genotype. An adaptive genetic algorithm proposed by Erol Gelenbe et al [183] applies a round-robin policy on the best routes acting as a load balancing mechanism to avoid saturation of any given path; it also offers the possibility to gather measurement data comprising a much bigger set of nodes.

The CPN with Reinforcement Learning learns autonomously the best route in the network simply through exploration in a very short time and is able to adapt to a disruption along its current route, switching to the new optimal route in the network as described by Ricardo Lent et al [184], data traffic is only rerouted when a better route is found. A new method to calculate a similar delay performance, proposed by Erol Gelenbe et al [185], is based on a metric that combines path length and buffer occupancy; this approach also reduces header size in packets. QoS metrics were also combined in static and dynamic measurements by Michael Gellman et al [186] such as shortest path and minimum delay. Routing oscillations occur due to the interaction of multiple flows producing packet desequencing where it is assumed that they have a QoS detrimental effect as demonstrated by Erol Gelenbe et al [187]; although if they are controlled by randomizing the route switching they can provide an improved performance.

Software implementations of the CPN routing algorithm are unsuitable in dedicated hardware or devices with low computational abilities due to its complexity; simpler alternative algorithms proposed by Laurence A. Hey [188] match the performance of the original CPN with respect to software and hardware implementation due to their reduced complexity and resource requirements. A FPGA based proof of concept hardware implementation of a dedicated CPN router design was proposed by Laurence A. Hey [189], it demonstrates that the traditional architectural approaches used in high speed IP routers are also applicable to CPN routers. A recursive routing algorithm for CPN developed by Peixiang Liu et al [190] breaks large scale route discovery tables into smaller ones; the solutions to those smaller routing tables are cached in the intermediate nodes of the network, which can be utilized to solve larger scale routing problems. This method reduces network connection establishment time without and the QoS is also improved.

5.7.7 Cybersecurity

Cognitive Packet Networks with a Denial of Service (DoS) Defence Technique was presented by Erol Gelenbe et al [191], it creates a self-monitoring sub-network that surrounds each critical infrastructure node. It is based on a Distributed DoS detection scheme that generates control traffic from the objects of the DDoS attack to the islands of protection where DDoS packet flows are dropped before they reach the critical infrastructure. Each node self-determines two parameters: the maximum bandwidth that it is able to receive and the maximum allocation of bandwidth that it is willing to allocate to any individual flow that crosses it. When a CPN router receives a packet from a flow that has not already seen before, as shown by Erol Gelenbe et al [192], it sends a specific flow acknowledgement packet back to the source along the reverse path and inform the source of its bandwidth allocation. The node monitors all the flows that traverse it and drop some or all of the packets of any flow that exceed this allocation; when the allocation is exceeded, the node informs upstream nodes that packets of this flow shall be dropped or diverted to a safe node for further analysis. The most significant inconvenience of packet dropping for DoS defence is the “collateral damage” of the loss of valid packets due to false alarms as presented by Erol Gelenbe et al [193]. Therefore a more sophisticated defence approach is proposed based on prioritization and checking, in which the probability that a packet is “valid” automatically determines the QoS that it receives.

The CPN Admission Control (AC) algorithm proposed by Georgia Sakellari et al [194] bases its decision on different QoS metrics. The self-observation and self-awareness capabilities of the CPN are used to collect data that allows the AC algorithm to decide whether to admit users based on their QoS needs, the QoS impact that its admission will have on existing connections and the existence of feasible paths for the projected incoming traffic. The CPN AC algorithm defined by Georgia Sakellari et al [195] estimates the new flow impact by probing it at a small rate, therefore, the probe packets will not contribute to the network's congestion. Users specify the QoS constraints they need in order to obtain the network service they require for a successful connection where each user can have different QoS requirements as proposed by Erol Gelenbe et al [196]. The decision of whether to accept a new connection was presented by Erol Gelenbe et al [197]; it is made using an algebra of QoS metrics based on Warshall's algorithm, which searches for a path with acceptable QoS values that can accommodate the new flow.

A combination of Bayesian decision making and Random Neural Networks (RNN) is applied by Georgios Loukas et al [198] to the detection of Denial of Service networking attack; the method measures different instantaneous behaviour following George Loukas et al [199] where the longer term statistical variables describe the incoming network traffic; the method acquires a probability density function estimating and evaluating the likelihood ratio where the detection decision making step measures the features of the incoming traffic according to each feature which are combined using both feedforward and recurrent architectures of the RNN. Four different implementations of the detection decision making process were used by Gülay Öke et al [200]: Average likelihood estimation, RNN with likelihood values, RNN with histogram categories and RNN with actual values. Seven different implementations provided by Gülay Öke et al [201] are compared where experimental results are evaluated in a large networking testbed.

5.7.8 Gene Regulatory Networks

A probability model for Gene Regulatory Networks (GRN) was proposed by Erol Gelenbe [202], it is represented by the dynamics of the concentration levels of each agent in the network; this approach includes the representation of excitatory and inhibitory interactions between agents, second-order interactions which allow any two agents to jointly act on other agents and Boolean dependencies between them. The result is an exact solution in “product form” where the joint equilibrium probability distribution of the concentration for each gene is the product of the marginal distribution for each of the concentrations.

The Bayesian approach with a prior Gibbs distribution was proposed by Haseong Kim et al [203], it provides a convenient way to integrate multiple sources of biological data to construct large-scale gene regulatory networks; a reverse engineering approach is based on Bayesian model averaging technique that ensembles appropriate regression models. A Bayesian model averaging based networks (BMAnet) defined by Haseong Kim et al [204] is used to build reliable and large-scale gene regulatory networks able to identify disease candidate genes. A new pathway analysis approach to detect differentially behaving pathways in abnormal conditions following Haseong Kim et al [205] is based on G-network theory where gene regulatory network model parameters are estimated from normal and abnormal samples using optimization techniques with corresponding constraints. A stochastic gene expression model that describes the switch-like behaviours of a gene presented by Haseong Kim et al [206] is based on Hill functions to the conventional Gillespie algorithm.

5.7.9 Web Search

An Intelligent Internet Search Assistant (ISA) was defined by Will Serrano et al [207] and fully presented on this Thesis, The Intelligent Agent is based on the Random Neural Network measures the user's relevance and selects the results from one or more Web search engines using the preferences that it has learned. The Internet Search Assistant acts as an interface between the user and Big Data search engines following Will Serrano [208]: Web search engines, metasearch engines, academic databases and recommender systems. The Intelligent Agent emulates a brain learning structure as defined by Will Serrano et al [209] when it is based on the Random Neural Network with Deep Learning clusters where ISA measures and evaluates Web result relevance by associating each Deep Learning cluster with a different Web Search Engine. A Deep Learning Cluster that performs as a Management Cluster was defined by Will Serrano et al [210], it is included to decide the final result relevance based on the inputs from the Deep Learning clusters.

6 Internet Search Assistant Model

6.1 Intelligent Search Assistant Model

Search for information or meaning needs three elements: an M-dimensional universe of X entities or ideas to be searched, a high level query that specifies the N-properties or concepts requested by a user and a method that searches and selects Y entities from the universe showing the first Z results to user according to an algorithm or rule. Each entity or concept in the universe is distinct from the others in some recognizable way; for instance two entities may be different just in the date or time-stamp that characterizes the time when they were last stored or in the ownership or origin of the entities. On the other hand, we consider concepts to be distinct if they contain any different meaning, even though if they are identical with respect to a user's query.

We consider the universe we are searching within as a relation U that consists of a set of X M-tuples, $U = \{v_1, v_2 \dots v_X\}$, where $v_i = (l_{i1}, l_{i2} \dots l_{iM})$ and l_i are the M different attributes for $i=1,2..X$. The relation U is a very large relation consisting on $M \gg N$ attributes (Fig. 27).

The important concept in the development of this thesis is a query can be defined as $R_t(n(t)) = (R_t(1), R_t(2), \dots, R_t(n(t)))$ where $n(t)$ is a variable N-dimension attribute vector with $1 < N < M$ and t is the search iteration being $t > 0$; $n(t)$ is variable so that attributes can be added or removed based on their relevance as the search progresses, i.e. as t increases.

Each $R_t(n(t))$ takes its values from the attributes within the domain $D(n(t))$, where D is the corresponding domain that forms the universe U. Thus $D(n(t))$ is a set of properties or meanings based in words or integers, but also words in another language, or a set of icons, images or sounds.

The answer A to the query $R_t(n(t))$ is a set of Y M-tuples $A = \{v_1, v_2 \dots v_Y\}$ where $v_o = (l_{o1}, l_{o2} \dots l_{oM})$ and l_o are the M different attributes for $o=1,2..Y$. Our Intelligent Search Assistant only shows to the user the first set of Z tuples that have the highest neuron potentials among the set of Y tuples. The neuron potential that represents the relevance of each M-tuple v_o is calculated at each t iteration.

The user or the high level query itself is limited mainly by two main factors: the user's lack of information about all the attributes that form the universe U of entities and ideas, or the user's lack of precise knowledge about what he is looking for.

Universe U is a set of X M-tuples	$U = \{v_1, v_2 \dots v_X\}$
M-tuple v_i is a M-dimension vector	$v_i = (l_{i1}, l_{i2} \dots l_{iM}) \quad l_i = M \text{ Attributes} \quad i = 1, \dots, X$
Query $R_t(n(t))$ is a variable N-dimension vector	$R_t(n(t)) = (R_t(1), R_t(2), \dots, R_t(n(t)))$
- $n(t)$ is variable N-dimension attribute vector with $1 < N < M$;	
- t is search iteration.	
Answer A is a set of Y M-tuples	$A = \{v_1, v_2 \dots v_Y\}$
M-tuple v_o is a M-dimension vector	$v_o = (l_{o1}, l_{o2} \dots l_{oM}) \quad l_o = M \text{ Attributes} \quad o = 1, \dots, Y$
Domain D	
$N < M \quad Y < X \quad D(n(t)) \subset D(m) \quad D'(n(t)) \subset D(m)$	

Figure 27: Internet Search Assistant Model

6.2 Result Cost Function

We consider the universe U is formed of the entire results that can be searched. We assign each result provided by a search engine to an M-tuple v_o of the answer set A . We calculate the result relevance based on a cost function described within this section.

The query $R_t(n(t))$ is a variable N-dimension vector that specifies the attributes the user consider relevant. The number of dimensions of the attribute vector $n(t)$ varies as the iteration t increases. Our Intelligent Search Assistant associates an M-tuple v_o to each result provided by the Search Engine creating an answer set A of Y M-tuples. Search Engines select their results from the universe U .

We apply our cost function to each result or M-tuple v_o from the answer set A of Y M-tuples. We consider each v_o as a M-dimensional vector. The cost function is firstly calculated based on the relevant N attributes the user introduced on the High Level Query, $R_1(n(1))$ within the domain $D(n(1))$ however, as the search progresses, $R_t(n(t))$, attributes may be added or removed based on the perceived relevance within the domain $D'(n(t))$.

We calculate the overall Result Score, RS , by measuring the relationship between the values of its different attributes:

$$RS = RV * HW \quad (1)$$

where RV is the Result Value which measures the result relevance and HW the Homogeneity Weight.

The Homogeneity Weight (HW) rewards results that have relevance or scores dispersed along their attributes. This parameter is also based on the idea that the first dimensions or attributes of the user query $R_t(n(t))$ are more important than the last ones:

$$HW = \frac{\sum_{n=1}^N HF[n]}{N} \quad (2)$$

where $HF[n]$, homogeneity factor, is a N-dimension vector associated to the result and n is the attribute index from the query $R_t(n(t))$:

$$HF[n] = \begin{cases} \frac{N-n}{N} & \text{if } SD[n] > 0 \\ 0 & \text{if } SD[n] = 0 \end{cases} \quad (3)$$

We define Score Dimension $SD[n]$ as a N-dimension vector that represents the attribute values of each result or M-tuple v_o in relation with the query $R_t(n(t))$.

The Result Value (RV) is the sum of each dimension individual score:

$$RV = \sum_{n=1}^N SD[n] \quad (4)$$

where n is the attribute index from the query $R_t(n(t))$.

Each dimension of the Score Dimension vector $SD[n]$ is calculated independently for each n-attribute value that forms the query $R_t(n(t))$:

$$SD[n] = S * PPW * RPW * DPW \quad (5)$$

We consider only three different types of domains of interest: words, numbers (as for dates and times) and prices.

S is the score calculated depending if the domain of the attribute is a word (WS), number (NS) or price (PS).

If the domain $D(n)$ is a word, our ISA calculates the score Word Score (WS) following the formula:

$$S = \frac{WR}{NW} \quad (6)$$

where the value of WR is 1 if the word of the n-attribute of the query $R_t(n(t))$ is contained in the search result or 0 otherwise. NW is the number of words in the search result.

If the domain $D(n)$ is a number, our ISA selects the best Number Score (NS) from the numbers they are contained within the search result that maximizes the cost function:

$$S = \frac{\left(1 - \left(\frac{|DV - RV|}{|DV| + |RV|}\right)\right)}{NN} \quad (7)$$

where DV is the value of the n-attribute of the query $R_t(n(t))$, RV is the value of a number in the result and NN is the total number of numbers in the result.

If the domain $D(n)$ is a price, our ISA chooses the best Price Score (PS) from the prices in the result that maximizes the cost function:

$$S = \frac{\left(\frac{DV}{RV}\right)}{NP} \quad (8)$$

where DV is value of the n-attribute of the query $R_t(n(t))$, RV is the value of a price in the result and NP is the total number of prices in the result.

We penalize if the search result provides unnecessary information by dividing the score by the total amount of elements in the Web result.

The Position Parameter Weight (PPW) is based on the idea that an attribute value shown within the first positions of the search result is more relevant than if it is shown at the final:

$$PPW = \frac{NC - DVP}{NC} \quad (9)$$

where NC is the number of characters in the result and DVP is the position within the result where the value of the dimension is shown.

The Relevance Parameter Weight (RPW) incorporates the user's perception of relevance by rewarding the first attributes of the query $R_t(n(t))$ as highly desirable and penalising the last ones:

$$RPW = 1 - \frac{PD}{N} \quad (10)$$

where PD is the position of the n-attribute of the query $R_t(n(t))$ and N is the total number of dimensions of the query vector $R_t(n(t))$.

The Dimension Parameter Weight (DPW) incorporates the observation of user relevance with the value of domains $D(n(t))$ by providing a higher score on the domain values that the user has filled more on the query:

$$DPW = \frac{NDT}{N} \quad (11)$$

where NDT is the number of dimensions with the same domain (word, number or price) on the query $R_t(n(t))$ and N is the total number of dimensions of the query vector $R_t(n(t))$.

We assign this final Result Score value (RS) to each M-tuple v_o of the answer set A. This value is used by our ISA to reorder the answer set A of Y M-tuples, showing to the user the first set of Z results which have the higher potential value.

6.3 User iteration

The user, based on the answer set A can now act as an intelligent critic and select a subset of P relevant results, C_p , of A. C_p is a set that consists of P M-tuples $C_p = \{v_1, v_2 \dots v_p\}$. We consider v_p as a vector of M dimensions; $v_p = (l_{p1}, l_{p2} \dots l_{pM})$ where l_p are the M different attributes for $p=1,2..P$. Similarly, the user can also select a

subset of Q irrelevant results, C_Q of A , $C_Q = \{v_1, v_2 \dots v_Q\}$. We consider v_q as a vector of M dimensions; $v_q = (l_{q1}, l_{q2} \dots l_{qM})$ where l_q are the M different attributes for $q=1,2..Q$.

Based on the user iteration (Fig. 28), our Intelligent Search Assistant provides to the user with a different answer set A of Z M -tuples reordered to MD , the minimum distance to the Relevant Centre for the results selected, following the formula:

$$RCP[n] = \frac{\sum_{p=1}^P SD_p[n]}{P} = \frac{\sum_{p=1}^P l_{pn}}{P} \quad (12)$$

where P is the number of relevant results selected, n the attribute index from the query $R_t(n(t))$ and $SD_p[n]$ the associated Score Dimension vector to the result or M -tuple v_p formed of l_{pn} attributes.

An equivalent equation applies to the calculation of the Irrelevant Centre Point:

$$ICP[n] = \frac{\sum_{q=1}^Q SD_q[n]}{Q} = \frac{\sum_{q=1}^Q l_{qn}}{Q} \quad (13)$$

where Q is the number of irrelevant results selected, n the attribute index from the query $R_t(n(t))$ and $SD_q[n]$ the associated Score Dimension vector to the result or M -tuple v_q formed of l_{qn} attributes.

Our Intelligent Search Assistant reorders the retrieved Y set of M -tuples showing only to the user the first Z set of M -tuples based on the lowest distance (MD) between the difference of their distances to both Relevant Centre Point (RD) and the Irrelevant Centre Point (ID) respectively:

$$MD = RD - ID \quad (14)$$

where MD is the result distance, RD is the Relevant Distance and ID is the Irrelevant Distance.

The Relevant Distance (RD) of each result or M -tuple v_q is formulated as below:

$$RD = \sqrt{\sum_{n=1}^N (SD[n] - RCP[n])^2} \quad (15)$$

where $SD[n]$ is the Score Dimension vector of the result or M-tuple v_q and $RCP[n]$ is the coordinate of the Relevant Centre Point. Equivalent equation applies to the calculation of the Irrelevant Distance:

$$ID = \sqrt{\sum_{n=1}^N (SD[n] - ICP[n])^2} \quad (16)$$

where $SD[n]$ is the Score Dimension vector of the result or M-tuple v_q and $ICP[n]$ is the coordinate of the Irrelevant Centre Point.

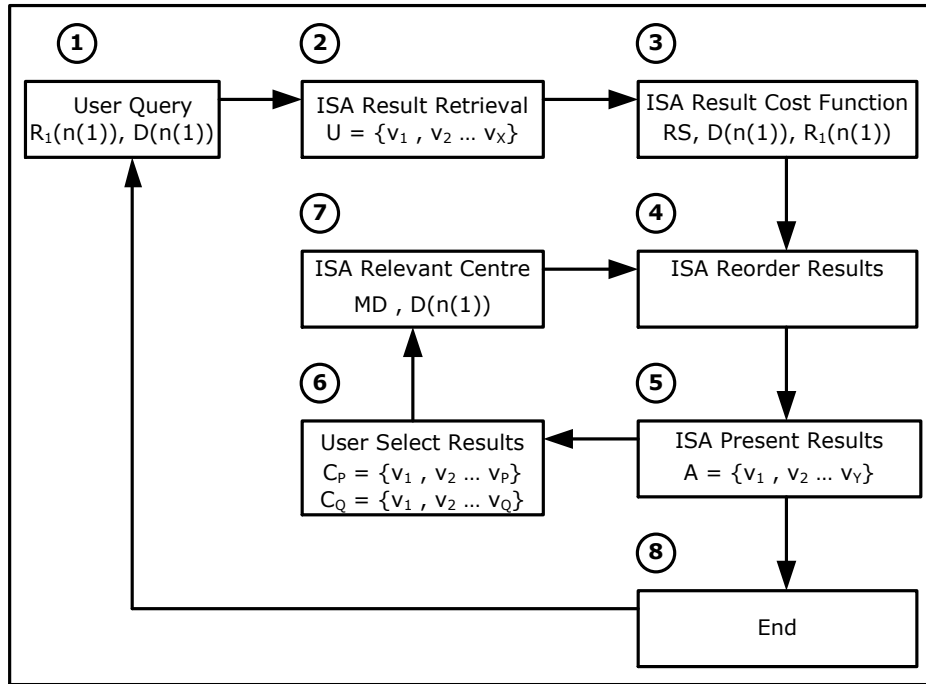


Figure 28: Intelligent Search Assistant User Iteration

Therefore we are presenting an iterative search progress that learns and adapts to the user relevance based on the dimensions or attributes the user has introduced on the initial query and the later result selection from the presented result list.

6.4 Dimension Learning

The answer set A to the query $R_1(n(1))$ is based on the N dimension query introduced by the user however results are formed of M dimensions therefore the subset of results the user has considered as relevant may have other relevant concepts hidden the user did not consider on the original query.

We consider the domain $D(m)$ or the M attributes from which our universe U is formed as the different independent words that form the set of Y results retrieved from the search engines (Fig. 29). Our cost function is expanded from the N attributes defined in the query $R_1(n(1))$ to the M attributes that form the searched results. Our Score Dimension vector, $SD[m]$, is now based on M -dimensions. An analogue attribute expansion is applied to the Relevance Centre Calculation, $RCP[m]$.

The query $R_1(n(1))$ is based on the N -Dimension vector introduced by the user however the answer set A consist of Y M -tuples. The user, based on the presented set A , selects a subset of P relevant results, C_p and a subset of Q irrelevant results, C_Q .

Let's consider C_p as a set that consists of P M -tuples $C_p = \{v_1, v_2 \dots v_p\}$ where v_p is a vector of M dimensions; $v_p = (l_{p1}, l_{p2} \dots l_{pm})$ and l_p are the M different attributes for $p=1,2..P$.

We describe our dimension learning per result using a multidimensional vector instead of per set using a multidimensional array for clarity.

The M -dimension vector Dimension Average, $DA[m]$, is the average value of the m -attributes for the selected relevant P results:

$$DA[m] = \frac{\sum_{p=1}^P SD_p[m]}{P} = \frac{\sum_{p=1}^P l_{pm}}{P} \quad (17)$$

where P is the number of relevant results selected, m the attribute index of the relation U and $SD_p[m]$ the associated Score Dimension vector to the result or M -tuple v_p formed of l_{pm} attributes.

We define ADV as the Average Dimension Value of the M -dimension vector $DA[m]$:

$$ADV = \frac{\sum_{m=1}^M DA[m]}{M} \quad (18)$$

where M is the total number of attributes that form the relation U.

The correlation vector $\sigma[m]$ is the difference between the dimension values of each result with the average vector:

$$\sigma[m] = \frac{\sum_{p=1}^P |SD_p[m] - DA[m]|}{P} = \frac{\sum_{p=1}^P |l_{pm} - DA[m]|}{P} \quad (19)$$

where P is the number of relevant results selected, m the attribute index of the relation U and $SD_p[m]$ the associated Score Dimension vector to the result or M-tuple v_p formed of l_{pm} attributes.

We define C as the average correlation value of the M-dimensions of the vector $\sigma[m]$:

$$C = \frac{\sum_{m=1}^M \sigma[m]}{M} \quad (20)$$

where M is the total number of attributes that form the relation U.

We consider an m-attribute relevant if its associated Dimension Average value $DA[m]$ is larger than the average dimension ADV and its correlation value $\sigma[m]$ is lesser than the average correlation C. We have therefore changed the relevant attributes of the searched entities or ideas by correlating the error value of its concepts or properties represented as attributes or dimensions.

On the next iteration, the query $R_2(n(2))$ is formed by the attributes our ISA has considered relevant. The answer to the query $R_2(n(2))$ is a different set A of Y M-tuples. This process iterates until there are not new relevant results to be shown to the user.

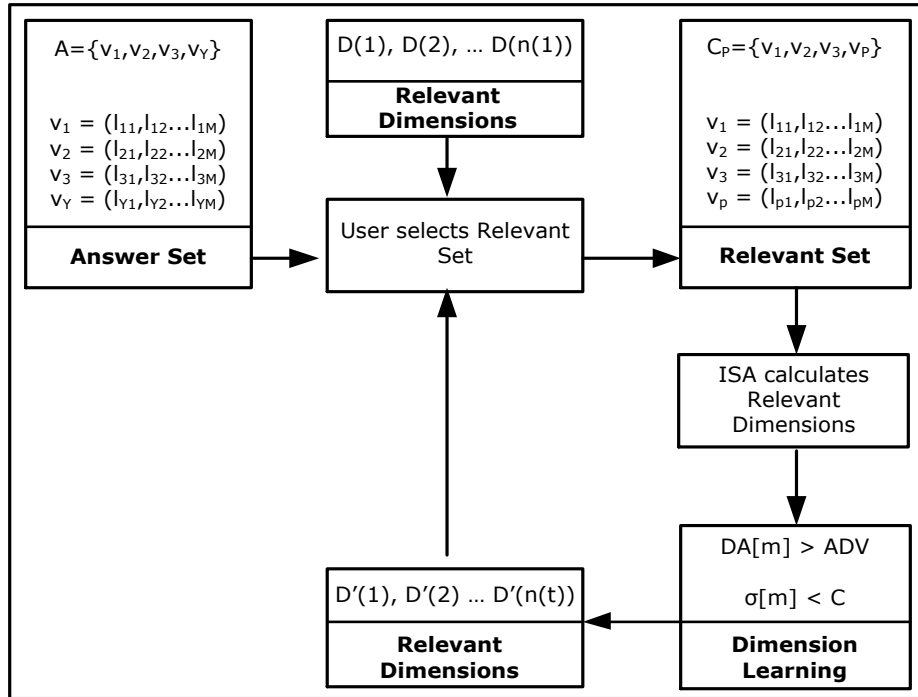


Figure 29: Intelligent Search Assistant Dimension Learning

6.5 Gradient Descent Learning

Gradient Descent learning is based on the adaptation to the perceived user interests or understanding of meaning by correlating the attribute values of each result to extract similar meanings and cancel superfluous ones.

ISA Gradient Descent learning algorithm is based on a recurrent model (Fig. 30). The inputs $i = \{i_1, \dots, i_p\}$ are the M-tuples v_p corresponding to the selected relevant result subset C_p and the desired outputs $y = \{y_1, \dots, y_p\}$ are the same values as the input. Our ISA then obtains the learned random neural network weights, calculates the relevant dimensions and finally reorders the results according to the minimum distance to the new Relevant Centre Point focused on the relevant dimensions.

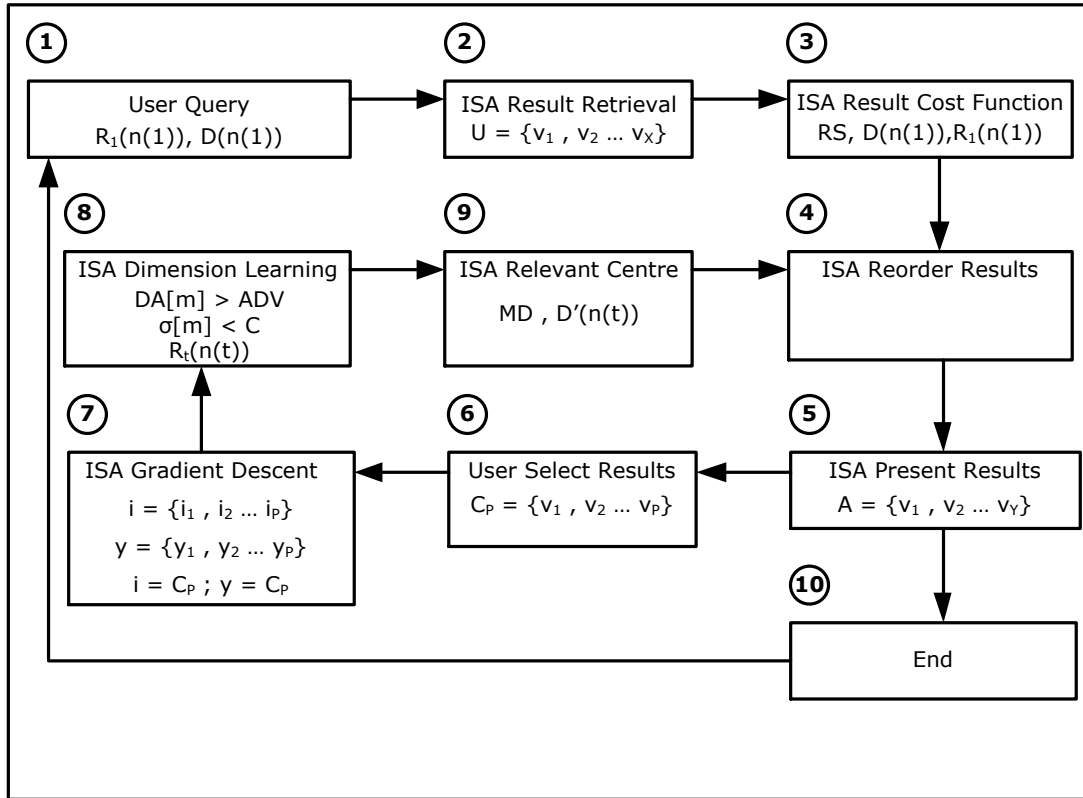


Figure 30: Intelligent Search Assistant Model – Gradient Descent Learning

6.6 Reinforcement Learning

The external interaction with the environment is provided when the user selects the relevant result set C_p . Reinforcement Learning adapts to the perceived user relevance by incrementing the value of relevant dimensions and reducing it for the irrelevant ones (Fig. 31). Reinforcement Learning modifies the values of the m attributes of the results, accentuating hidden relevant meanings and lowering irrelevant properties.

Our ISA associates the Random Neural Network weights to the answer set A ; $W = A$ and ISA updates the network weights W by rewarding the result relevant attributes following our proposed reinforcement:

$$w(p,m) = l_{pm}^{s-1} + l_{pm}^{s-1} * \left(\frac{l_{pm}^{s-1}}{\sum_{m=1}^M l_{pm}^{s-1}} \right) \quad (21)$$

where p is the result or M -tuple v_p formed of l_{pm} attributes, m the result attribute index, M the total number of attributes and s the iteration number.

ISA also updates the network weights by penalising the result irrelevant attributes following our proposed punishing:

$$w(p,m)=l_{pm}^{s-1}-l_{pm}^{s-1}*\left(\frac{l_{pm}^{s-1}}{\sum_{m=1}^M l_{pm}^{s-1}}\right) \quad (22)$$

where p is the result or M -tuple v_p formed of l_{pm} attributes, m the result attribute index, M the total number of attributes and s the iteration number.

Our ISA then recalculates the potential of each of the result based on the updated network weights and reorders them, showing to the user the results which have a higher potential or score.

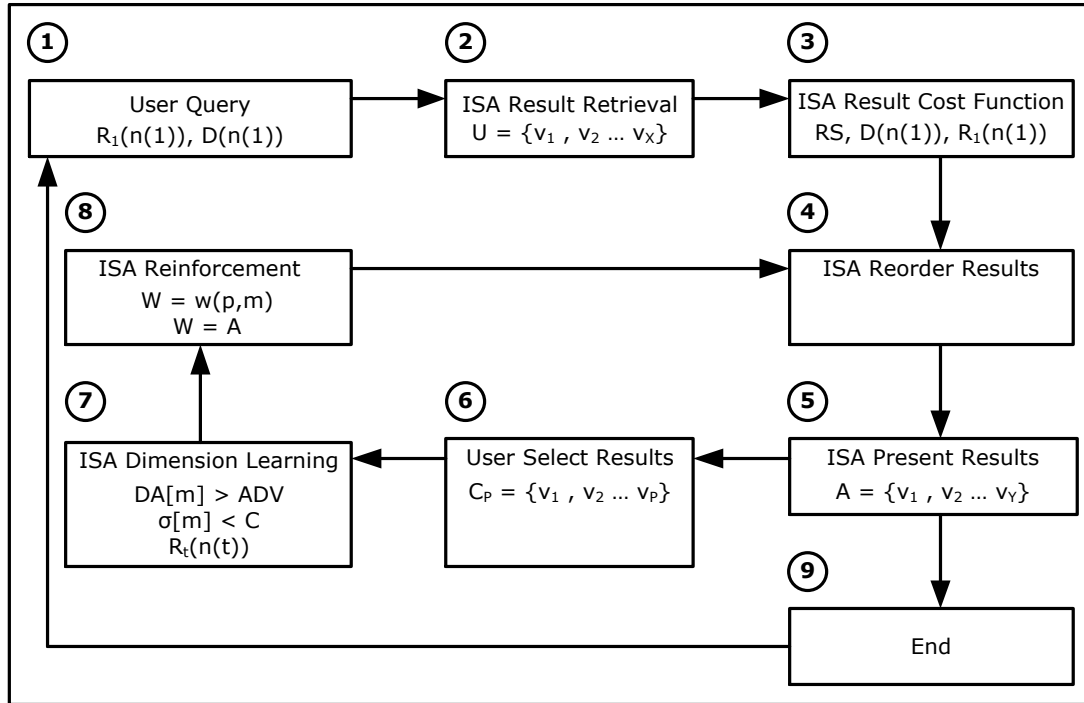


Figure 31: Intelligent Search Assistant Model – Reinforcement Learning

7 Unsupervised Evaluation

This section presents an unsupervised evaluation for ISA Cost function based on the Spearman Rank Correlation Coefficient between a Result Master List (RML) and the results provided by a Web Search Engine (WSE). The Result Master List is calculated by adding the rank scores assigned to the same results provided by the different Web Search Engines or ranks assigned to results by real human validators. Validators are 10 personal friends from Imperial College students, researchers and London young professionals degree educated.

The evaluation presented of this section compares ISA performance individually against Google search, generally against other Web Search Engines (Google, Yahoo, Ask, Lycos and Bing) and Metasearch Engines (Metacrawler and Ixquick).

We have been based on available Web Search Engines in the current market. Although Yahoo search is powered by Bing and Ask.com has outsourced their search engine to an unknown third party; we have considered them independent and uncorrelated as they run on different Web pages; unless they provide the same results on the same order.

7.1 Implementation

The software utilized to program our Intelligent Search Assistant is Java. The API between Java and the Internet is through Selenium – Web driver. Selenium is a software package developed by Google (Fig. 32). The interface with the user has been achieved by programming a Java Swing-GUI. The operating system used is Fedora based on a Linux kernel.

Selenium Web Driver is a tool for automating Web applications based on their HTML or CSS properties. It has different drivers according to the browser installed in the computer. The main options have been HtmlUnit which it is faster as it does not open a browser window and Google Chrome Driver which it is possible to track on a browser the way the program interacts with the Web Site. Chrome driver is supported by the Chromium project itself.

The main disadvantage of the Selenium Web Automation project is its early stage of deployment. Although it is very reliable sometimes it delivers unforeseen executable

errors. Web interaction depends on the page loading speed (RAM memory, CPU and internet connection) where we have added implicit waits to synchronize our Java executable with the Web pages.

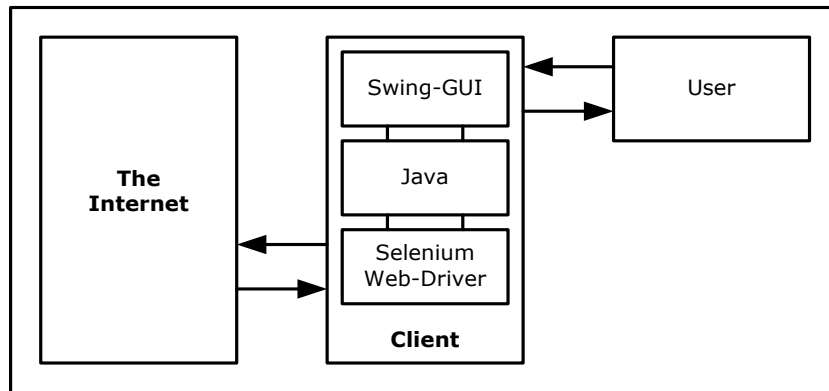


Figure 32: ISA Client side implementation

The Internet Search Assistant starts the main program where the user fills the dimensions of its high level query, the preferred Web search engines and the number of results to be retrieved (Fig. 33). There is the option to search results only in a particular site.

Intelligent Search Assistant

Dimension 1

Dimension 2

Dimension 3

Dimension 4

Dimension 5

Dimension 6

Dimension 7

Dimension 8

Dimension 9

Dimension 10

☐ Google
 ☐ Yahoo
 ☐ Ask
 ☐ Lycos
 ☐ Bing
 ☐ Metacrawler
 ☐ Ixquick
 ☐ Yandex

Search!

Load

Save

Site:

Number of results:

10

20

90

100

Like

Dislike

Reorder

Search Engine:

Action

Results:

Figure 33: ISA Client interface

We could have implemented our Intelligent Search Assistant as a Web search engine however due the high cost required to crawl, parse and index the Web, we have decided to develop our algorithm with its cost function in a Metasearch engine where we extract the snippets provided by the different Web search engines; we are therefore not indexing the Web neither ranking entire Web pages but taking a representation of it.

We cluster results per Web Site; this avoids jamming where Web sites with many pages with similar information floods the result list forcing the results of other Web sites with relevant information into lower positions. Our ISA also removes sites like Youtube and Wikipedia unless these names are specified in the high level query.

7.2 Spearman's Rank Correlation Coefficient

The Spearman Rank Correlation Coefficient measures the statistical dependence between the rankings of a Result Master List (RML) and Web Search Engine (WSE):

$$r_{WSE} = \rho_{rg_{RML}, rg_{WSE}} = \frac{\text{cov}(rg_{RML}, rg_{WSE})}{\sigma_{rg_{RML}} \sigma_{rg_{WSE}}} \quad (1)$$

where ρ represents the Pearson correlation coefficient applied to the RML and WSE, $\text{cov}(rg_{RML}, rg_{WSE})$ is the covariance of the RML and WSE and σ_{rg_x} , σ_{rg_y} are the standard deviations of the RML and WSE.

We use this value to evaluate Web Search performance; the closer the r_{WSE} value to 1 the better search the Web Search Engine has delivered.

7.3 Google Search

We have selected a high level query, $R_t(n(t))$, and retrieved Y results from only one Web search engine, Google. We have asked some human validators to select a subset of Z more applicable results and rank them from Z , as the most relevant, to 1, as the most irrelevant. Results not selected are scored 0. To generate the Result Master List, which consists on Z elements, we have simply added the values provided from the different validators to each result where the most relevant result have the greatest score.

7.3.1 Experimental Results

We have run our ISA with the high level query “**Flight New York Holiday 2013 £500**” to be searched on Google retrieving the first 100 results; this query contains the three different dimension types we have defined. We have asked the validators to select a set of 10 results from the 100 option list obtained and score them consecutively from 10 as the most relevant to 1 as the most irrelevant; if a result is not selected, it scores 0 points. We then add the scores provided by each validator to each result to produce the Result Master List. We consider 100 results to be a number large enough to provide a deep search selection while avoiding result clutter. The Result Master List is shown on Table 1. We have also included the rank that Google has provided them on its search and the Spearman Rank Correlation Coefficient at the end.

Table 1: Google search evaluation – Master Result List

Master Rank	Page	Google Rank
1	www.statravel.co.uk/cheapest-flights-new-york.htm	1
2	www.globehunters.com > Flights	30
3	www.justtheflight.co.uk > Flights > North America > USA	50
4	www.cheapflights.co.uk/handpicked-travel-deals/	25
5	www.lowcostholidays.com/	3
6	www.britishairways.com/travel/flight-deals/public/en_gb	19
7	www.flightcentre.co.uk > Holidays > Usa Canada	61
8	www.globehunters.com > Holidays	18
9	www.statravel.co.uk/virgin-atlantic-flights.htm	87
10	www.carltonleisure.com/travel/flights/first-class/united.../new-york/	28
	r_s: 1.0	

We notice Google provides only 2 results from its 10 best ones based on the Result Master List. The first ten results obtained from Google when we have searched the high level query are shown on Table 2 where the value in brackets represents the order in the master list if the result is within the top ten values. We have included the result score from our cost function that due to its reduced value it is normalized by 100 and the Spearman Rank Correlation Coefficient.

Table 2: Google search evaluation – Google first 10 results

Google Rank	Page	Result Value
1 (1)	www.statravel.co.uk/cheapest-flights-new-york.htm	1.526
2	www.travelsupermarket.com/c/holidays/summer/2013/	1.117
3 (5)	www.lowcostholidays.com/	1.316
4	www.thomascook.com/promotional-terms-conditions/	0.0835
5	www.thomascook.com/	1.443
6	www.thomascook.com/lp/1xd-beach-holidays/	0.997
7	www.thomascook.com > Holidays > Europe	1.034
8	www.barrheadtravel.co.uk/holidays/usa	0.568
9	www.tripadvisor.co.uk > ... > New York (NY) > Forums	1.567
10	www.tripadvisor.co.uk > ... > New York City Holiday Apartments	1.018
	r_s: 0.9437	

Our ISA applies the result cost function we have defined in the previous section reordering the options using this value. The first ten results retrieved from ISA are shown on Table 3; we have included also its associated normalized result score and the Spearman Rank Correlation Coefficient.

Table 3: Google search evaluation – ISA first 10 results

ISA Rank	Page	Result Value
1 (7)	www.flightcentre.co.uk > Holidays > Usa Canada	2.597
2	www.flightcentre.co.uk/travel-guides/region/New+York+City	2.074
3 (3)	www.justtheflight.co.uk > Flights > North America > USA	2.024
4	www.easyvoyage.co.uk/flights/best-prices-boh-nyc	2.021
5	www.homeaway.co.uk > World > North America > USA	1.888
6	www.flightcentre.co.uk/deals/holidays/new-york?...holiday_type...	1.845
7 (10)	www.carltonleisure.com/travel/flights/first-class/united.../new-york/	1.843
8	www.easyvoyage.co.uk/flights/best-prices-bfs-nyc	1.771

9	www.letsgo2.com > Holidays > USA	1.673
10 (2)	www.globehunters.com > Flights	1.603
	r_s: 0.9261	

Our ISA algorithm provides four results from its ten best ones that are included within the Result Master List. Figure 34 shows the r_s values for the Result Master List, Google and ISA.

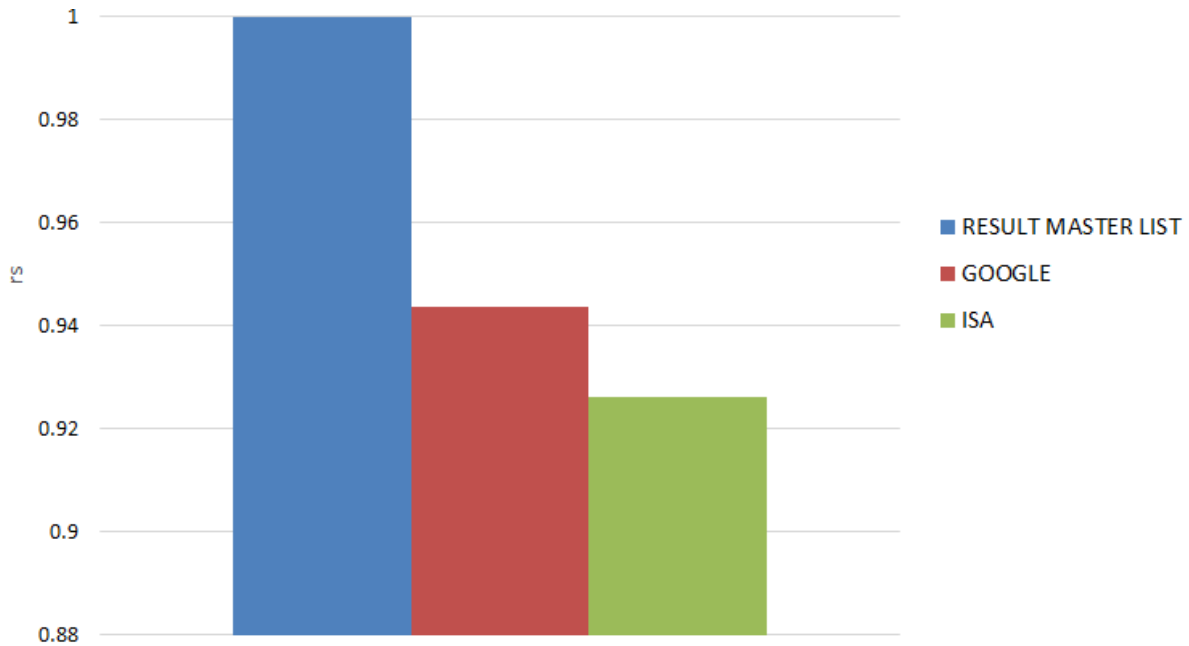


Figure 34: Google Search evaluation

Initially our ISA has scored worse than Google (Fig. 34), mainly because Google has provided the best result on its first position. It is noticed that Google does not show results considered as relevant by the users on its first places and even it ranks on its first ten locations irrelevant results several times from Trip Advisor and Thomas Cook.

7.4 Web Search Evaluation

We select a high level query, $R_t(n(t))$, and then ISA searches for the different options by using X different Web search engines to retrieve the first Y results from each one. We have scored Y points to the Web site result that is displayed in the top position and 1 point to the Web site result that is shown in the last position. If Web search engines provide different results from the same Web site on several positions we cluster them by choosing only the higher ranking one and discarding, or scoring 0, the others in order to penalize result jamming. We then generate the Result Master List by adding the score

value of each Web site result value retrieved from the X different Web search engines. We have therefore created a Result Master List with $X*Y$ elements where the most relevant result have the greatest score.

7.4.1 Experimental results

Our ISA acquires 10 different high level queries based on the travel industry from a user; some of them only vary in one dimension value so we can detect correlations between results. Our ISA retrieves the first 30 results from each of the main Web search engine driver programmed (Google, Yahoo, Ask, Lycos and Bing), we have therefore scored 30 points to the Web site result that is displayed in the top position, 1 point to the Web site result that is shown in the last position and 0 points to each of the result that belongs to the same Web site and it is shown more than once.

After we have scored the 150 results provided by the 5 different Web search engines, we combine them by adding the scores of the results that have the same Web site and rank them to generate the Result Master List. We have done this evaluation exercise for each high level query. The values of Web search quality, r_s , with its associated Standard Deviation σ and 95% Confidence Range from each of the Web search Engines selected and ISA for the different high level queries are shown on Table 4.

Table 4: Web search evaluation

Query Q1: Flight New York holiday summer 2013 £500					
Google	Yahoo	Ask	Lycos	Bing	ISA
-0.1048	0.2667	-0.3782	0.1606	0.2667	0.1297
Query Q2: Hotel New York holiday summer 2013 £500					
Google	Yahoo	Ask	Lycos	Bing	ISA
-0.1150	0.0596	-0.1951	-0.4692	-0.3364	-0.1379
Query Q3: Car rental New York holiday summer 2013 \$200					
Google	Yahoo	Ask	Lycos	Bing	ISA
-0.3370	0.0979	-0.2525	-0.1462	0.1689	0.0552
Query Q4: Vietnamese restaurants New York					
Google	Yahoo	Ask	Lycos	Bing	ISA
0.0703	0.3795	-0.2641	0.0957	0.3170	-0.2338
Query Q5: Modern art galleries New York					
Google	Yahoo	Ask	Lycos	Bing	ISA
-0.3717	0.2027	-0.2427	0.2216	0.2071	0.0478
Query Q6: Flight Tokio holiday summer 2013 £500					
Google	Yahoo	Ask	Lycos	Bing	ISA
-0.2340	0.1112	-0.2536	-0.8948	0.1112	-0.0725

Query Q7: Flight Berlin holiday summer 2013 £50					
Google	Yahoo	Ask	Lycos	Bing	ISA
0.1878	0.3435	0.3197	-0.0205	-0.0096	0.0507
Query Q8: South East Asia holiday travel pack					
Google	Yahoo	Ask	Lycos	Bing	ISA
0.2036	0.1931	0.0877	-0.3061	0.1789	-0.4979
Query Q9: London attractions summer					
Google	Yahoo	Ask	Lycos	Bing	ISA
0.1815	0.2436	0.3137	-0.2558	0.1844	-0.3464
Query Q10: London music festivals					
Google	Yahoo	Ask	Lycos	Bing	ISA
0.4251	0.3188	0.1869	-0.2013	0.2865	0.2278

With average values:

Metric	Google	Yahoo	Ask	Lycos	Bing	ISA
r_s	-0.0094	0.2217	-0.0678	-0.1816	0.1375	-0.0777
σ	0.2635	0.1089	0.2656	0.3328	0.1908	0.2268
95%CR	0.1633	0.0675	0.1646	0.2062	0.1183	0.1406

Figure 35 shows the r_s with its associated Standard Deviation σ and the 95% Confidence Interval that corresponds to $r_s \pm 95\%CR$ for the different Web Search Engines, including our ISA.

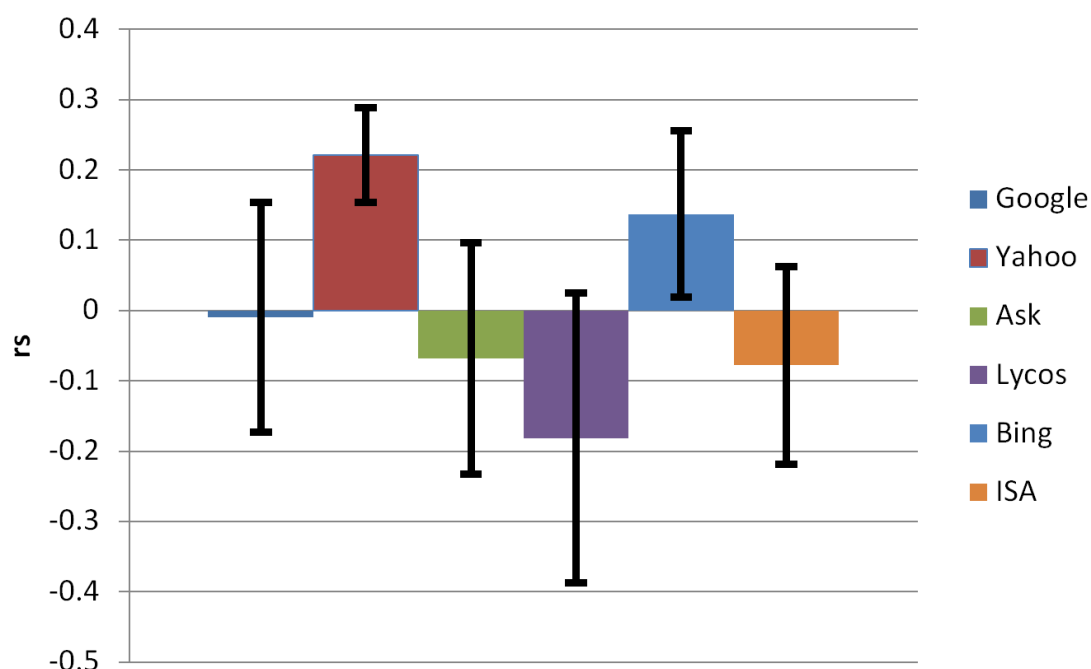


Figure 35: Web search evaluation – Average Values

Our ISA has not scored better than the average of the other Web search engines (Fig. 35), the main reason is the Result Master List has been evaluated by the score of the

results provided by the Web search engines rather than the end user's evaluation. The 95% Confidence Interval shows that the results are dispersed where the quality from the different Web Search Engines are within a similar range.

The results presented by the different Web search engines are retrieved from the same Web sites; however the order in which these are shown to the user constantly differs between them due to their own ranking algorithms. We conclude that there is not common approach between the different Web search engines regarding how to measure relevance when ordering results based on the same high level query. Although the Quality of our proposed Intelligent Search Assistant is not high, this should not be deemed it does not accurately measure result relevance as we have not included the final user's feedback on this evaluation.

7.5 Metasearch Evaluation

To evaluate our proposed ISA with current Metasearch engines we retrieve the results from the Web search engines they use to generate the Result Master List and then compare the results provided by the Metasearch engines against this Result Master List. This proposed method has the inconvenience that we are not considering any result obtained from Internet Web directories neither Online databases from where Metasearch engines may have retrieved some results displayed.

Let's consider each of the N different Metasearch engines uses X different Web search engines; some of them, W , may be common; We select a high level query, $R_t(n(t))$, and then search for the different options by using $N \times X - W$ different Web search engines to retrieve the first Y results from each one. We have scored Y points to the Web site result that is displayed in the top position and 1 point to the Web site result that is shown in the last position. If Web search engines provide different results from the same Web site on several positions we cluster them by choosing only the higher ranking one and discarding, or scoring 0, the others in order to penalize result jamming. We can then generate the master result list by adding the score value of each Web site result value retrieved from the X different Web search engines. We have therefore created a Result Master List with $(N \times X - W) \times Y$ elements where the most relevant result will have the greatest score.

7.5.1 Experimental Results

We have selected both Ixquick and Metacrawler as the Metasearch engines we compare our ISA and its algorithm. After analysing the main characteristics of both Metasearch engines we consider Metacrawler uses (Google Yahoo and Yandex) and Ixquick uses (Google Yahoo and Bing) as their main source of search results.

Our ISA acquires 10 different high level queries based on the travel industry from a user. ISA then retrieves the first 30 results from each of the main Web search engine driver programmed (Google, Yahoo, Bing, and Yandex); we have therefore scored 30 points to the Web site result that is displayed in the top position, 1 point to the Web site result that is shown in the last position and 0 points to each of the result that belongs to the same Web site and it is shown more than once. After we have scored the 120 results provided by the 4 different Web search engines, we combine them by adding the scores of the results which have the same Web site and rank them to generate the Result Master List. We have done this evaluation exercise for each high level query. We then retrieve the first 30 results from Metacrawler and Ixquick and benchmark them against the Result Master List using the proposed Quality formula.

The normalized values of Web search quality, r_s , with its associated Standard Deviation σ and its associated 95% Confidence Range from each of the Web search selected including ISA for the different high level queries are shown on Table 5. We have noticed Bing and Yahoo have provided exactly the same results on the same order to several high level queries mainly because Yahoo search is powered by Bing; therefore on these situations we have not included the score of the results provided by Yahoo in our Result Master List.

Table 5: Metasearch evaluation

Query Q1: Flight New York holiday summer 2013 £500					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
0.4647	0.2454	-0.0576	-0.0374	-0.2894	0.3268
Query Q2: Hotel New York holiday summer 2013 £500					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
0.0245	0.2189	0.2269	-0.1778	-0.5471	0.1159
Query Q3: Car rental New York holiday summer 2013 \$200					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
-0.0759	-0.3426	-0.5092	-0.1146	-0.4529	-0.3662
Query Q4: Vietnamese restaurants New York					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
0.3758	0.2060	-0.2481	0.2801	-0.2670	0.3504
Query Q5: Modern art galleries New York					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA

0.1364	0.2145	-0.1070	0.1631	0.3212	0.2238
Query Q6: Flight Tokio holiday summer 2013 £500					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
-0.5281	0.0498	-0.0436	-0.5166	-0.2414	0.1860
Query Q7: Flight Berlin holiday summer 2013 £50					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
0.1244	-0.0218	-0.0042	0.0398	-0.6085	-0.1112
Query Q8: South East Asia holiday travel pack					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
-0.3617	0.3444	-0.2992	-0.0174	-0.0547	0.3184
Query Q9: London attractions summer					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
0.1406	0.2659	0.3273	-0.0821	-0.3657	0.3112
Query Q10: London music festivals					
Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
0.4367	0.2981	-0.0002	-0.0082	-0.1273	0.1624

with average values:

Metric	Google	Yahoo	Yandex	Metacrawler	Ixquick	ISA
r_s	0.0737	0.1479	-0.0715	-0.0471	-0.2633	0.1518
σ	0.3272	0.2045	0.2437	0.2120	0.2689	0.2282
95%CR	0.2028	0.1268	0.1510	0.1314	0.1667	0.1414

Figure 36 shows the r_s with its associated Standard Deviation σ and the 95% Confidence Interval that corresponds to $r_s \pm 95\%CR$ for the different Metasearch Engines, including our ISA.

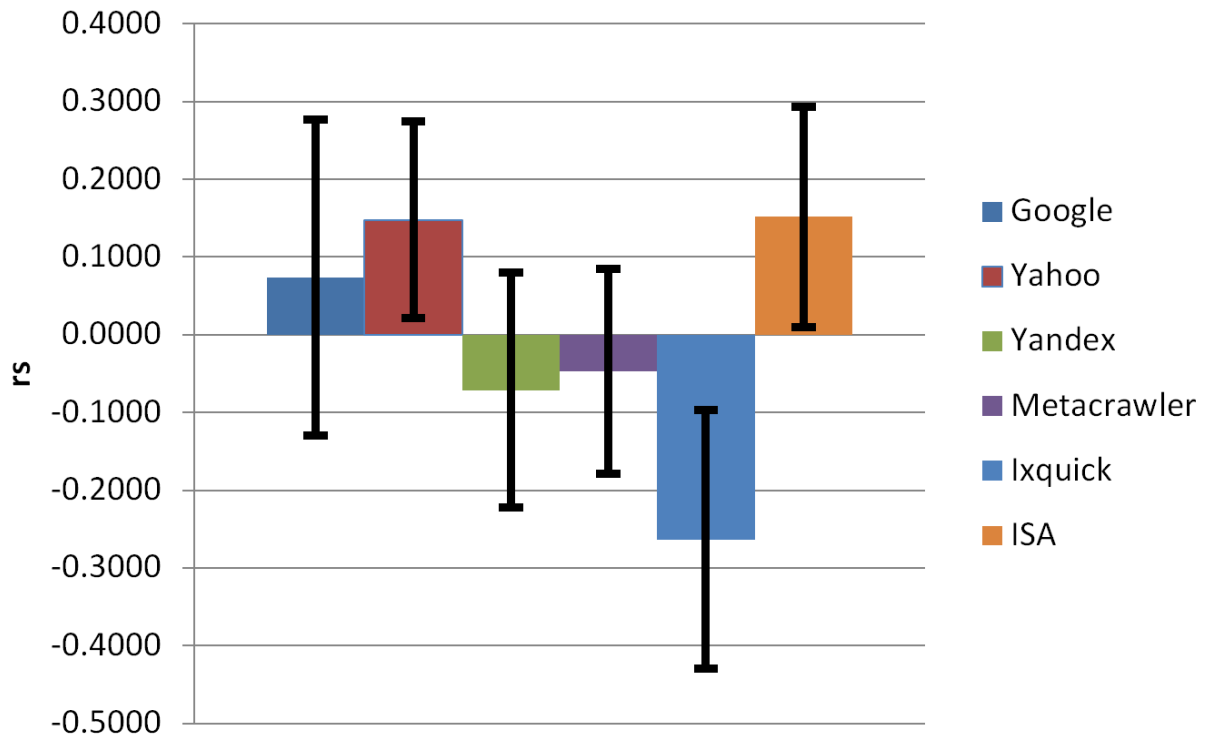


Figure 36: Metasearch Evaluation – Average Values

We were expecting Metasearch engines to score better than Web search engines due to their increased Web coverage (Fig. 36). Similar to the previous validation; the 95% Confidence Interval shows that the results are dispersed and the Quality figures from the different Metasearch Engines are within a similar range.

We can appreciate that Yahoo scores better than other Web Search Engines due to its strong correlation with Bing where the Result Master List is calculated using both Web Search Engines. Our proposed algorithm has scored better than both Metasearch engines proposed; however Metacrawler and Ixquick may have retrieved valid results from other sources (Online databases and Web directories) that we have not included in our Result Master List.

8 User Evaluation – First Iteration

This section evaluates the initial result cost function and the increase of Web Search performance after the first user iteration where results are reordered according to the minimum distance to Relevant Center Point of the selected relevant results.

The evaluation assesses ISA against Google search with a fixed query and ISA against other Web Search Engines (Google, Yahoo, Ask Lycos and Bing) with open queries using a new Quality metric that combines both relevance and rank.

In our evaluation we have asked users to select relevant results, not to rank them, as they normally do when using a Web search engine therefore we consider a result is either relevant or irrelevant. Validators are 15 personal friends from Imperial College students, researchers and London young professionals degree educated.

We have used available Web Search Engines in the current market. Although Yahoo search is powered by Bing and Ask.com has outsourced their search engine to an unknown third party; we have considered them independent and uncorrelated as they run on different Web pages; unless they provide the same results on the same order.

8.1 Implementation

We have implemented the Intelligent Search Assistant in a server to provide direct access to external users. The main index page is programed using JSP and HTML calling to different Java servlets while interacting iteratively with the user (Fig. 37). The main platform used is Java Netbeans and the interaction with the Internet is through Selenium Web-driver. The chosen server for this application is Apache Tomcat due its designed functionality to support Java.

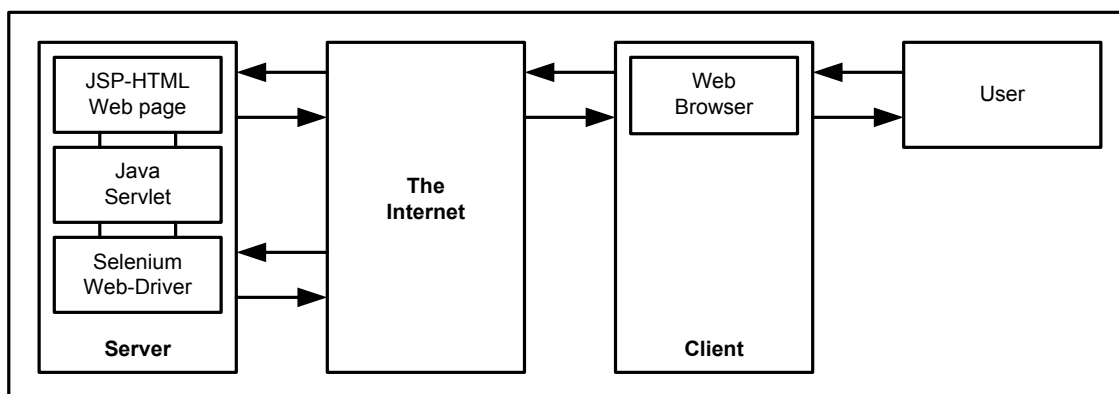


Figure 37: ISA Server implementation

The Intelligent Search Assistant we have proposed emulates how Web search and Metasearch engines work by using a very similar interface to introduce and display information. The Intelligent Search Assistant acquires up to ten different dimensions values (Fig. 38) from the user along with the options of which Web search engine to use and the number of results to be retrieved.

Figure 38: ISA Server interface

Our ISA has been programmed to retrieve snippets from the Web search or Metasearch engines (Google, Yahoo, Ask, Lycos, Bing, Ixquick, Metacrawler and Yandex) the user has selected. Our process is transparent; ISA gets a query from the user and sends it to the search engines selected without altering it. Once results have been retrieved from different sources, our ISA applies the same cost function to all of them to calculate their relevance. If results from different search engines correspond to the same Web page, we only keep the one with the greater value and we eliminate results that are retrieved from Youtube unless the word 'Youtube' is specifically on the user query. We finally reorder the results according to their relevance value.

The Intelligent Search Assistant provides to the user with a reordered list of results re-ranked based on a predetermined cost function using the Random Neural Network; the user then ticks the relevant results and then ISA can either provide with the links or continue the search iteration process. We do not consider the order in which the user ticks the results.

8.2 Quality Metric

In order to measure search quality we can affirm a better search engine provides with a list of more relevant results on top positions. We propose the following quality description where within a list of N results we score N to the first result and 1 to the last result, the value of the quality proposed is then the summation of the position score based of each of the selected results. Our definition of Quality, Q , can be defined as:

$$Q = \sum_{i=1}^Y RSE_i \quad (1)$$

where RSE_i is the rank of the result i in a particular search engine with a value of N if the result is in the first position and 1 if the result is the last one. Y is the total number of results selected by the user. The best Web search engine would have the largest Quality value. We define normalized quality, $\bar{Q}$, as the division of the quality, Q , by the optimum value which it is when the user considers relevant all the results provided by the particular search engine. On this situation Y and N have the same value:

$$\bar{Q} = \frac{Q}{\frac{N(N+1)}{2}} \quad (2)$$

We define I as the quality improvement between a Web search engine and a reference:

$$I = \frac{QW - QR}{QR} \quad (3)$$

where I is the Improvement, QW is the quality of the Web search engine and QR is the quality reference; we use the Quality of Google as QR in our evaluation exercise.

8.1 Google Search – Result Cost Function

We have asked validators to search for different queries using only Google; ISA then provides with a set of reordered results from which the user needs to select the relevant results. We present the results of on Table 6 where we show the average normalized quality of Google and ISA with the improvement of the average quality of our algorithm

against Google for 36 Queries. In addition; the Standard Deviation σ , the variance σ^2 and the p value between the quality figures with different variance and a two tailed distribution are also shown to represent the statistical significance.

Table 6: Google Search – Result Cost Function

Average values: 36 Queries – Student T-Test Value: 0.5984			
Metric	Google	Quality ISA	Improvement ISA
Q	0.4105	0.4430	7.92%
σ	0.2742	0.2467	
CR 95%	0.0896	0.0806	
σ^2	0.0752	0.0609	
p value	0.5984		

Figure 39 shows the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$ for Google and ISA.

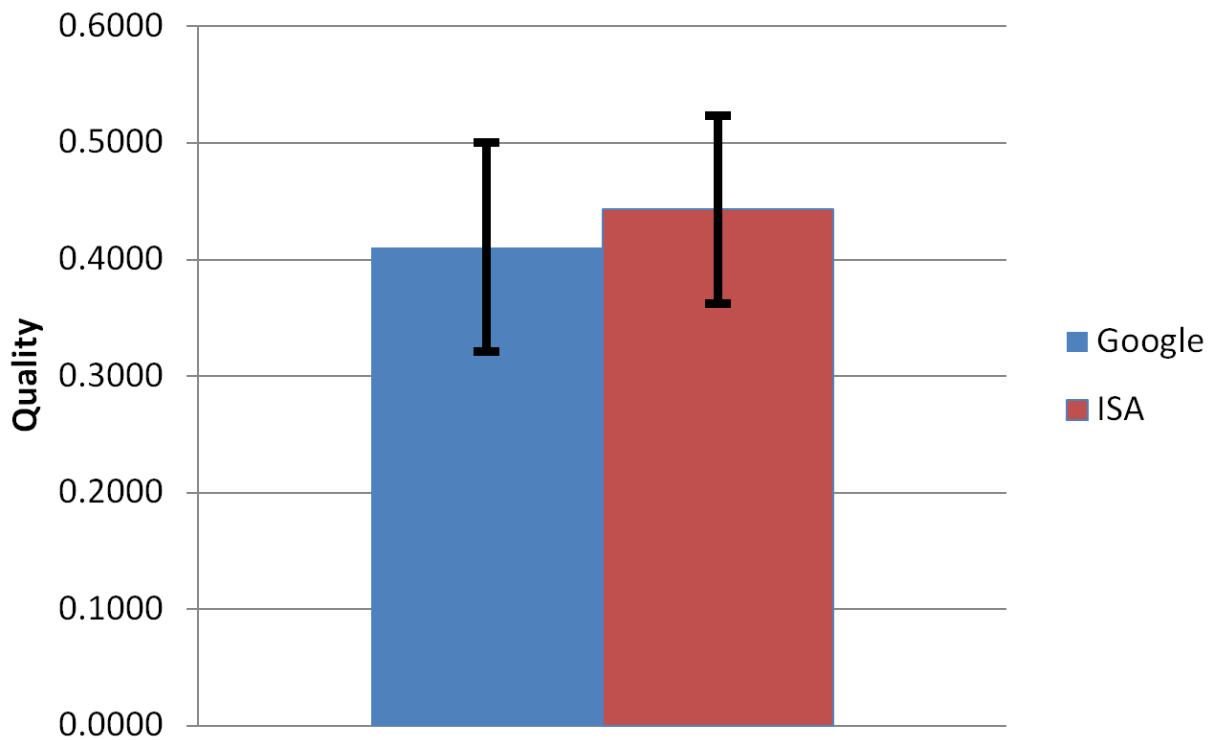


Figure 39: Google Search – Result Cost Function

On this experiment, our ISA performs slightly better than Google (Fig. 39) with an improvement of the average quality of 7.92%. The 95% Confidence Interval shows that the results are dispersed where the quality between Google and ISA are on a similar range. The p value shows that the results are not statically significant for a significance level of $\alpha=0.05$.

8.2 Web Search – Result Cost Function

Validators can select from which Web search engine they would their results to be retrieved from where the users need to select the relevant results. We do not show from which Web search engine each result has been retrieved.

We show the values on Table 7 where we present the normalized quality of each Web search engine selected including our ISA for 27 queries; because users can choose any Web search engine; we are not introducing the improvement value or p value as we do not have a unique Web search engine reference.

Table 7: Web Search – Result Cost Function

Average values: 27 Queries						
Metric	Google	Yahoo	Ask	Lycos	Bing	ISA
Q	0.2920	0.2469	0.3805	0.4096	0.3485	0.4382
σ	0.0147	0.0122	0.0372	0.0228	0.0125	0.0104
CR 95%	0.0070	0.0058	0.0243	0.0141	0.0056	0.0039

Figure 40 shows the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$ for the different Web Search Engines, including ISA.

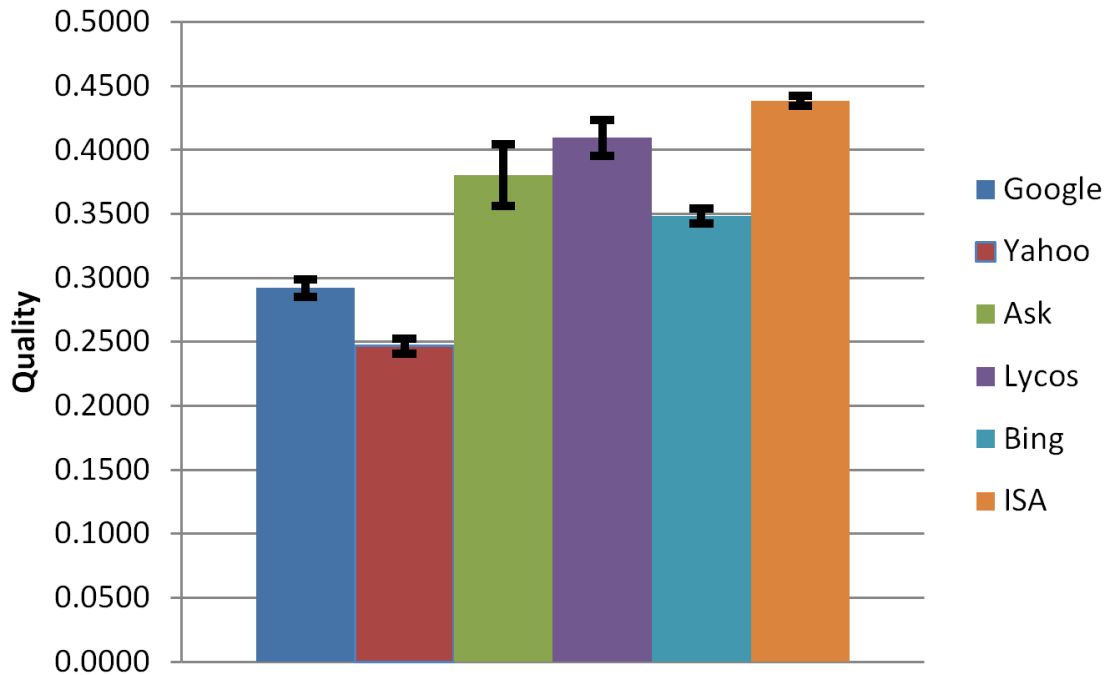


Figure 40: Web Search – Result Cost Function

Ask and Lycos performs better than the other commercial Web search engines (Fig. 40) although on average, our ISA performs better. The 95% Confidence Interval shows that the results are more concentrated with a better precision because users were able to select their best preferred Web Search Engines rather than the Google fixed option.

8.3 Google Search - Fixed Query – Relevant Centre Point

We have run our ISA with the high level query “**Flight New York Holiday 2013 £500**” to be searched on Google retrieving the first 100 results; this query contains the three different dimension types we have defined. We have asked validators to choose only the three best results ones for simplicity from the 10 result set our ISA has selected as more relevant. With these selections we have calculated the Relevant Centre Point and reorder the results according to the minimum distance to this point.

The value of the Quality for ISA and Google is shown on Table 8 where it is represented the Improvement of Quality after reordering results to the Relevant Centre Point.

Table 8: Google Search - Fixed Query – Relevant Center Point

Query: Flight New York Holiday 2013 £500					
Metric	Google	ISA	Improvement ISA	ISA Circle	Improvement ISA Circle
Q	0.3273	0.4182	27.78%	0.4545	8.69%

Figure 41 shows the Quality of Google ISA and ISA Relevant Center Point after the first user iteration.

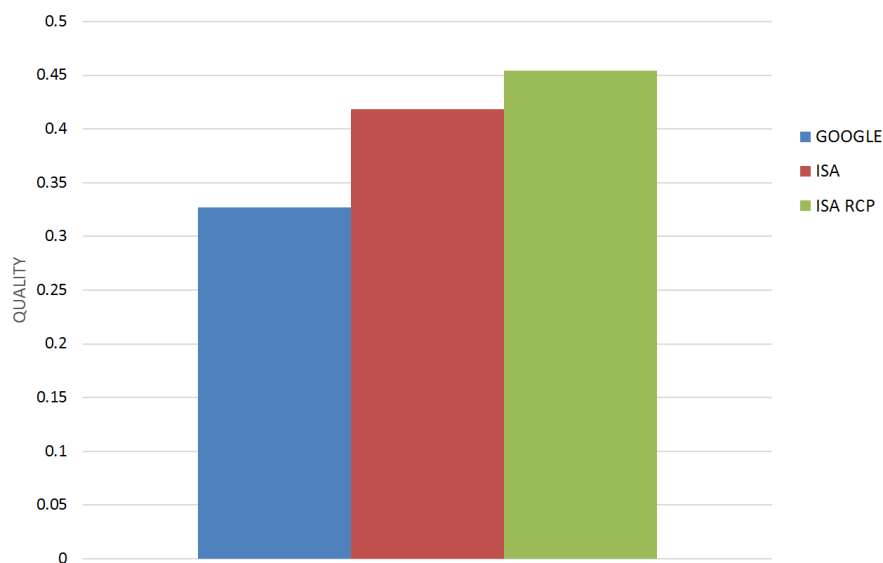


Figure 41: Google Search - Fixed Query – Relevant Center Point

After the first user iteration with the user, where results are reordered according to the minimum distance to the Relevant Center Point, our Quality figure has improved (Fig. 41).

8.4 Google Search – Open Query - Relevant Centre Point

Users can select only Google and ISA provides with a reordered list from where the user needs to select which results are relevant. ISA reorders the results using the dimension relevant centre point providing to the user with another reordered result list from where the users need to select again the relevant ones.

We present the results on Table 9 where we show the normalized quality of Google, ISA and ISA with the relevant circle iteration including the average improvement against Google in both situations for 21 queries. In addition; the Standard Deviation σ , the variance σ^2 and the p value between the quality figures at different stages with different variance and a two tailed distribution are also shown to represent the statistical significance.

Table 9: Google Search – Open Query - Relevant Centre Point

Average values: 21 Queries					
Metric	Google	ISA	Improvement ISA	ISA Circle	Improvement ISA Circle
Q	0.4537	0.4645	2.38%	0.4984	9.86%
σ	0.2835	0.2847		0.2970	
CR 95%	0.1212	0.1218		0.1270	
σ^2	0.0804	0.0811		0.0882	
p value	0.9027			0.7071	

Figure 42 shows the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$ for Google, ISA and ISA Relevant Centre Point after the first user iteration.

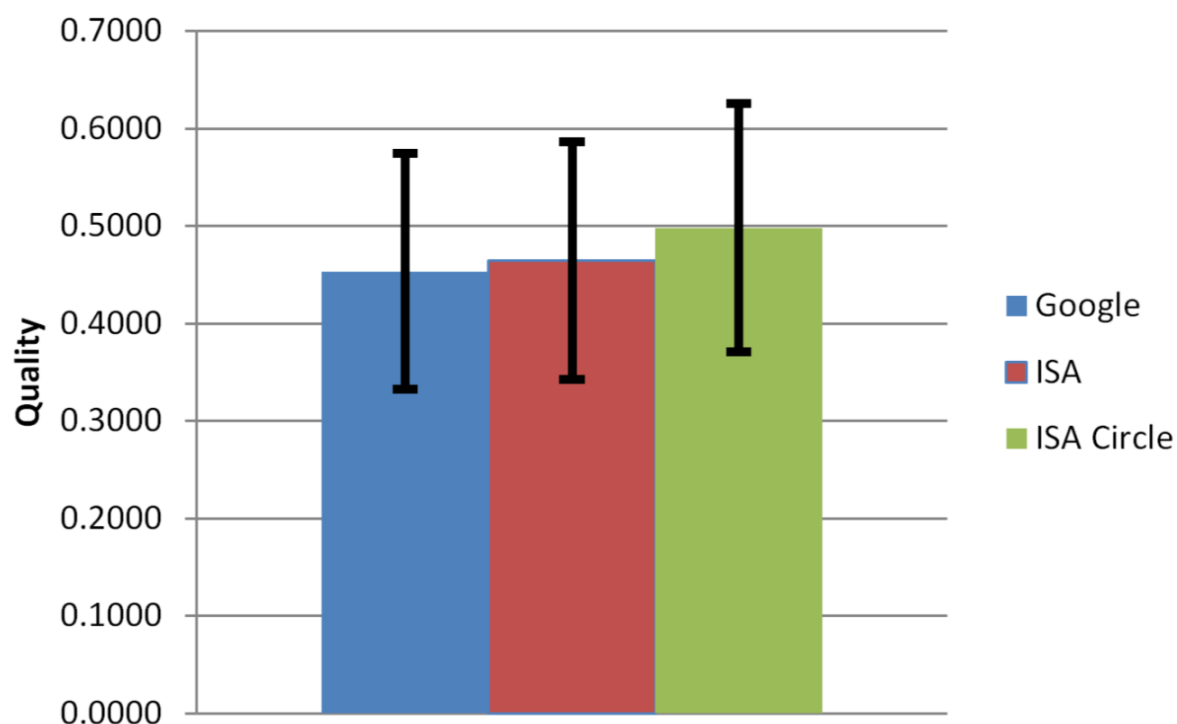


Figure 42: Google Search – Open Query - Relevant Center Point

When our ISA reorders the results according to the relevant circle with the introduction of the feedback from the user (Fig. 42), its performance has increased with this extra iteration. The 95% Confidence Interval shows that the results are dispersed where the quality between Google and ISA are on a similar range although Quality improves after the first user iteration. The p value for both iteration shows that the results are not statically significant for a significance level of $\alpha=0.05$.

Our Intelligent Search Assistant performs slightly better than Google with an average improvement, however, this evaluation may be biased because users tend to concentrate on the first results provided which were the ones we showed in our algorithm. ISA performs better than the average of other Web search engines, although ISA provides with all the relevant results due its Metasearch nature this advantage is compensated with the fact that ISA also provides also with the entire set of irrelevant results from the Web search engines selected. We notice that ISA adapts and learns from the user's previous relevance measurements; it increases significantly its quality and improvement within the first iteration.

9 User Evaluation – Learning algorithms

This section evaluates the increase of Web Search performance due to the independent introduction of two Learning Algorithms: Reinforcement Learning and Gradient Descent that learn the perceived user relevance on an iterative process.

The evaluation assesses ISA against Web Search Engines (Google or Bing) with open queries, Academic Databases (Google Scholar, IEEE Xplore, CiteseerX or Microsoft Academic) and Recommender Systems (GroupLens film, Trip Advisor and Amazon) with new Quality metric that combines both relevance and rank.

In our evaluation we have asked users to select relevant results, not to rank them, as they normally do when using a Web search engine therefore we consider a result is either relevant or irrelevant. Validators are 15 personal friends from Imperial College students, researchers and London young professionals degree educated.

We have used available Web Search Engines in the current market. Although Yahoo search is powered by Bing and Ask.com has outsourced their search engine to an unknown third party; we have considered them independent and uncorrelated as they run on different Web pages; unless they provide the same results on the same order.

9.1 Quality Metric

In order to measure search quality we can affirm a better search engine provides with a list of more relevant results on top positions. We propose the following quality description where within a list of N results we score N to the first result and 1 to the last result, the value of the quality proposed is then the summation of the position score based of each of the selected results. Our definition of Quality, Q, can be defined as:

$$Q = \sum_{i=1}^Y RSE_i \quad (1)$$

where RSE_i is the rank of the result i in a particular search engine with a value of N if the result is in the first position and 1 if the result is the last one. Y is the total number of results selected by the user. The best Web search engine would have the largest Quality value. We define normalized quality, $\bar{Q}$, as the division of the quality, Q, by the optimum

value which it is when the user considers relevant all the results provided by the particular search engine. On this situation Y and N have the same value:

$$\bar{Q} = \frac{Q}{\frac{N(N+1)}{2}} \quad (2)$$

We define I as the quality improvement between a Web search engine and a reference:

$$I = \frac{QW - QR}{QR} \quad (3)$$

Where I is the Improvement, QW is the quality of the Web Search Engine and QR is the quality reference; we use the Quality of Google or Bing as QR in our evaluation exercise in the first iteration and on further iterations, we use as Quality reference the value of the previous iteration.

9.2 Web Search Evaluation

The Intelligent Search Assistant we have proposed emulates how Web search engines work by using a very similar interface to introduce and display information. Users in the experiments can choose which Web search engine would like to use and the learning type. Our ISA then collects the first 50 results from the search engine selected, reorders them according to its cost function and finally show to the user the first 20 results. We consider 50 results is a good approximation of search deep as more results can add clutter and irrelevance; 20 results is the average number of results read by a user before he launches another search if he does not find any relevant one. ISA reorders results while learning on the 2 step iterative process showing only the best 20 results to the user. There are no rules in what users can search, however they have been advised their queries may be published.

9.2.1 Implementation

We have implemented the Intelligent Search Assistant in a server with external access (Fig. 43). The Intelligent Search Assistant acquires up to eight different dimensions values from the user along with the options of which Web search engine to use and the type of learning to be implemented.

Intelligent Search Assistant

Dimension 1
Dimension 2
Dimension 3
Dimension 4
Dimension 8

☐ Google ☐ Bing

Learning: Gradient Descent
Reinforcement Learning

Search!

Figure 43: ISA Web Search interface

The Intelligent Search Assistant provides to the user with a reordered list of results re-ranked based on a predetermined cost function using the Random Neural Network; the user then ticks the relevant results and ISA can either provide with the links or continue the search iteration process. We do not consider the order in which the user ticks the results.

9.2.2 Gradient Descent

We present on Table 10 the Gradient Decent average Quality results of ISA and Web Search Engines with its associated 95% Confidence Range for the 17 different queries. The first I represents the improvement from ISA against the Web Search Engine; the second I is between ISA iterations 2 and 1 and finally the third I is between ISA iterations 3 and 2. In addition; the Standard Deviation σ , the variance σ^2 and the p value between the quality figures at different stages with different variance and a two tailed distribution are also shown to represent the statistical significance.

Table 10: Web Search evaluation – Gradient Descent - Average Values

Average Values – Gradient Descent – 17 queries						
Search	Iteration	Q	σ	CR 95%	σ^2	p value
Web	1	0.4105	0.2072	0.0985	0.0429	0.0616
ISA	1	0.5840	0.3034	0.1442	0.0921	
I	-	42.24%	-	-	-	
Web	2	0.4527	0.2264	0.1076	0.0513	0.0677
ISA	2	0.6121	0.2633	0.1252	0.0693	
I	-	4.83%	-	-	-	
Web	3	0.4649	0.2197	0.1045	0.0483	0.0637
ISA	3	0.6203	0.2507	0.1192	0.0629	
I	-	1.33%	-	-	-	

Figure 44 shows the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$ for the three iterations.

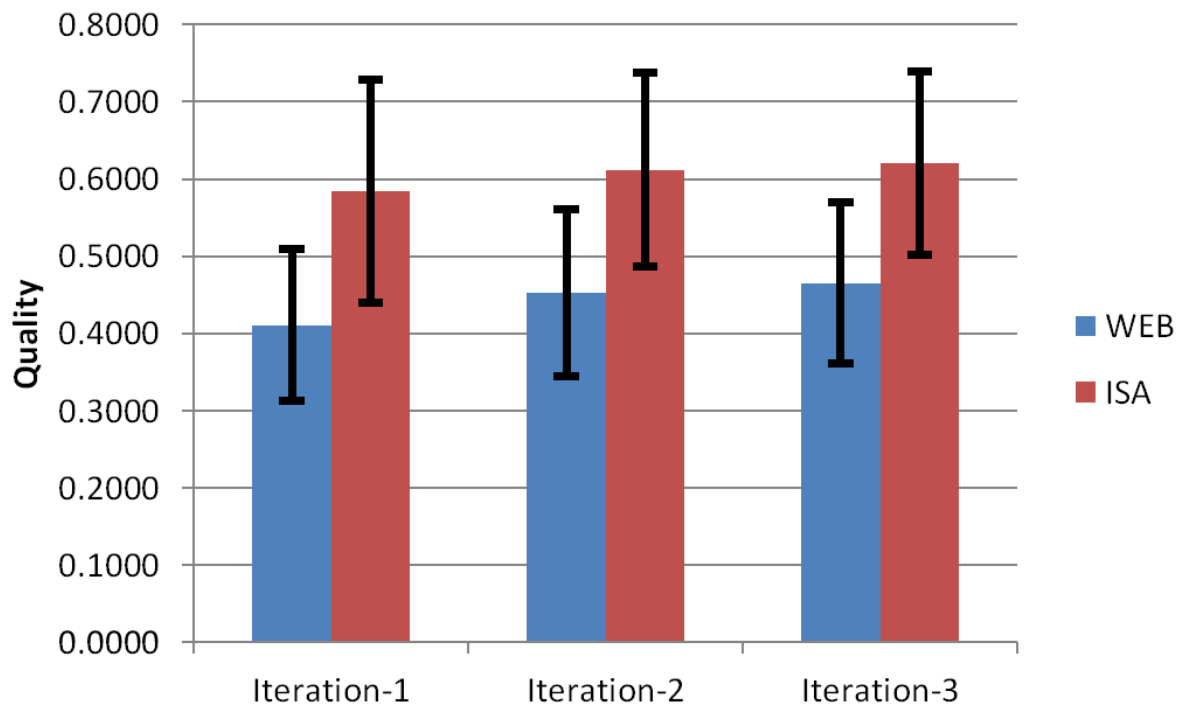


Figure 44: Web Search evaluation – Gradient Descent - Average Values

Figure 45 shows the associated improvement for the three iterations.

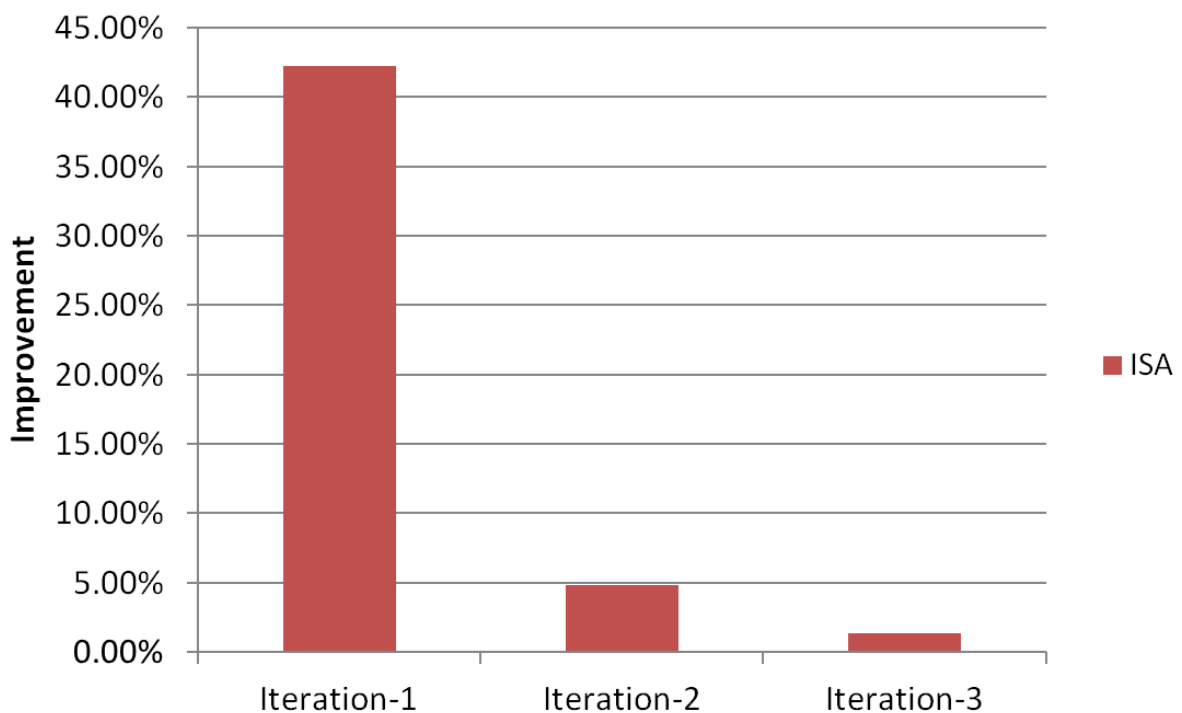


Figure 45: Web Search evaluation – Gradient Descent - Improvement

There is an increment in Quality values when Gradient Descent learning is included (Fig. 44). Gradient Descent highly learns on its first iteration gradually decreasing its learning rate on successive iterations (Fig. 45). The 95% Confidence Interval shows that the results overlap in some regions. The p value shows that the results are close to be significant for a significance level of $\alpha=0.05$.

9.2.3 Reinforcement Learning

We present on Table 11 the Reinforcement Learning average Quality results of ISA and Web Search Engines with its associated 95% Confidence Range for the 12 different queries. The first I represents the improvement from ISA against the Web Search Engine; the second I is between ISA iterations 2 and 1 and finally the third I is between ISA iterations 3 and 2 respectively. In addition; the Standard Deviation σ , the variance σ^2 and the p value between the quality figures at different stages with different variance and a two tailed distribution are also shown to represent the statistical significance.

Table 11: Web Search evaluation – Reinforcement Learning - Average Values

Average Values – Reinforcement Learning – 12 queries						
Search	Iteration	Q	σ	CR 95%	σ^2	p value
Web	1	0.4227	0.2527	0.1430	0.0639	0.2315
ISA	1	0.5750	0.3448	0.1951	0.1189	
I		36.02%	-	-	-	
Web	2	0.4726	0.2342	0.1325	0.0549	0.0880
ISA	2	0.6729	0.3085	0.1745	0.0952	
I		17.03%	-	-	-	
Web	3	0.4861	0.2344	0.1326	0.0549	0.1036
ISA	3	0.6823	0.3224	0.1824	0.1039	
I		1.39%	-	-	-	

Figure 46 shows the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$ for the three iterations.

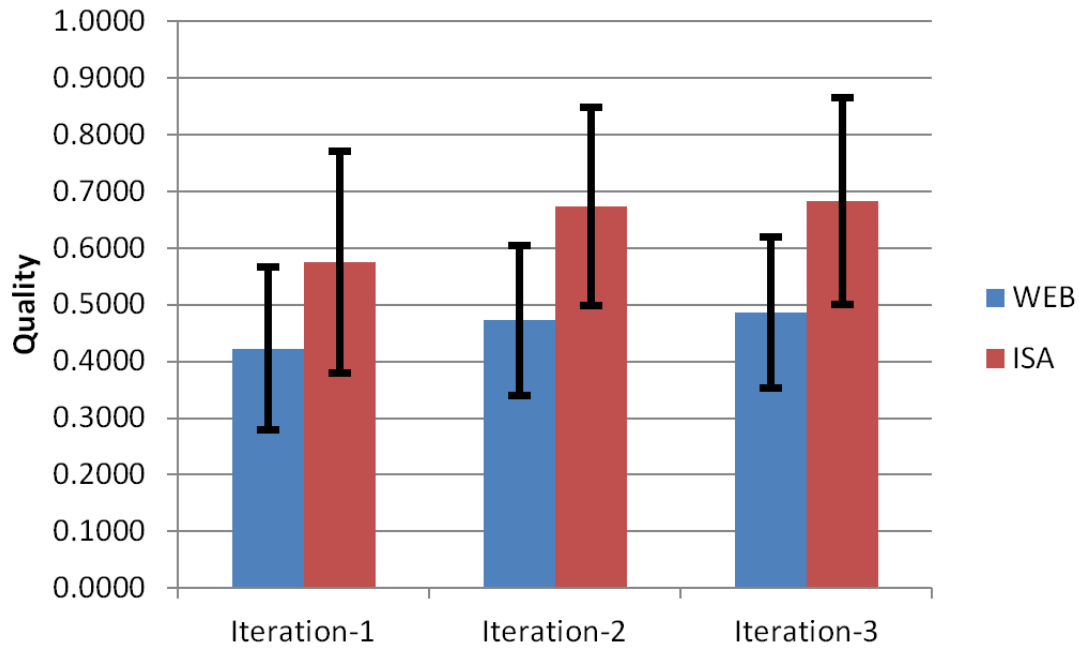


Figure 46: Web Search evaluation – Reinforcement Learning - Average Values

Figure 47 shows the associated improvement for the three iterations.

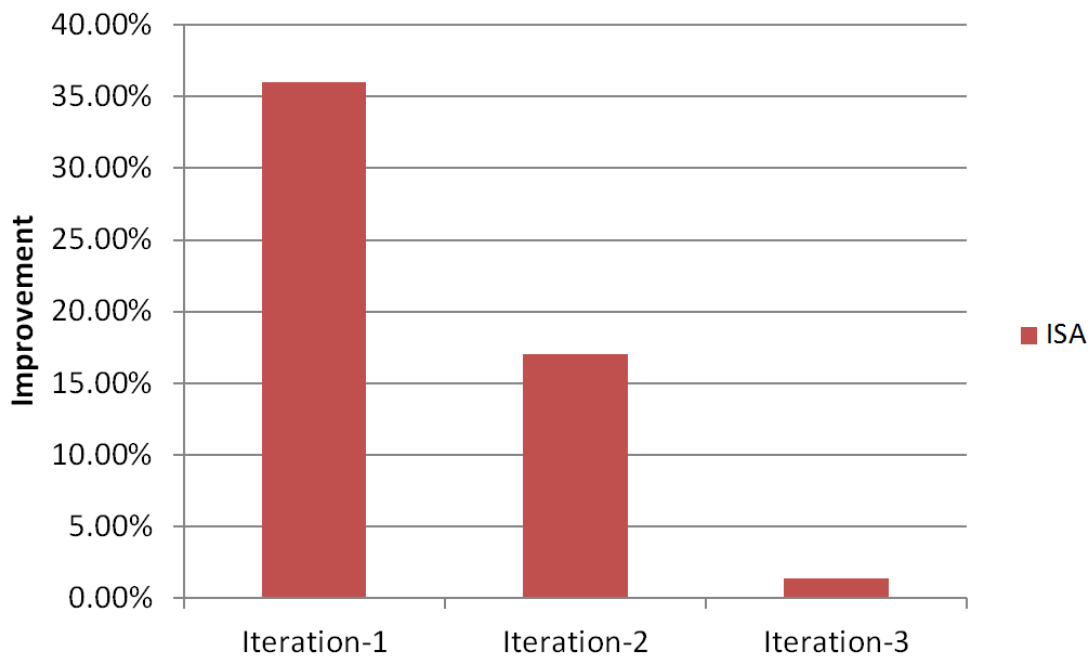


Figure 47: Web Search evaluation – Reinforcement Learning – Improvement

There is an increment in all performance values when Reinforcement Learning is included (Fig. 46). Reinforcement Learning highly learns on its first two iterations with a residual or corrective learning on its latest one (Fig. 47). Similar to our previous evaluation, the 95% Confidence Interval shows that the results overlap in some regions. The p value shows that the results are close to be significant for a significance level of $\alpha=0.05$.

Figure 48 presents an evaluation between Gradient Descent and Reinforcement Learning algorithms with the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

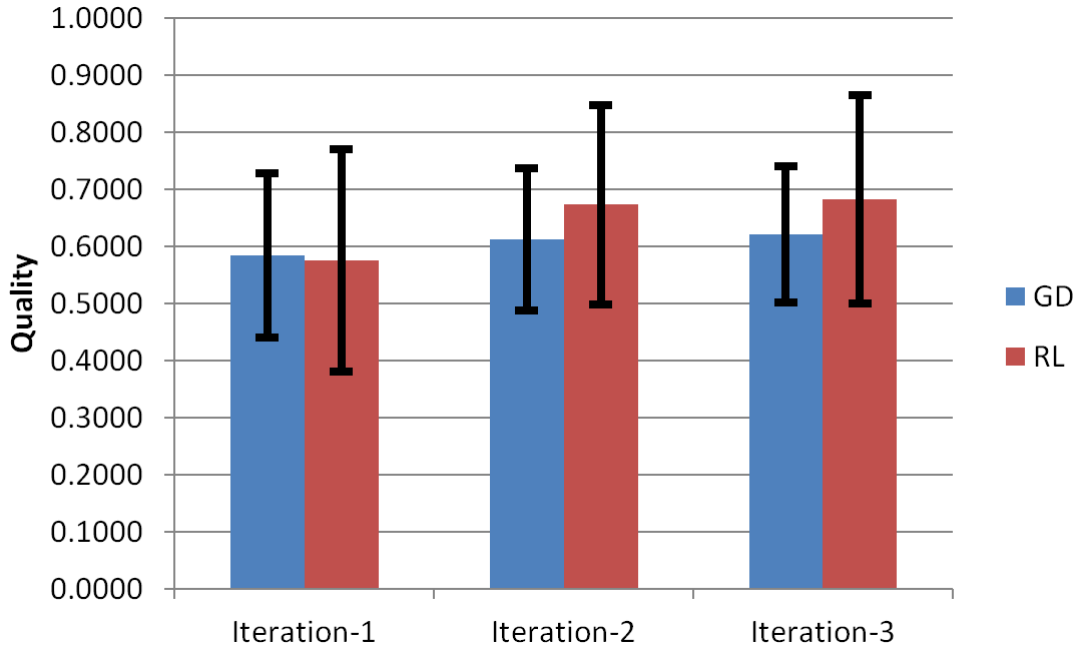


Figure 48: Web Search evaluation – Evaluation between learnings

Reinforcement Learning algorithm performs better than Gradient Descent (Fig. 48). Although Gradient Descent provides a slightly better evaluation on the first iteration; Reinforcement Learning outperforms on the second one due its higher learning rate. The 95% Confidence Interval shows that statically Reinforcement Learning values are more disperse than Gradient Descent; this is due to Gradient Descent statistics results are more significant than the results obtained from Reinforcement Learning.

9.2.4 Evaluation between Relevance Metrics

We have defined different relevance measurements and this research has proposed a new quality metric based on relevance and rank. The following tables and figures show the correlation between them and our proposed quality for the two different learning methods. We define correlation as:

$$\text{Correlation} = \frac{\text{Metric-Q}}{Q} \quad (4)$$

where Q is our Quality metric.

Table 12 covers Gradient Descent Learning for the 17 queries and Table 13 represents Reinforcement Learning for the 12 queries. The first column Table 14 and Table 15 represent MAP (mean average precision), TSAP (TREC style average precision), P (precision), NDCG (Normalized discounted cumulative gain), Q (quality), MRR (medium reciprocal rank) and ERR (expected reciprocal rank) for the first Iteration.

Table 12: Relevance Metric evaluation – Gradient Descent - Average Values

Average Values – Gradient Descent – 17 queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.3582	0.0661	0.5765	0.6635	0.4105	0.5502
ISA	1	0.7145	0.1160	0.5765	0.8339	0.5840	0.8696
Correlation							
Web	1	-12.74%	-83.90%	40.44%	61.63%	0	34.03%
ISA	1	22.35%	-80.14%	-1.28%	42.79%	0	48.90%

Table 13: Relevance Metric evaluation – Reinforcement Learning - Average Values

Average Values– Reinforcement Learning 12 queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.3819	0.0648	0.5750	0.6722	0.4227	0.5042
ISA	1	0.6914	0.1091	0.5750	0.8333	0.5750	0.7827
Correlation							
Web	1	-9.65%	-84.67%	36.03%	59.03%	0	19.28%
ISA	1	20.24%	-81.03%	0.00%	44.92%	0	36.12%

Figure 49 and Figure 50 show their correlation value for Gradient Descent and Reinforcement Learning respectively.

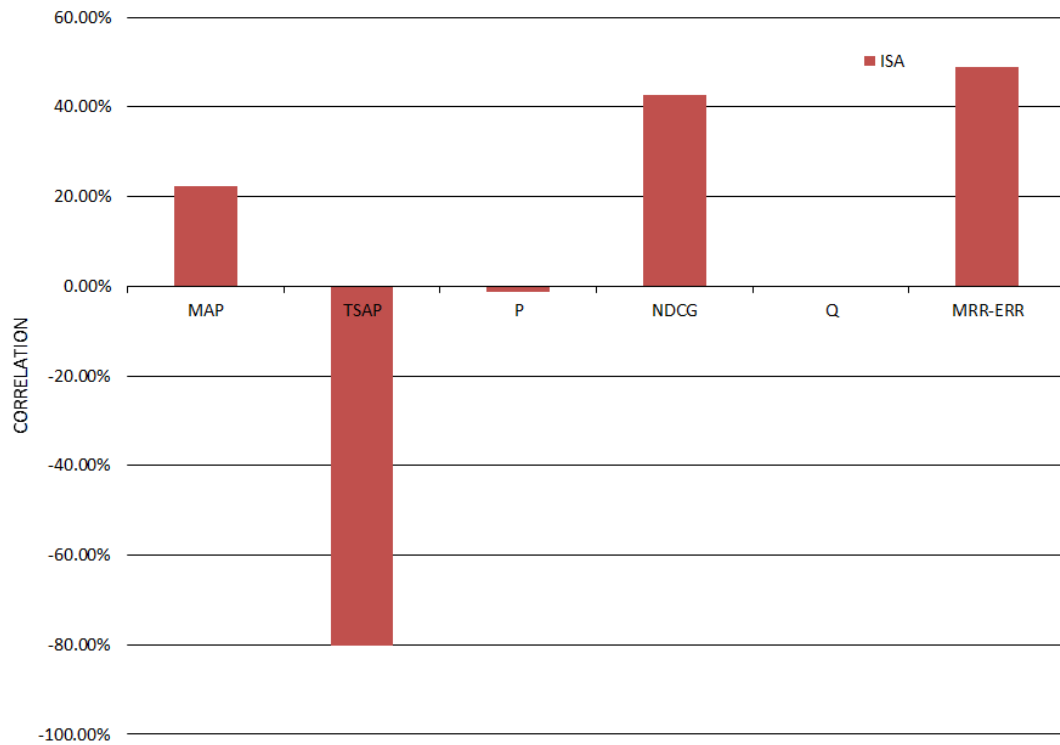


Figure 49: Relevance Metric Evaluation – Gradient Descent

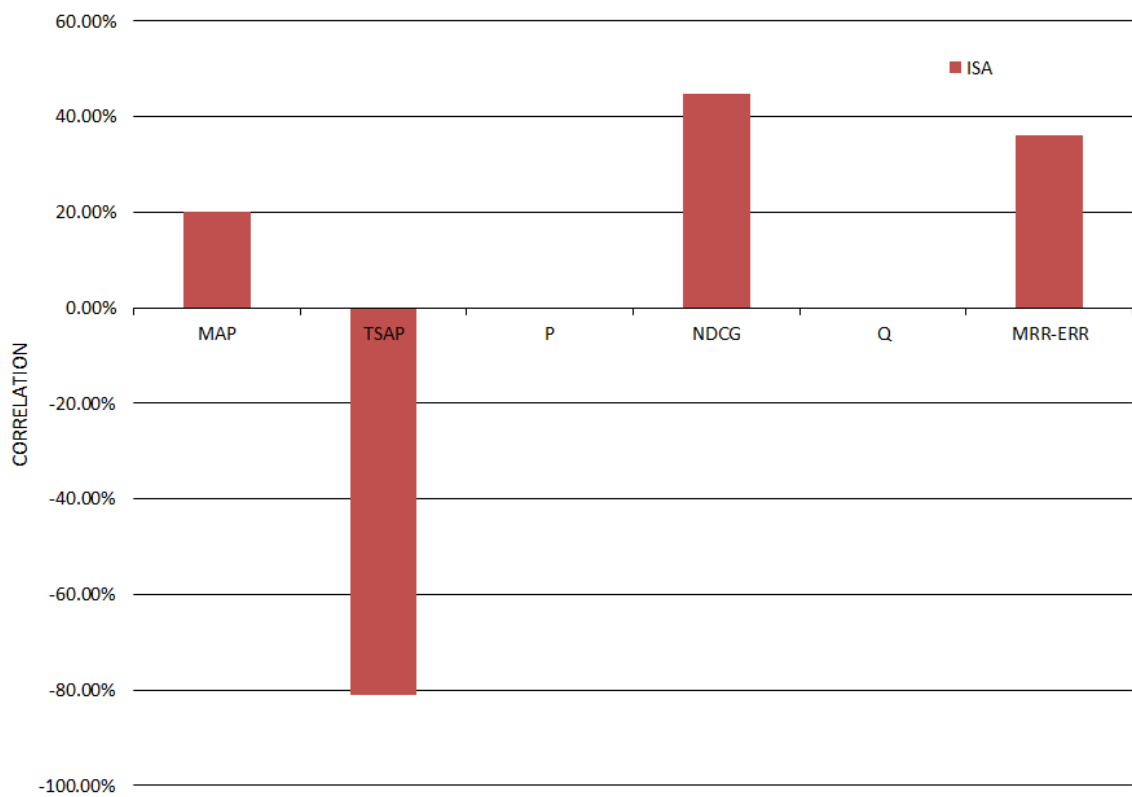


Figure 50: Relevance Metric Evaluation – Reinforcement Learning

Our results show that we cannot determine which metric would have measured better the relevance of the different results in our user queries (Fig. 49, Fig. 50). The metric we have proposed, quality, is consistent with other metrics; it is high correlated with Precision and slightly correlated with Medium Average Precision measurements.

9.3 Academic Database Evaluation

The Intelligent Search Assistant we have proposed emulates how an Online Academic Database works by using a very similar interface to introduce and display information. We have validated our own research bibliography by searching our study topics. Our ISA then collects the first 50 results from the search engine selected, reorders them according to its cost function and finally show to the user the first 20 results. We consider 50 results is a good approximation of search deep as more results can add clutter and irrelevance; 20 results is the average number of results read by a user before he launches another search if he does not find any relevant one. ISA reorders results while learning on the 2 step iterative process showing only the best 20 results to the user.

9.3.1 Implementation

We have implemented the Intelligent Search Assistant in a server with only internal access (Fig. 51). Our Intelligent Search Assistant acquires up to eight different dimensions values from the user along with the options of which Online Academic Database to use (Google Scholar, IEEE Xplore, CiteseerX or Microsoft Academic) and the type of learning to be implemented.

Intelligent Search Assistant

Dimension 1 Dimension 2 Dimension 3 Dimension 4 Dimension 8

☐ Google Scholar ☐ IEEE Xplore ☐ CiteseerX ☐ Microsoft Academic

Learning: Gradient Descent
Reinforcement Learning

Search!

Figure 51: ISA Academic Database interface

The Intelligent Search Assistant provides to the user with a reordered list of results re-ranked based on a predetermined cost function using the Random Neural Network; the user then to tick the relevant results and ISA can either provide with the links or continue the search iteration process.

9.3.2 Gradient Descent

We present on Table 16 the Gradient Decent average Quality results of ISA and the Online Academic Databases with its associated 95% Confidence Range for the 12 different queries. The first I represents the improvement from ISA against the Web Search Engine; the second I is between ISA iterations 2 and 1 and finally the third I is between ISA iterations 3 and 2 In addition; the Standard Deviation σ , the variance σ^2 and the p value between the quality figures at different stages with different variance and a two tailed distribution are also shown to represent the statistical significance.

Table 14: Database evaluation – Gradient Descent - Average Values

Average Values – Gradient Descent – 12 queries						
Search	Iteration	Q	σ	CR 95%	σ^2	p value
Web	1	0.4390	0.1168	0.0661	0.0136	0.0212
ISA	1	0.5642	0.1297	0.0734	0.0168	
I		28.52%	-	-	-	
Web	2	0.4785	0.1099	0.0622	0.0121	0.0045
ISA	2	0.6399	0.1375	0.0778	0.0189	
I		13.42%	-	-	-	
Web	3	0.4959	0.1195	0.0676	0.0143	0.0064
ISA	3	0.6630	0.1492	0.0844	0.0223	
I		3.60%	-	-	-	

Figure 52 shows the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$ for the three iterations. The results presented (Fig. 52, Fig. 53) are consistent with the previous learning evaluation; Gradient Descent has it highest learning rate on its first iteration with a gradual decrease on successive iterations. The 95% Confidence Interval shows that the results do not overlap with statistically significant results for a significance level of $\alpha=0.05$.

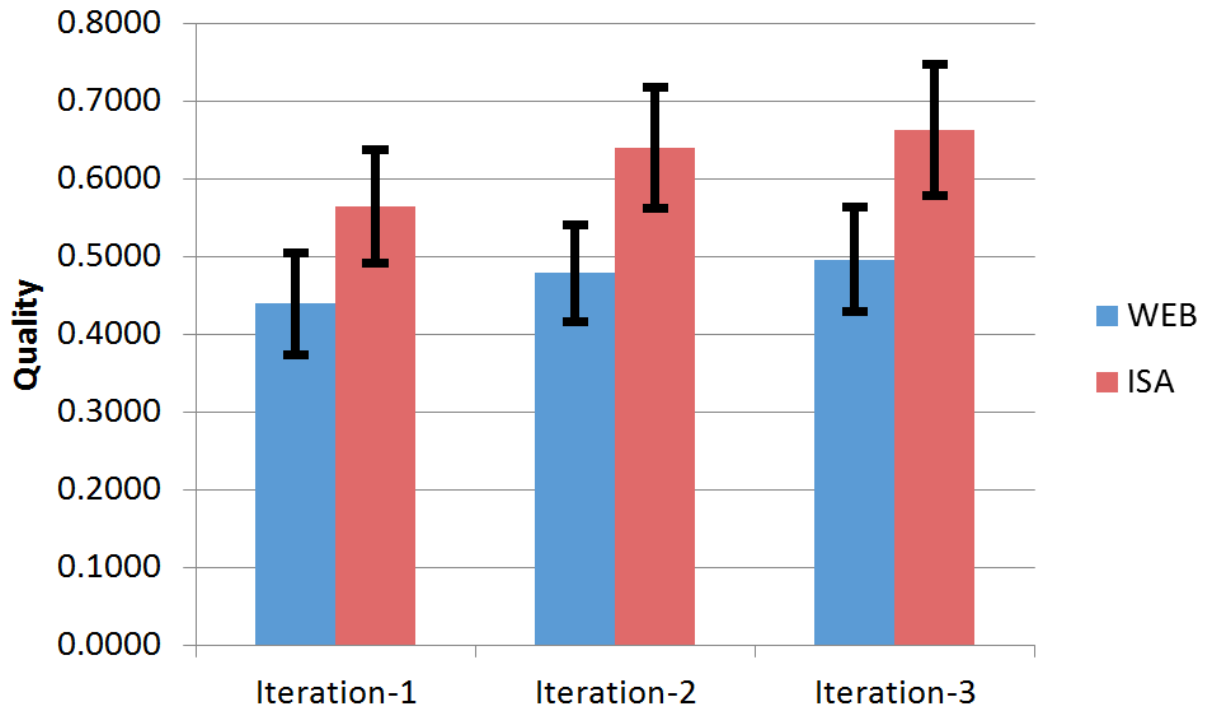


Figure 52: Database evaluation – Gradient Descent - Average Values

Figure 53 shows the associated improvement for the three iterations.

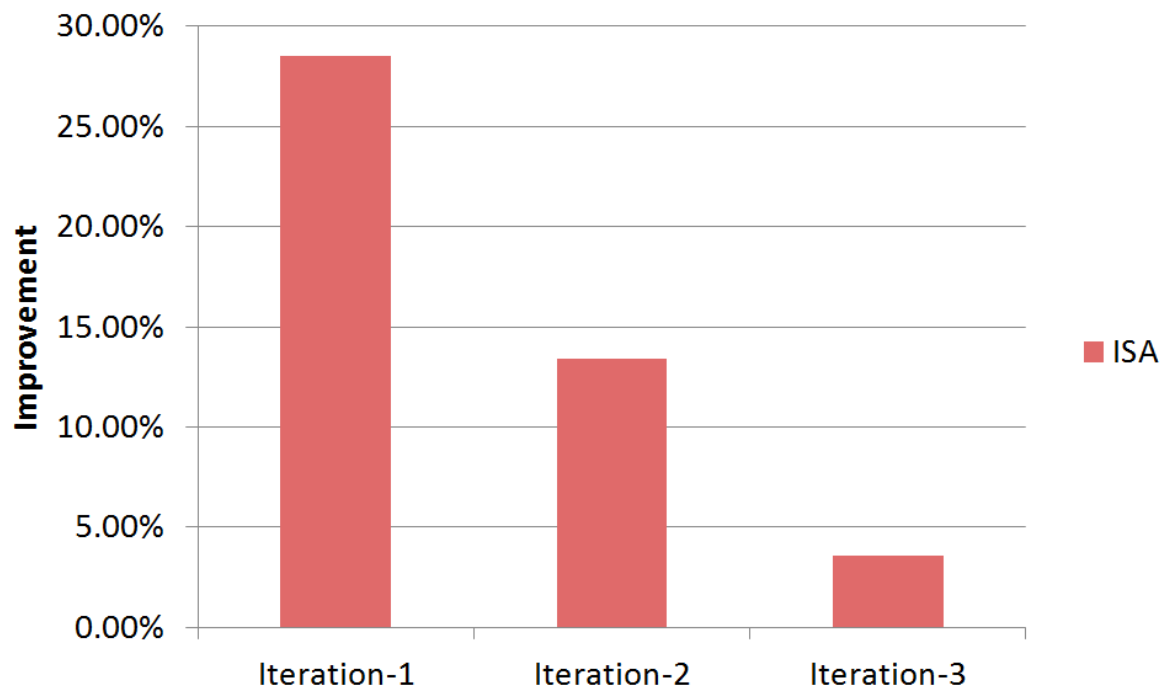


Figure 53: Database evaluation – Gradient Descent – Improvement

9.3.3 Reinforcement Learning

We present on Table 15 the Reinforcement Learning average Quality results of ISA and the Online Academic Databases with its associated 95% Confidence Range for the 12 different queries. The first I represents the improvement from ISA against the Web Search Engine; the second I is between ISA iterations 2 and 1 and finally the third I is between ISA iterations 3 and 2 respectively. In addition; the Standard Deviation σ , the variance σ^2 and the p value between the quality figures at different stages with different variance and a two tailed distribution are also shown to represent the statistical significance.

Table 15: Database evaluation – Reinforcement Learning - Average Values

Average Values – Reinforcement Learning – 12 queries						
Search	Iteration	Q	σ	CR 95%	σ^2	p value
Web	1	0.4127	0.2292	0.1297	0.0525	0.3384
ISA	1	0.5141	0.2761	0.1562	0.0762	
I		24.58%	-	-	-	
Web	2	0.4433	0.2061	0.1166	0.0425	0.1036
ISA	2	0.6124	0.2753	0.1558	0.0758	
I		19.13%	-	-	-	
Web	3	0.4614	0.1890	0.1069	0.0357	0.0649
ISA	3	0.6440	0.2629	0.1487	0.0691	
I		5.16%	-	-	-	

Figure 54 shows the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$ for the three iterations. The results presented are also consistent with the previous learning evaluation (Fig. 54, Fig. 55); Reinforcement Learning maintains a high learning rate on its second iteration with a residual learning rate on its third one. The 95% Confidence Interval shows that the results overlap altogether on this evaluation. The p value shows that the results are not significant for a significance level of $\alpha=0.05$.

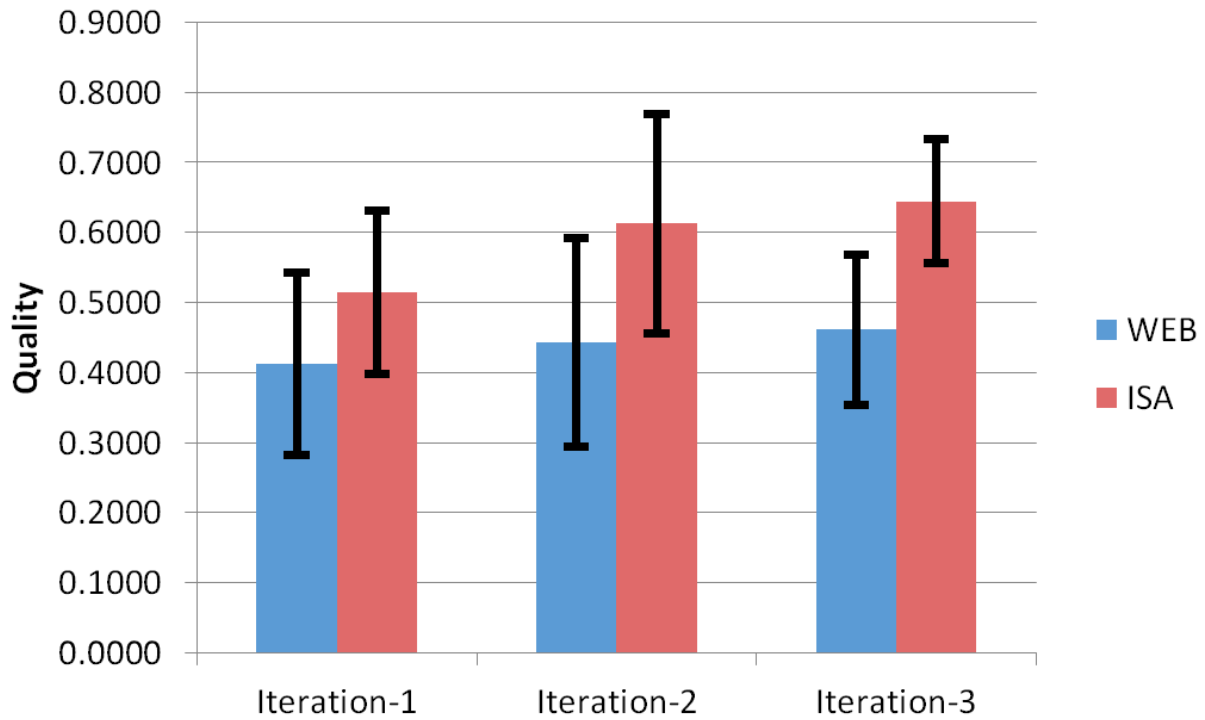


Figure 54: Database evaluation – Reinforcement Learning - Average Values

Figure 55 shows the associated improvement for the three iterations.

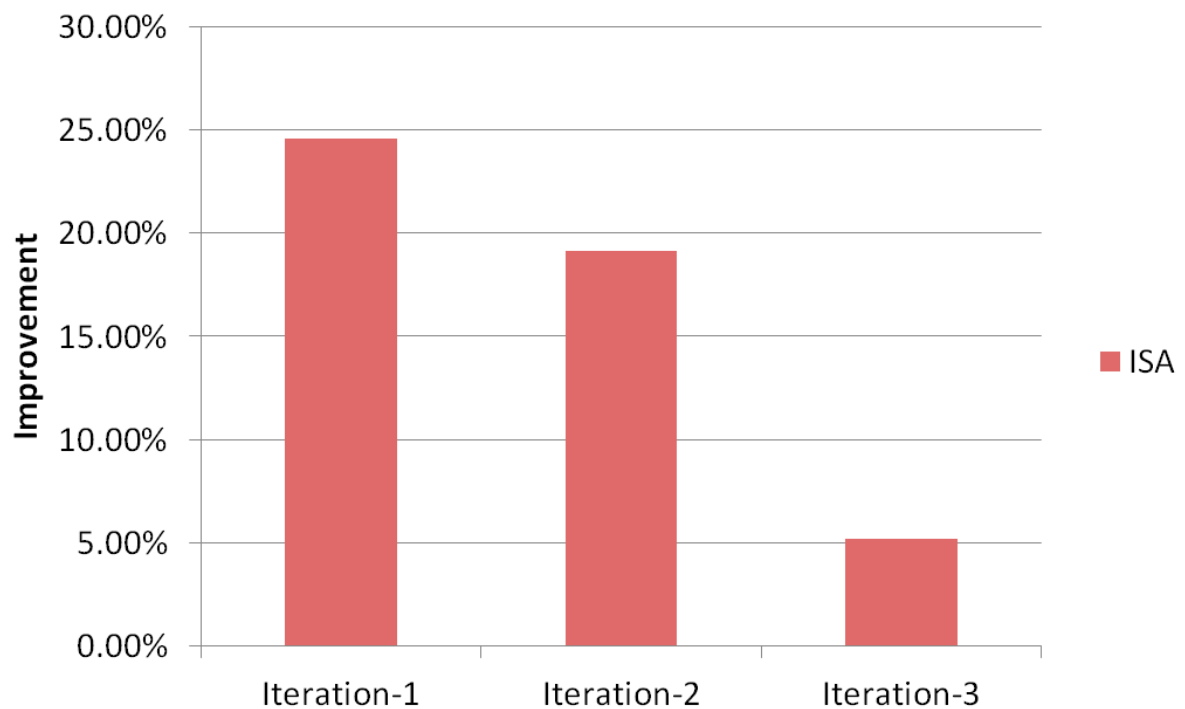


Figure 55: Database evaluation – Reinforcement Learning - Improvement

Figure 56 presents an evaluation between Gradient Descent and Reinforcement Learning algorithms with the Quality and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

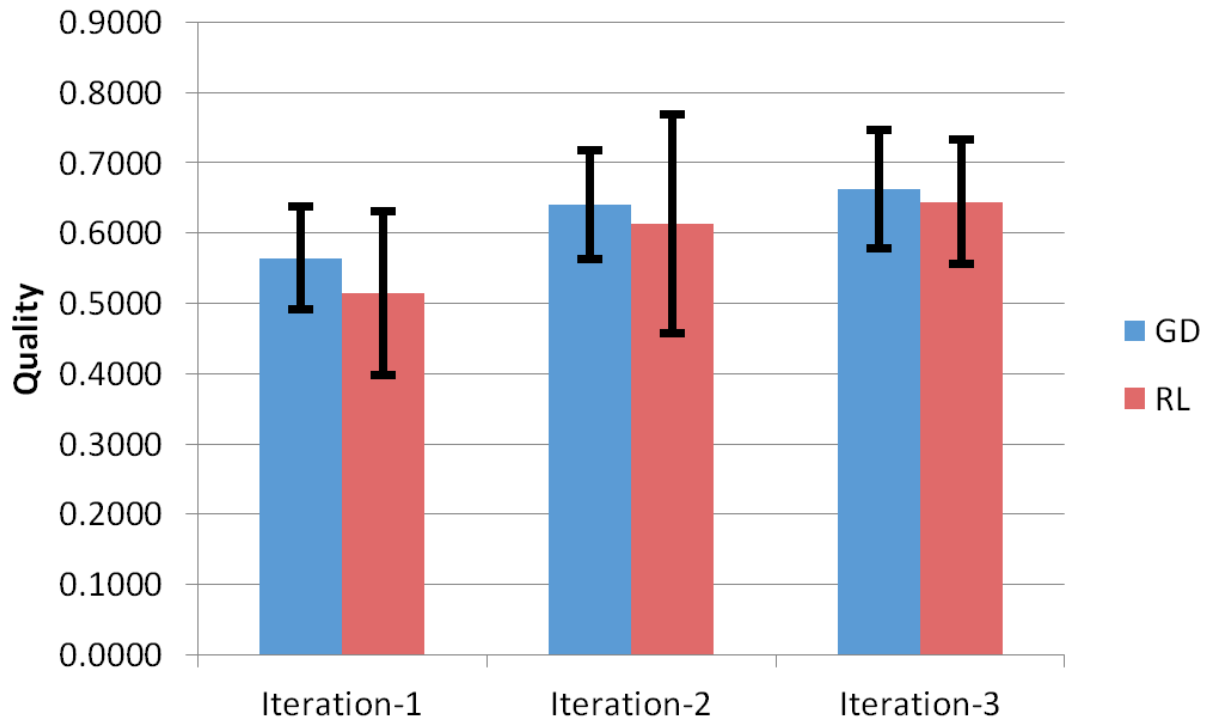


Figure 56: Database evaluation – Evaluation between learnings

We have consistent results on the database evaluation; Gradient Descent performs better on its first iteration however Reinforcement Learning overtakes on the second one because of its greater learning rate (Fig. 56). Similar as the previous validation, the 95% Confidence Interval shows that statistically Reinforcement Learning values are more disperse than Gradient Descent this is due Gradient Descent results are more significant than Reinforcement Learning. Both algorithms have a similar residual learning on their third iteration.

9.3.4 Online Academic Database Order

Our ISA provides better quality than the other Online Academic Search Engines, however this improved value may be biased because we show to the user the first 20 results in the order1 calculated by our proposed algorithm.

Table 16 shows the Quality for the four different Academic Databases (Google Scholar, IEEE Xplore, CiteSeerX and Microsoft Academic) for the first Iteration with its associated improvement when the results are shown to the user according to our ISA algorithm.

Table 16: Database Evaluation - Quality order by ISA

Query: Average Value – 6 Queries						
Search	Iteration	Q	σ	CR 95%	σ^2	p value
Google Scholar	1	0.5177	0.1350	0.1080	0.0182	0.0900
ISA	1	0.6494	0.1045	0.0836	0.0109	
I		25.44%	-	-	-	
IEEE Xplore	1	0.3282	0.1431	0.1145	0.0205	0.3358
ISA	1	0.4206	0.1716	0.1373	0.0294	
I		28.15%	-	-	-	
CiteSeerX	1	0.3630	0.2419	0.1936	0.0585	0.6270
ISA	1	0.4391	0.2822	0.2258	0.0796	
I		20.98%	-	-	-	
Microsoft Academic	1	0.4944	0.1314	0.1051	0.0173	0.1179
ISA	1	0.6475	0.1736	0.1389	0.0301	
I		30.96%	-	-	-	

Figure 57 below the Quality for the four different Online Academic Databases and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

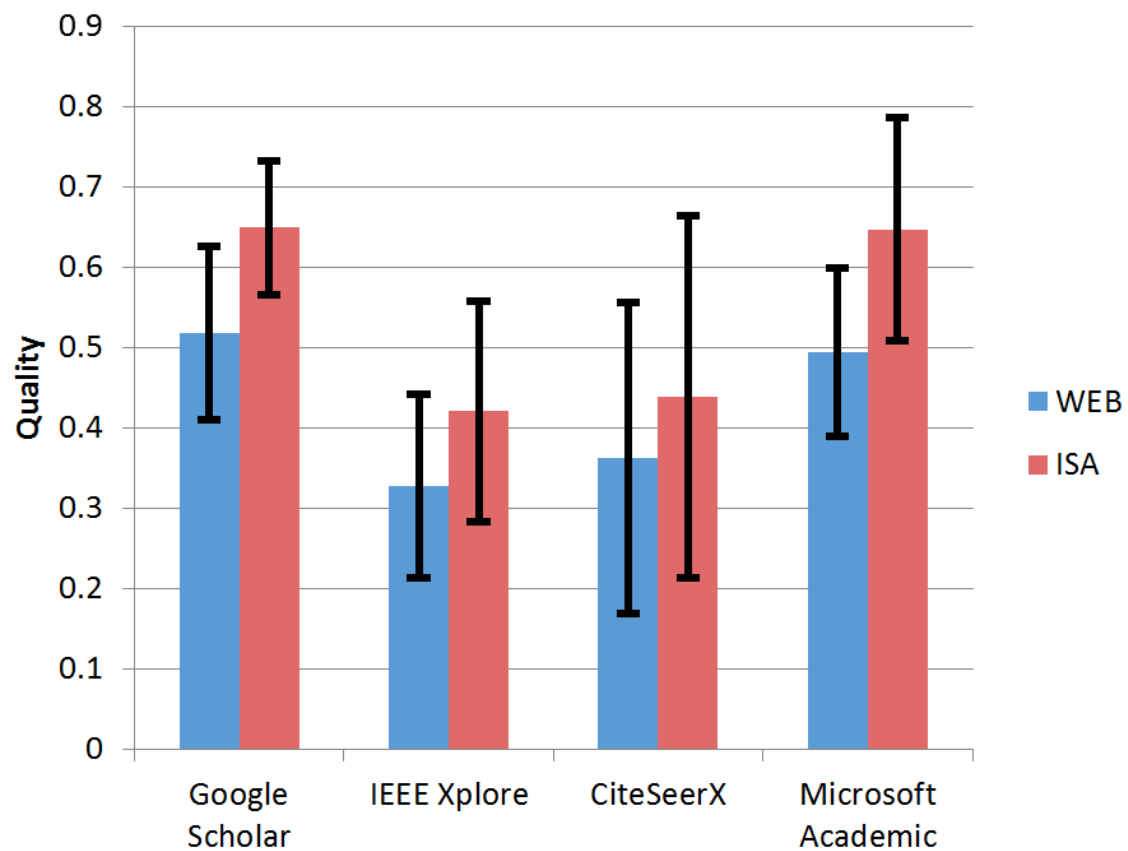
**Figure 57:** Database Evaluation - Quality order by ISA

Figure 58 shows the associated improvement.

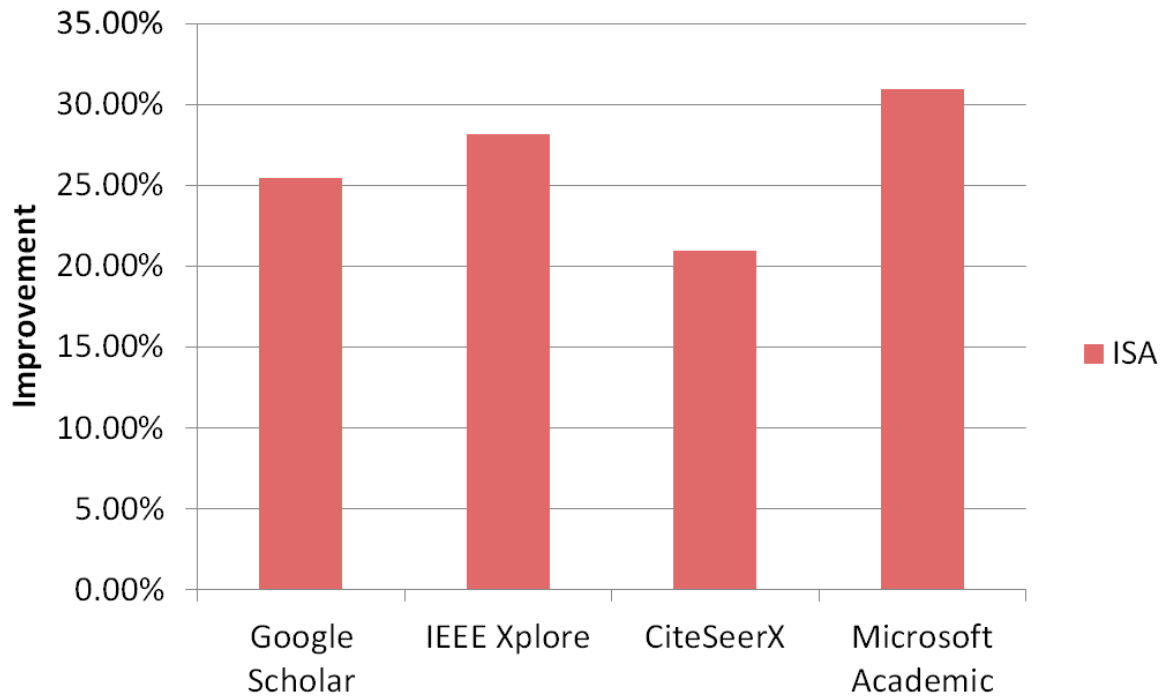


Figure 58: Database Evaluation - Improvement order by ISA

ISA outperforms other Web Search Engines (Fig. 57 and Fig. 58) when results are shown to the user following ISA Cost Function. Therefore, we have produced a further experiment on which we present to the user the results ordered according to the original Online Academic Database algorithms where the results are shown on Table 17.

Table 17: Database Evaluation - Quality order by Academic Database

Query: Average Value – 6 Queries						
Search	Iteration	Q	σ	CR 95%	σ^2	p value
Google Scholar	1	0.5340	0.1500	0.1200	0.0225	0.4676
ISA	1	0.4710	0.1386	0.1109	0.0192	
I		-11.79%		-		
IEEE Xplore	1	0.4243	0.2023	0.1619	0.0409	0.4546
ISA	1	0.3436	0.1522	0.1218	0.0232	
I		-19.01%		-		
CiteSeerX	1	0.4337	0.2501	0.2001	0.0626	0.6745
ISA	1	0.3765	0.2051	0.1641	0.0421	
I		-13.19%		-		
Microsoft Academic	1	0.4920	0.1510	0.1208	0.0228	0.3845
ISA	1	0.4175	0.1319	0.1055	0.0174	
I		-15.14%		-		

Figure 59 below the Quality for the four different Online Academic Databases and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

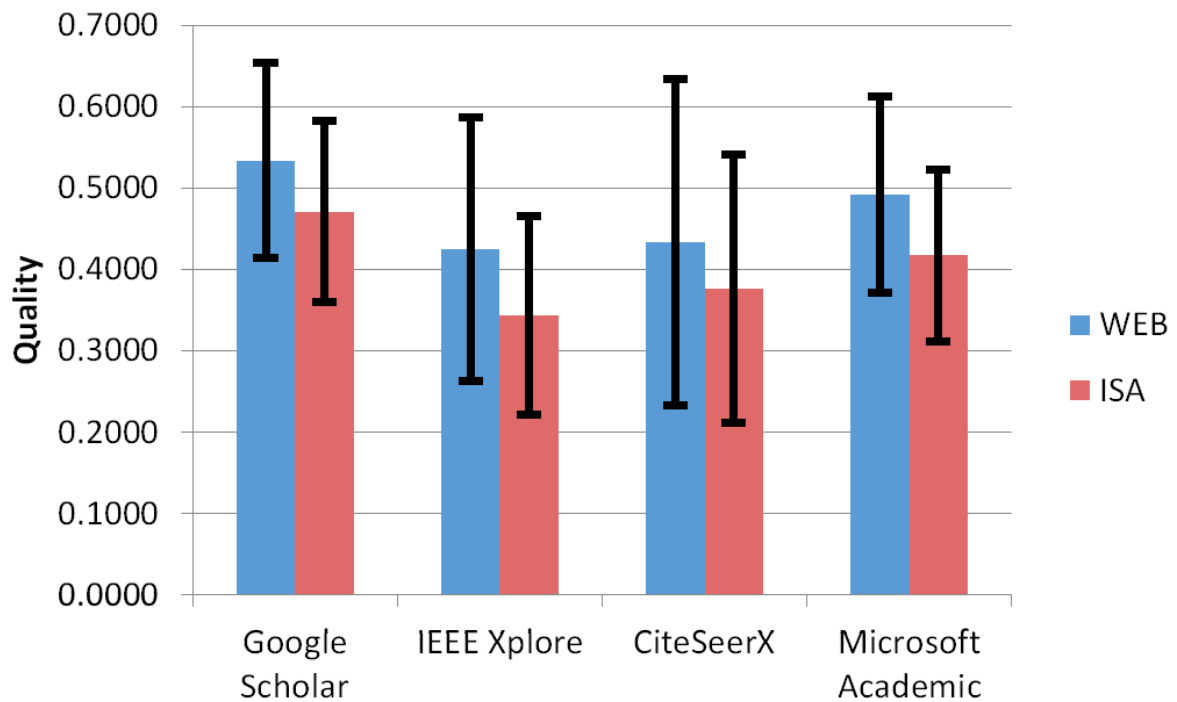


Figure 59: Database Evaluation - Quality order by Academic Database

Figure 60 shows the associated improvement.

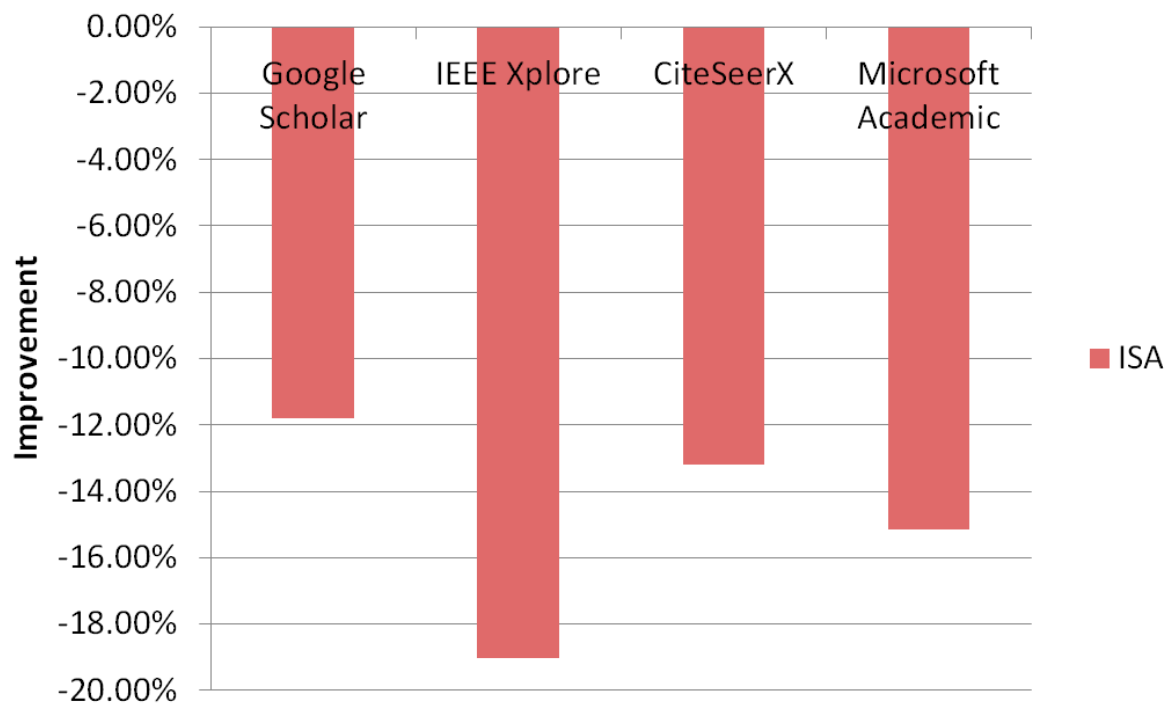


Figure 60: Database Evaluation - Improvement order by Academic Database

The order on which the results are presented to the user is relevant in the evaluation of the Quality measurements (Fig. 59, Fig. 60). Our ISA algorithm has now worse values than the other Online Academic databases although the 95% Confidence is similar with a great overlap region. A worse performing algorithm may seem better than its rivals if it is selected as a benchmark in the evaluation stage. We propose the final evaluation; we add both benchmark values to represent the fact that the better algorithm will score better on both combined situations where the results are shown on Table 18.

Table 18: Database Evaluation - ISA and Online Academic Database

Query: Average Value – 6 Queries						
Search	Iteration	Q	σ	CR 95%	σ^2	p value
Google Scholar	1	0.5258	0.1363	0.0771	0.0186	0.5624
ISA	1	0.5602	0.1496	0.0846	0.0224	
I		6.53%	-	-	-	
IEEE Xplore	1	0.3762	0.1744	0.0987	0.0304	0.9323
ISA	1	0.3821	0.1598	0.0904	0.0255	
I		1.56%	-	-	-	
CiteSeerX	1	0.3984	0.2375	0.1344	0.0564	0.9231
ISA	1	0.4078	0.2375	0.1344	0.0564	
I		2.38%	-	-	-	
Microsoft Academic	1	0.4932	0.1349	0.0763	0.0182	0.5654
ISA	1	0.5325	0.1898	0.1074	0.0360	
I		7.97%	-	-	-	

Figure 60 below the Quality for the four different Online Academic Databases and the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

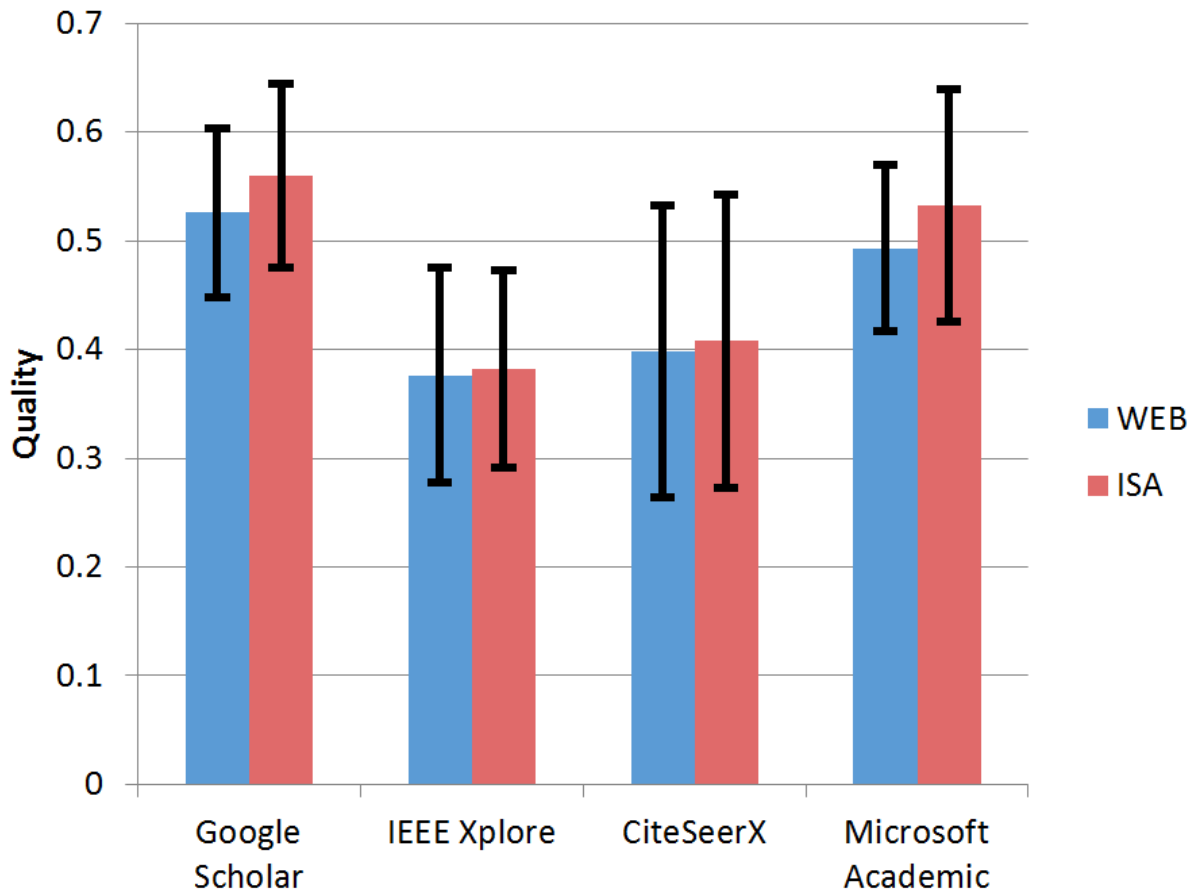


Figure 61: Database Evaluation - Quality ISA and Online Academic Database

Figure 61 shows the associated improvement.

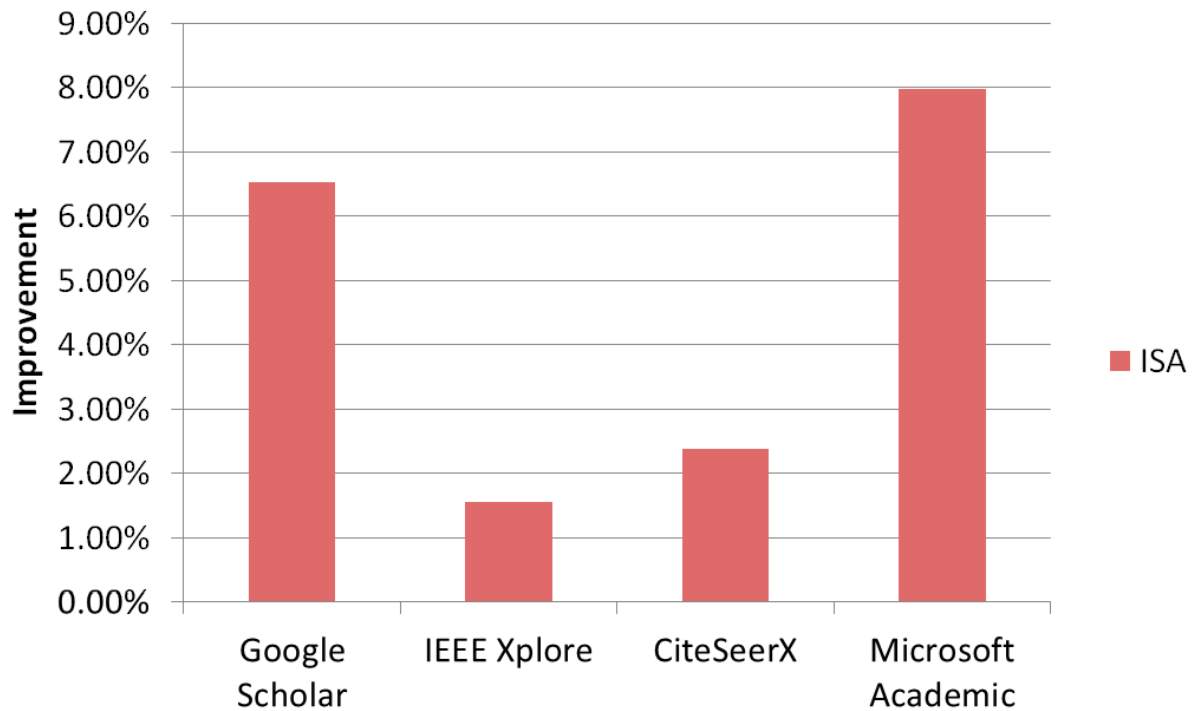


Figure 62: Database Evaluation - Improvement ISA and Academic Database

We can appreciate ISA still outperforms when both measurements are combined (Fig. 61, Fig. 62). The 95% Confidence Interval shows that both results greatly where the p value shows that the results are not significant for a significance level of $\alpha=0.05$. Although an arbitrary order could have been selected to provide a fair comparison on a single evaluation stage, we discarded this option because we show to the user the first 20 results from a universe of 50 results. Usually users only read the first 20 results losing interest for the following ones; relevant results shown arbitrarily on final orders may not have selected due the user's lack of interest.

9.3.5 Evaluation between Relevance Metrics

Table 19 covers Gradient Descent Learning for the 6 queries and Table 20 represents Reinforcement Learning for the 6 queries. The first column Table 19 and Table 20 represent MAP (mean average precision), TSAP (TREC style average precision), P (precision), NDCG (Normalized discounted cumulative gain), Q (quality), MRR (medium reciprocal rank) and ERR (expected reciprocal rank) for the first Iteration.

Table 19: Relevance Metric evaluation – Gradient Descent Average

Average Values – Gradient Descent – 6 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.4449	0.0814	0.5542	0.7244	0.4390	0.6334
ISA	1	0.7050	0.1201	0.5542	0.8550	0.5642	0.9333
Correlation							
Web	1	1.34%	-81.46%	26.24%	65.01%	0	44.28%
ISA	1	24.96%	-78.71%	-1.77%	51.54%	0	65.42%

Table 20: Relevance Metric evaluation –Reinforcement Learning Average

Average Values – Reinforcement Learning – 6 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.5133	0.0864	0.4958	0.7871	0.4127	0.7199
ISA	1	0.7542	0.1055	0.4958	0.8656	0.5141	0.7662
Correlation							
Web	1	24.38%	-79.06%	20.14%	90.72%	0	74.44%
ISA	1	46.70%	-79.48%	-3.56%	68.37%	0	49.04%

Figure 63 and Figure 64 show the correlation value for Gradient Descent and Reinforcement Learning respectively.

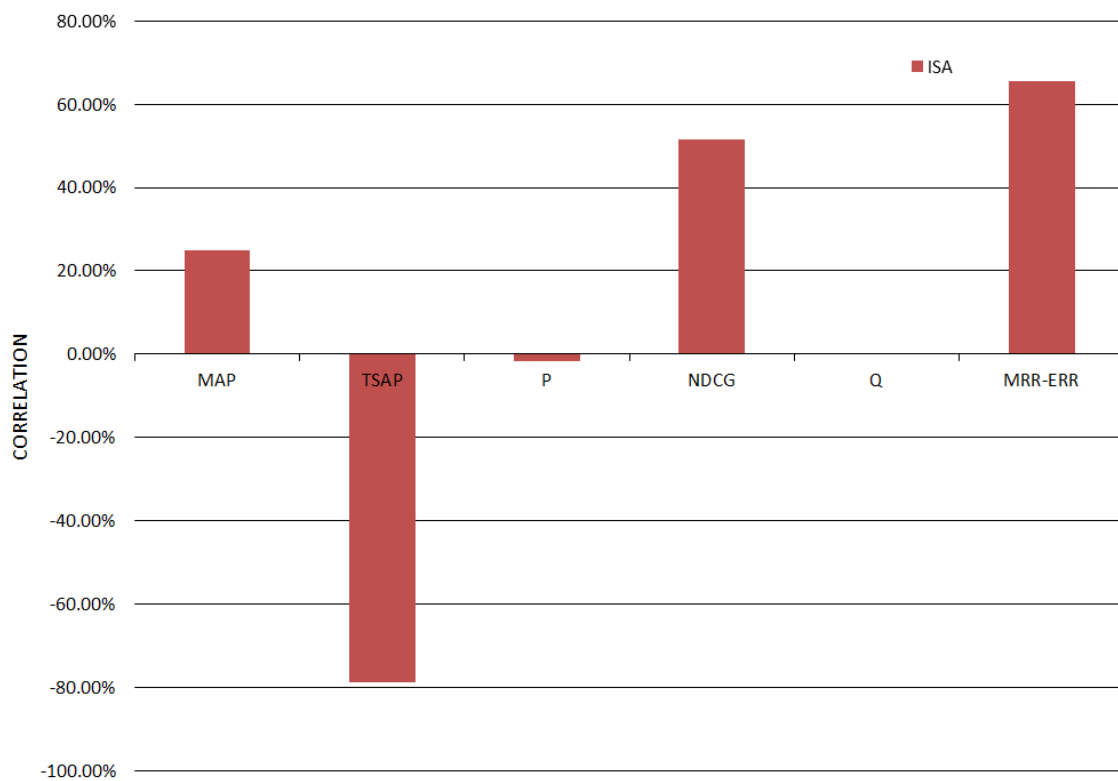


Figure 63: Relevance Metric Evaluation – Gradient Descent

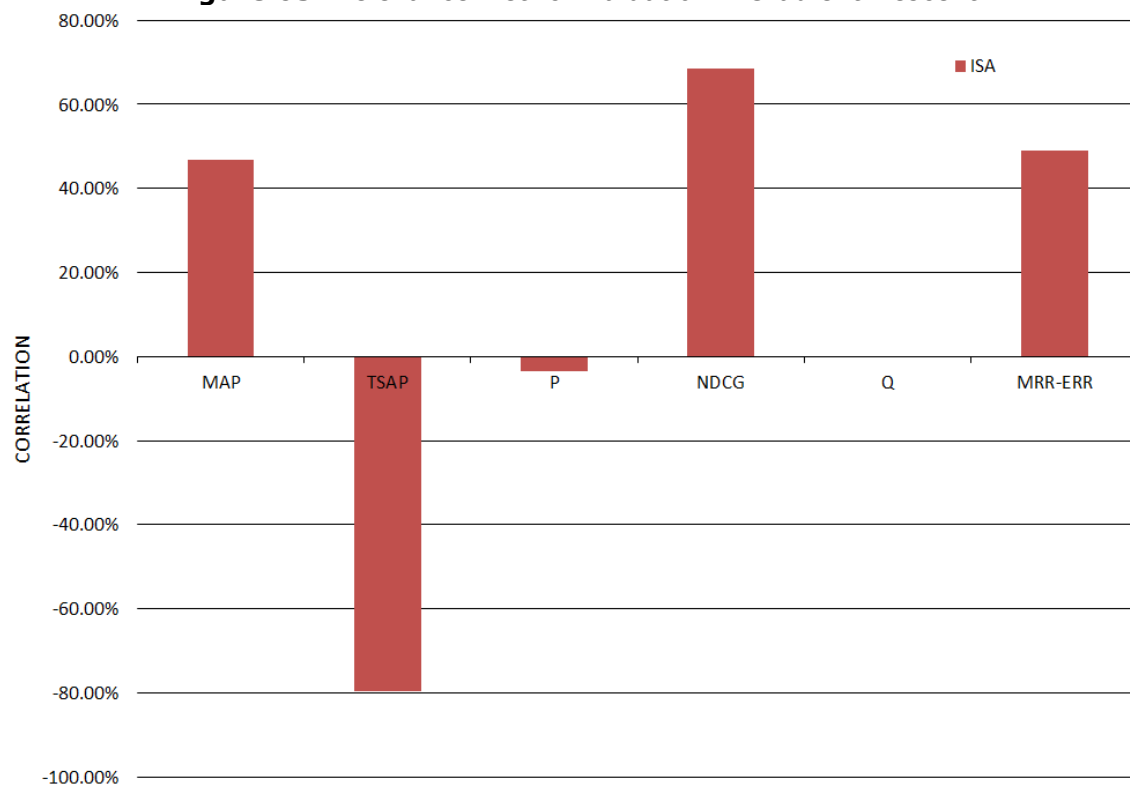


Figure 64: Relevance Metric Evaluation – Reinforcement Learning

Similar to our previous evaluation, our results show that we cannot determine which metric would have measured better the relevance of the different results in our user queries (Fig. 63, Fig. 64). The metric we have proposed, quality, is consistent with other metrics; it is very correlated with Precision and slightly correlated with Medium Average Precision measurements.

9.4 Recommender System Evaluation

We have implemented our Intelligent Search Assistant to reorder the results from three different independent Recommender Systems: GroupLens film database, Trip Advisor and Amazon.

Our ISA reorders the films or products based on the updated result relevance calculated by combining only the value of the relevant selected dimensions. The higher the value the more relevant the film or product should be. It shows to the user the first 20 results including its ranking. The user then selects the films or products with higher ranking; this ranking has been previously calculated by adding user reviews to the same products and calculating the average value.

In our evaluation we have asked users to select relevant results, not to rank them, as they normally do when using a Recommender System therefore we consider a result is either relevant or irrelevant. Validators are 15 personal friends from Imperial College students, researchers and London young professionals degree educated.

9.4.1 GroupLens film dataset

GroupLens is a research group in the Department of Computer Science and Engineering at the University of Minnesota. Since its creation in 1992 GroupLens' research projects have consisted on Recommender Systems, Online Communities, Mobile Technologies, Digital Libraries and Local Geographic Information Systems.

The dataset is based on a 5 star rating and genre tagging from MovieLens. It contains 21063128 ratings and 470509 tags across 27303 films. The dataset was created by 229060 users between January-1995 and March-2015. Any publication based on this dataset needs to acknowledge its origin.

The datasets files are written as comma-separated values, csv extension. The files are encoded as UTF-8. The ratings data file consist on userId, movieId, rating, timestamp whereas the film data file consist on movieId, title, genres. Genres are: Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western and IMAX.

We have processed the data set by extracting the relevant information; movieId, rating title and genres. We have combined all the ratings from individual users to the same different products where the average value is the final product rating. We consider each film as a multidimensional vector consisting on the different genres. The film relevance is the averaged film rating which is equally divided between the different genres it is classified.

We have programmed our ISA to retrieve the user's relevant film genres and the type of learning (Fig. 65). Our ISA then reorders the films based on their relevance which is the combined value of the user selected genres only.

Intelligent Search Assistant

☐ Action
☐ Adventure
☐ Animation
☐ Children
☐ Comedy
☐ Crime
☐ Documentary
☐ Drama
☐ Fantasy
☐ Romance
☐ Film-Noir
☐ Horror
☐ Musical
☐ Mystery
☐ Sci-Fi
☐ Thriller
☐ War
☐ Western
☐ IMAX

Learning Type:

Gradient Descent
Reinforcement Learning

Search!

Figure 65: ISA Recommender - Film interface

The Intelligent Search Assistant provides a reordered list of film titles to the user based on the higher values of the selected genres; the user then selects the results with a higher overall rating; our ISA then continues the search iteration process iteratively or provides with the final film titles.

We have selected Gradient Descent and Reinforcement Learning for five different queries with ten searches in total. Table 21 and Table 22 show the Quality for different iterations with its associated 95% Confidence Range. I represents the improvement from ISA against the Recommender System; the second I is between ISA iterations 2 and 1 and finally the third I is between ISA iterations 3 and 2. This section only evaluates the increment of quality between successive iterations by the learning algorithms therefore p values are not included.

Table 21: Recommender Evaluation – Film – Gradient Descent

First	IT-1	IT-2	IT-3
Query 01: Crime Documentary Drama Mystery – Gradient Descent			
0.735385343	0.877072363	0.877072363	0.876957785
Query 02: Action Adventure Thriller – Gradient Descent			
0.754249415	0.806270953	0.806270953	0.806270953
Query 03: Fantasy Mystery SciFi – Gradient Descent			
0.686551218	0.737118267	0.732560623	0.82911265
Query 04: Animation Children Fantasy Musical – Gradient Descent			
0.705778738	0.885939737	0.89649093	0.875310848
Query 05: Crime Drama Horror – Gradient Descent			
0.690477993	0.889532584	0.8896372	0.8896372

With average values:

Metric	First	IT-1	IT-2	IT-3
Q	0.7145	0.8392	0.8404	0.8555
σ	0.0294	0.0664	0.0702	0.0358
CR 95%	0.0257	0.0582	0.0615	0.0314
I	-	17.45%	0.15%	1.79%
σ^2	0.0009	0.0044	0.0049	0.0013

Table 22: Recommender Evaluation – Film – Reinforcement Learning

First	IT-1	IT-2	IT-3
Query 01: Crime Documentary Drama Mystery – Reinforcement Learning			
0.803931437	0.840779016	0.948078093	1

Query 02: Action Adventure Thriller – Reinforcement Learning			
0.754249415	0.754249415	0.800473605	0.842576506
Query 03: Fantasy Mystery SciFi – Reinforcement Learning			
0.718818775	0.772887283	0.839954079	0.874032301
Query 04: Animation Children Fantasy Musical – Reinforcement Learning			
0.713255804	0.733775852	0.795061068	0.888826819
Query 05: Crime Drama Horror – Reinforcement Learning			
0.789521127	0.876957785	0.958783806	0.999761905

With average values:

Metric	First	IT-1	IT-2	IT-3
Q	0.7560	0.7957	0.8685	0.9210
σ	0.0407	0.0606	0.0796	0.0739
CR 95%	0.0357	0.0532	0.0697	0.0648
I	-	5.26%	9.14%	6.05%
σ^2	0.0017	0.0037	0.0063	0.0055

Figure 66 shows the Quality for across the three different iterations for Gradient Descent and Reinforcement Learning Algorithms with the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

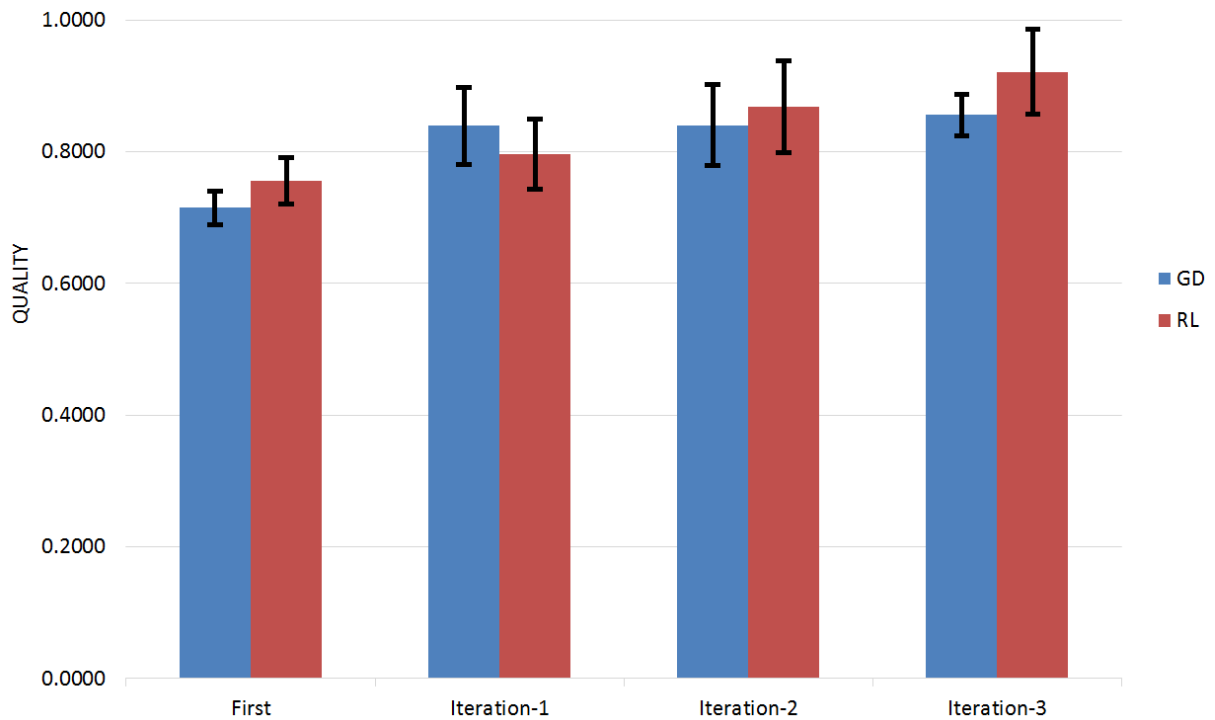


Figure 66: Recommender Evaluation – Film Database

Figure 67 shows the associated improvement.

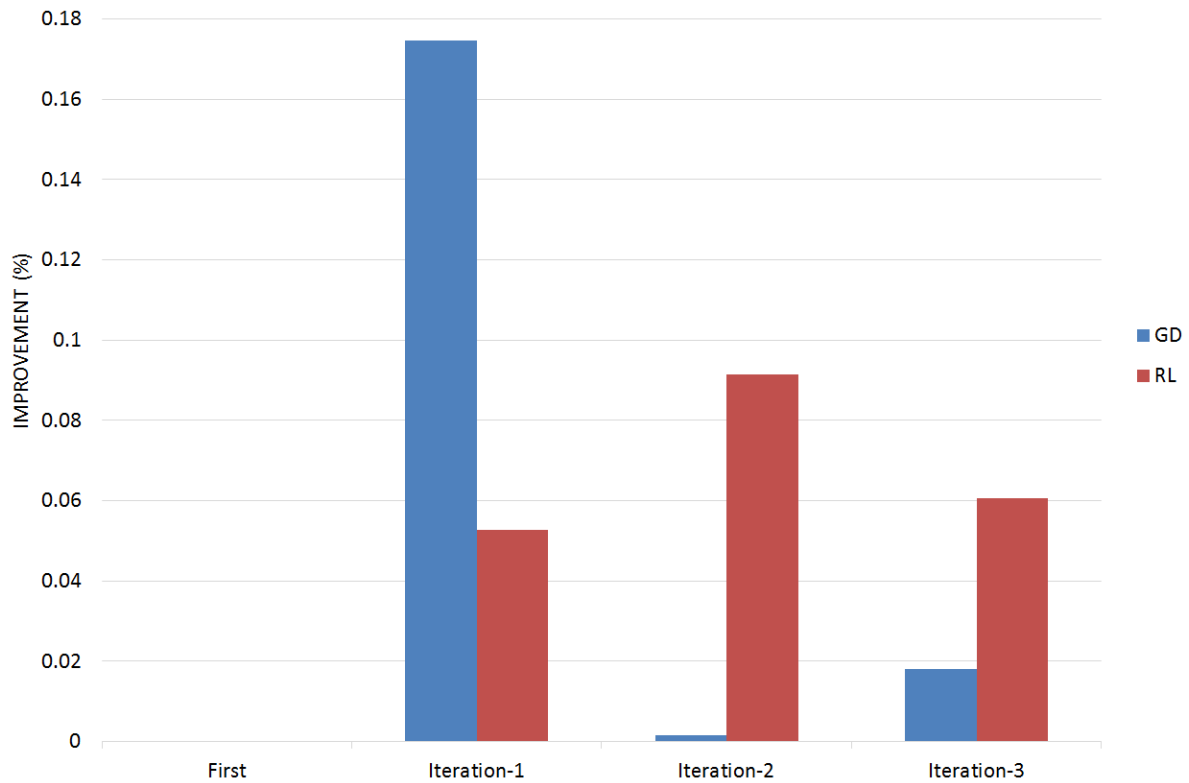


Figure 67: Recommender Evaluation - Improvement – Film Database

We can appreciate Gradient Descent learns mostly on its first iteration whereas Reinforcement Learning improvement is dispersed between the different user iterations (Fig. 66, Fig. 67). We can appreciate the 95% Confidence Interval is reduced when a Dataset is used; although the result regions still overlap. Gradient Descent outperforms Reinforcement Learning in the first iteration however Reinforcement Learning overtakes Gradient Descent due its continued learning rate.

9.4.2 Trip Advisor dataset

Trip Advisor dataset has been obtained from University of California-Irvine (UCI), Machine Learning repository, Centre for Machine Learning and Intelligent Systems. It is formed on two data sets; car and hotels reviews.

The car dataset is the full review of cars for model years 2007, 2008, and 2009. There are approximately from 140 to 250 different cars for each model year. The total number of reviews is approximately 42,230, (Year 2007: 18,903 reviews, Year 2008: 15,438 reviews and Year 2009: 7,947 reviews). Any publication based on this dataset needs to cite Kavita Ganesan et al [218].

The dataset format is car model, year, number of reviews, Fuel, Interior, Exterior, Build, Performance, Comfort, Reliability, Fun and overall rating.

The Hotel dataset is the full reviews of hotels in 10 different cities (Dubai, Beijing, London, New York City, New Delhi, San Francisco, Shanghai, Montreal, Las Vegas, Chicago). There are approximately from 80 to 700 hotels in each city. The total number of reviews is approximately 259,000.

The dataset format is hotel id, hotel name, hotel url, street, city, state, country, post code, number of reviews, Cleanliness, Room, Service, Location, Value, Comfort and overall rating.

We have processed the data set by extracting the relevant information; we have combined the ratings from different years into the same car type and we have also joined the hotel and car datasets into one. The average rating value is the final product (hotel or car) rating. Each product is a multidimensional vector consisting on the different variables the user can select. The product relevance is the product rating which it is equally divided between the different variables it is classified.

We have programmed our ISA to retrieve the user search product (hotel or car), relevant dimensions and the type of learning (Fig. 68). Our ISA then reorders the products based on their relevance which it is the combined value of the user selected variables only.

Intelligent Search Assistant

Product Type: Hotel
Car

Hotel

- ☐ Beijing
- ☐ Chicago
- ☐ Dubai
- ☐ Las Vegas
- ☐ London
- ☐ Montreal
- ☐ New Delhi
- ☐ New York
- ☐ San Francisco
- ☐ Shanghai
- ☐ Cleanliness
- ☐ Room
- ☐ Service
- ☐ Location
- ☐ Value

Car

- ☐ Fuel
- ☐ Interior
- ☐ Exterior
- ☐ Build
- ☐ Performance
- ☐ Comfort
- ☐ Reliability
- ☐ Fun

Learning Type: Gradient Descent
Reinforcement Learning

Search!

Figure 68: ISA Recommender – Trip Advisor interface

The Intelligent Search Assistant provides to the user with a reordered list of either hotels or cars re-ranked based on the higher values for the relevant dimensions using the Random Neural Network; the user then selects the results with a higher overall rating and our ISA continues the search iteration process or provides with the final hotel or car descriptions.

We have evaluated Gradient Descent and Reinforcement Learning for five different queries with ten searches in total for cars. Table 23, Table 24 show the Quality for different iterations with the associated 95% Confidence Range. I represents the improvement from ISA against the Recommender Systems; the second I is between ISA iterations 2 and 1 and finally the third I is between ISA iterations 3 and 2. This section only evaluates the increment of quality between successive iterations by the learning algorithms therefore p values are not included.

Table 23: Recommender Evaluation – Trip Advisor Car – Gradient Descent

First	IT-1	IT-2	IT-3
Query 01: Fuel Comfort Fun – Gradient Descent			
0.935868	0.941482	0.940070	0.939997
Query 02: Fuel Performance Reliability – Gradient Descent			
0.928926	0.936083	0.933617	0.934075

Query 03: Exterior Build Performance – Gradient Descent			
0.938193	0.932203	0.934075	0.934075
Query 04: Fuel Performance Comfort Reliability – Gradient Descent			
0.937842	0.941482	0.940070	0.939717
Query 05: Interior Exterior – Gradient Descent			
0.938748	0.929864	0.934075	0.934075

With Average values:

Metric	First	IT-1	IT-2	IT-3
Q	0.9359	0.9362	0.9364	0.9364
σ	0.0041	0.0053	0.0034	0.0032
CR 95%	0.0036	0.0046	0.0030	0.0028
I	-	0.0328%	0.0170%	0.0007%
σ^2	1.6E-05	2.8E-05	1.1E-05	1.0E-05

Table 24: Recommender Evaluation – Trip Advisor Car – Reinforcement Learning

First	IT-1	IT-2	IT-3
Query 01: Fuel Comfort Fun – Reinforcement Learning			
0.938733	0.945458	0.945507	0.945544
Query 02: Fuel Performance Reliability – Reinforcement Learning			
0.931425	0.944636	0.944621	0.945395
Query 03: Exterior Build Performance – Reinforcement Learning			
0.939349	0.945469	0.945469	0.945425
Query 04: Fuel Performance Comfort Reliability – Reinforcement Learning			
0.939617	0.945518	0.945510	0.945507
Query 05: Interior Exterior – Reinforcement Learning			
0.939621	0.945100	0.945251	0.945269

With average values:

Metric	First	IT-1	IT-2	IT-3
Q	0.9377	0.9452	0.9453	0.9454
σ	0.0036	0.0004	0.0004	0.0001
CR 95%	0.0031	0.0003	0.0003	0.0001
I	-	0.7984%	0.0038%	0.0165%
σ^2	1.3E-05	1.4E-07	1.4E-07	1.2E-08

Figure 69 shows the Quality for across the three different iterations for Gradient Descent and Reinforcement Learning Algorithms with the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

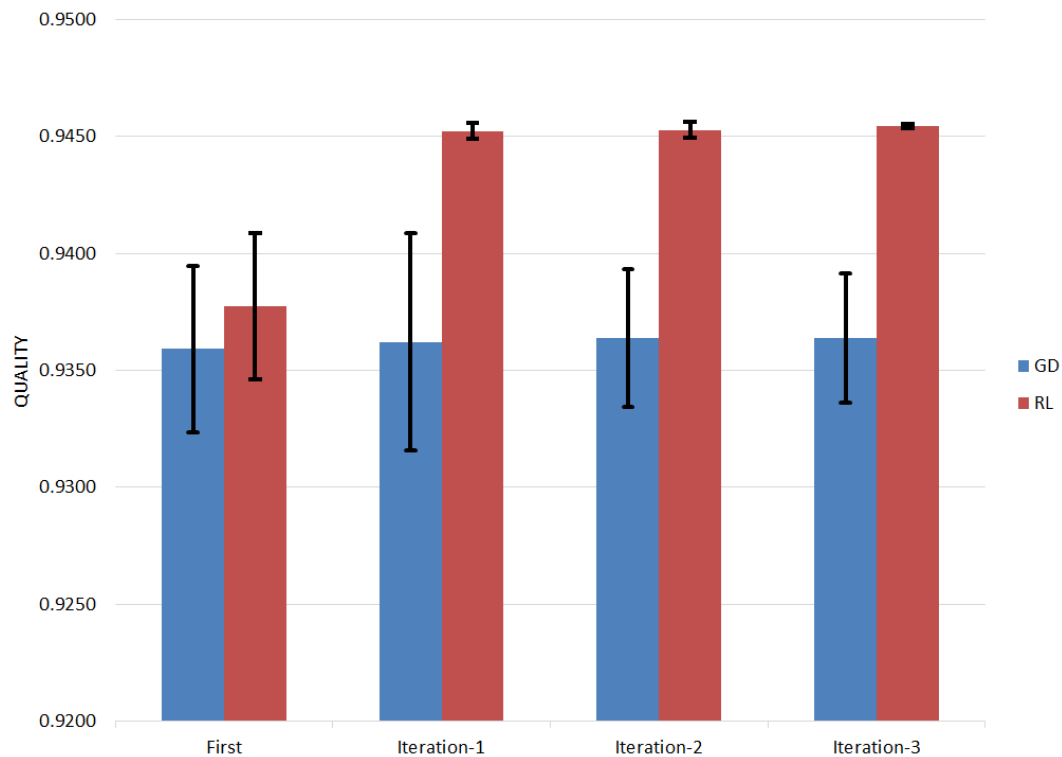


Figure 69: Recommender Evaluation – Trip Advisor Car Database

Figure 70 shows the associated improvement.

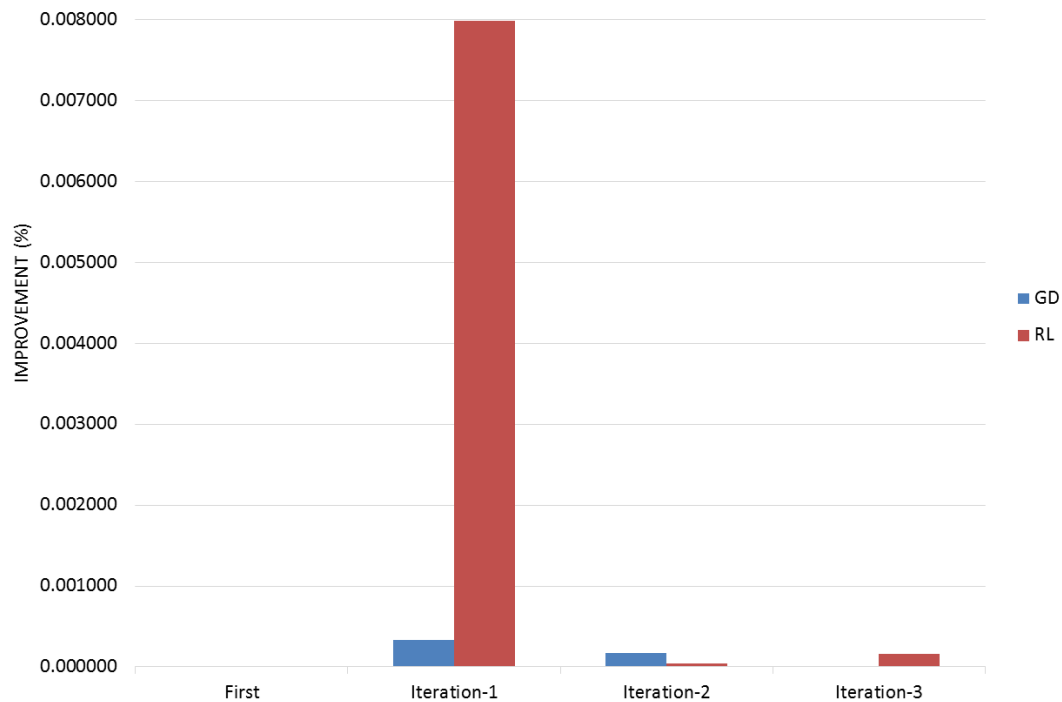


Figure 70: Recommender Evaluation - Improvement – Trip Advisor Car Database

We can appreciate both Gradient Descent and Reinforcement learns mostly on their first iteration with a residual learning on their successive iterations (Fig. 69, Fig. 70). This learning rate is mostly because the initial quality was very high therefore difficult to improve. The 95% Confidence Interval is reduced and negligible when Reinforcement Learning reaches its optimum value; Reinforcement Learning manages to get a higher value of quality than Gradient Descent.

We have evaluated Gradient Descent and Reinforcement Learning for five different queries with ten searches in total for hotels. Table 25 and Table 26 show the Quality for different iterations with the associated 95% Confidence Range. I represents the improvement from ISA against the Recommender Systems; the second I is between ISA iterations 2 and 1 and finally the third I is between ISA iterations 3 and 2. This section only evaluates the increment of quality between successive iterations by the learning algorithms therefore p values are not included.

Table 25: Recommender Evaluation – Trip Advisor Hotel – Gradient Descent

First	IT-1	IT-2	IT-3
Query 01: London Cleanliness Room Service – Gradient Descent			
0.957977	0.959278	0.959278	0.959278
Query 02: Beijing Cleanliness Room Service – Gradient Descent			
0.942178	0.949731	0.949731	0.949731
Query 03: Chicago Location Value – Gradient Descent			
0.925014	0.934280	0.934280	0.934280
Query 04: SanFrancisco Room Location – Gradient Descent			
0.921212	0.926948	0.924085	0.922225
Query 05: NewYork Cleanliness Room Service Location – Gradient Descent			
0.944469	0.945997	0.945997	0.945997

With average values:

Metric	First	IT-1	IT-2	IT-3
Q	0.9382	0.9432	0.9427	0.9423
σ	0.0151	0.0128	0.0137	0.0144
CR 95%	0.0132	0.0112	0.0120	0.0126
I	-	0.5411%	-0.0607%	-0.0395%
σ^2	2.3E-04	1.6E-04	1.9E-04	2.1E-04

Table 26: Recommender Evaluation – Trip Advisor Hotel– Reinforcement Learning

First	IT-1	IT-2	IT-3
Query 01: London Cleanliness Room Service – Reinforcement Learning			
0.959538	0.961927	0.998367	0.998531
Query 02: Beijing Cleanliness Room Service – Reinforcement Learning			
0.940871	0.954223	0.976550	0.983406
Query 03: Chicago Location Value – Reinforcement Learning			
0.927293	0.936577	0.987676	0.987581
Query 04: SanFrancisco Room Location – Reinforcement Learning			
0.922121	0.929453	0.987907	0.987907
Query 05: NewYork Cleanliness Room Service Location – Reinforc. Learning			
0.944697	0.946495	0.998442	0.998442

With average values:

Metric	First	IT-1	IT-2	IT-3
Q	0.9389	0.9457	0.9898	0.9912
σ	0.0148	0.0131	0.0091	0.0069
CR 95%	0.0130	0.0115	0.0080	0.0061
I	-	0.7276%	4.6581%	0.1399%
σ^2	2.2E-04	1.7E-04	8.3E-05	4.8E-05

Figure 71 shows the Quality for across the three different iterations for Gradient Descent and Reinforcement Learning Algorithms with the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

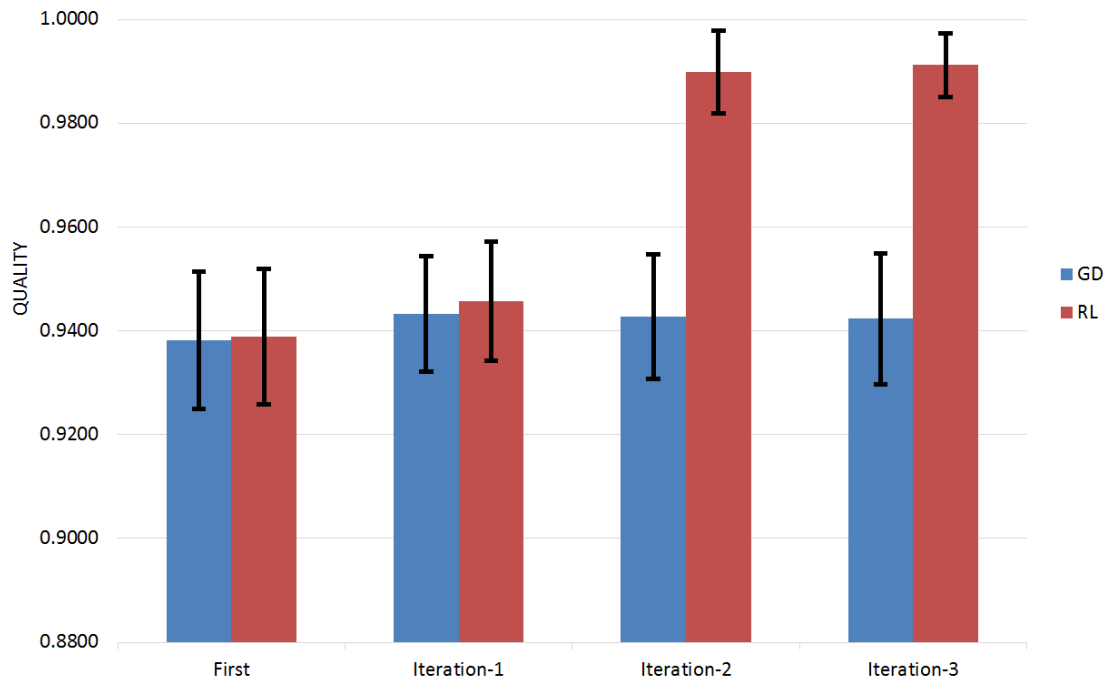


Figure 71: Recommender Evaluation – Trip Advisor Hotel Database

Figure 72 shows the associated improvement.

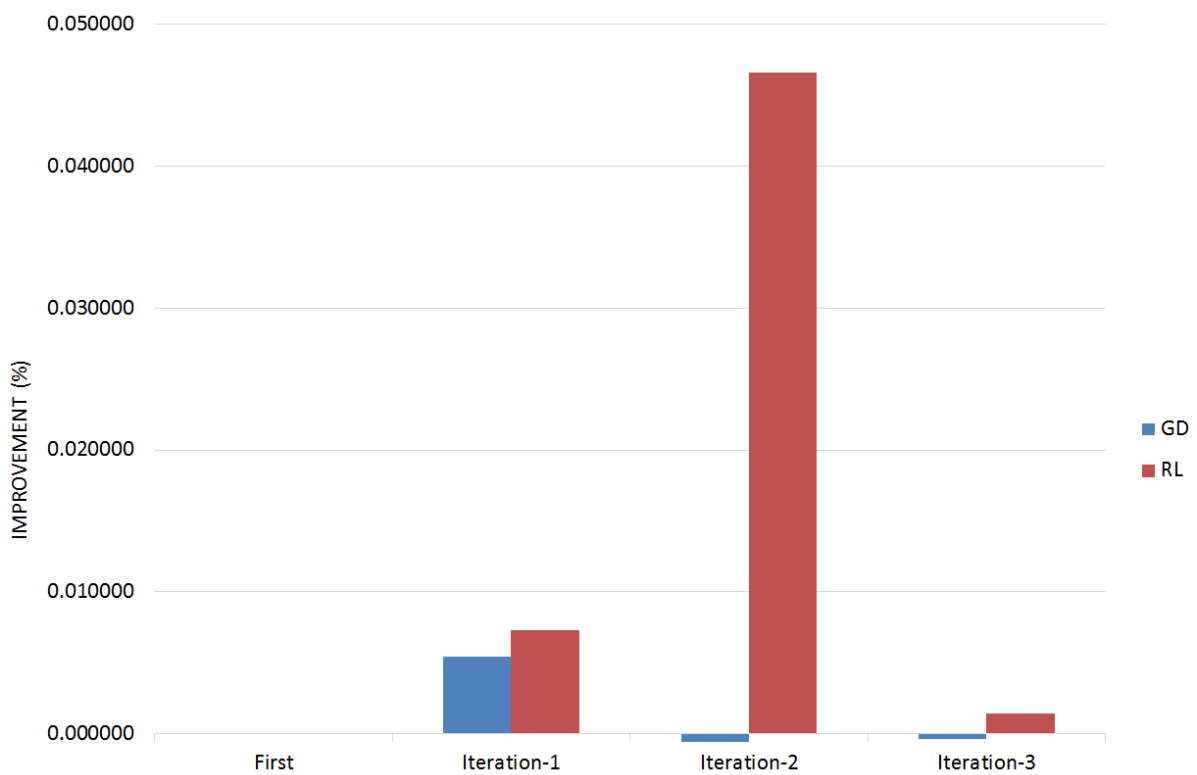


Figure 72: Recommender Evaluation -Improvement – Trip Advisor Hotel Database

The obtained results are consisting with the previous evaluation; we can appreciate both Gradient Descent and Reinforcement algorithms learn mostly on their first iteration with a residual learning on their successive iterations (Fig. 71, Fig. 72). We can appreciate Reinforcement Learning has a learning peak on its second iteration, this is because it provides the best scoring hotels in all cities whereas Gradient Descent still provides the best scoring hotels in the selected city only.

9.4.3 Amazon dataset

The Amazon dataset contains product reviews and metadata, including 143.7 million reviews spanning from May 1996 to July 2014. The review data is a 18 GByte file. The subsets are: Books; Electronics; Movies and TV; CDs and Vinyl; Clothing; Shoes and Jewelry; Home and Kitchen; Kindle Store; Sports and Outdoors; Cell Phones and Accessories; Health and Personal Care; Toys and Games; Video Games; Tools and Home; Improvement; Beauty; Apps for Android; Office Products; Pet Supplies; Automotive; Grocery and Gourmet Food; Patio, Lawn and Garden; Baby; Digital Music; Musical Instruments and Amazon Instant Video.

Due the large processing time to analyse the entire data set, we have selected only the Films & TV subset. Any publication based on this dataset needs to cite Julian McAuley et al [219] and [220]

Each Amazon subclass dataset is form of two different sub sets:

The review set contains the reviewerID, productID, reviewer name, rating of the review, review text, rating of the product, summary of the review, time of the review.

The metadata set contains productID, name of the product, price, url of the product image, related products (also bought, also viewed, bought together, buy after viewing), sales rank information, brand name and the list of categories the product belongs to.

We have processed the data set by extracting the relevant information and combining all the ratings from individual users to the same different products. The average value is the final product rating. We consider each product as its description or snippet; like in the previous Web Search evaluation (Fig. 73). Each product is a multidimensional vector consisting of the different properties or dimensions defined by the user on his high level query.

Intelligent Search Assistant

Dimension 1
Dimension 2
Dimension 3
Dimension 4
Dimension 8

Learning Type:

Gradient Descent
 Reinforcement Learning

Search!

Figure 73: ISA Recommender – Amazon interface

The Intelligent Search Assistant provides to the user with a reordered list of products re-ranked based on the same cost described for Web search using the Random Neural Network; the user then selects the results with a higher overall rating and our ISA continues the search iteration process or provides with the final products.

We have validated Gradient Descent and Reinforcement Learning for five different queries with ten searches in total. Table 27 and Table 28 show the Quality for different iterations with its associated 95% Confidence Range. I represents the improvement from ISA against the Recommender System; the second I is between ISA iterations 2 and 1 and finally the third I is between ISA iterations 3 and 2. This section only evaluates the increment of quality between successive iterations by the learning algorithms therefore p values are not included.

Table 27: Recommender Evaluation – Amazon – Gradient Descent

First	IT-1	IT-2	IT-3
Query 01: New York City– Gradient Descent			
0.150068	0.269169	0.273765	0.273765
Query 02: space exploration research– Gradient Descent			
0.161905	0.139810	0.139714	0.139714
Query 03: London– Gradient Descent			
0.089714	0.105714	0.105714	0.105714
Query 04: Silicon valley – Gradient Descent			
0.171231	0.236374	0.317516	0.244231
Query 05: great depression– Gradient Descent			
0.152262	0.219881	0.215667	0.215667

With average values:

Metric	First	IT-1	IT-2	IT-3
Q	0.1450	0.1942	0.2105	0.1958

σ	0.0320	0.0686	0.0887	0.0709
CR 95%	0.0281	0.0602	0.0777	0.0621
I	-	33.89%	8.39%	-6.96%
σ^2	1.0E-03	4.7E-03	7.9E-03	5.0E-03

Table 28: Recommender Evaluation – Amazon – Reinforcement Learning

First	IT-1	IT-2	IT-3
Query 01: New York City– Reinforcement Learning			
0.150068	0.248333	0.371734	0.372634
Query 02: space exploration research– Reinforcement Learning			
0.161905	0.176190	0.176190	0.176667
Query 03: London– Reinforcement Learning			
0.089714	0.133143	0.133143	0.133143
Query 04: Silicon valley – Reinforcement Learning			
0.171231	0.171231	0.171231	0.171231
Query 05: great depression– Reinforcement Learning			
0.152262	0.216476	0.216476	0.220476

With average values:

Metric	First	IT-1	IT-2	IT-3
Q	0.1450	0.1891	0.2138	0.2148
σ	0.0320	0.0444	0.0931	0.0935
CR 95%	0.0281	0.0389	0.0816	0.0819
I	-	30.36%	13.05%	0.50%
σ^2	1.0E-03	2.0E-03	8.7E-03	8.7E-03

Figure 74 shows the Quality for across the three different iterations for Gradient Descent and Reinforcement Learning Algorithms with the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

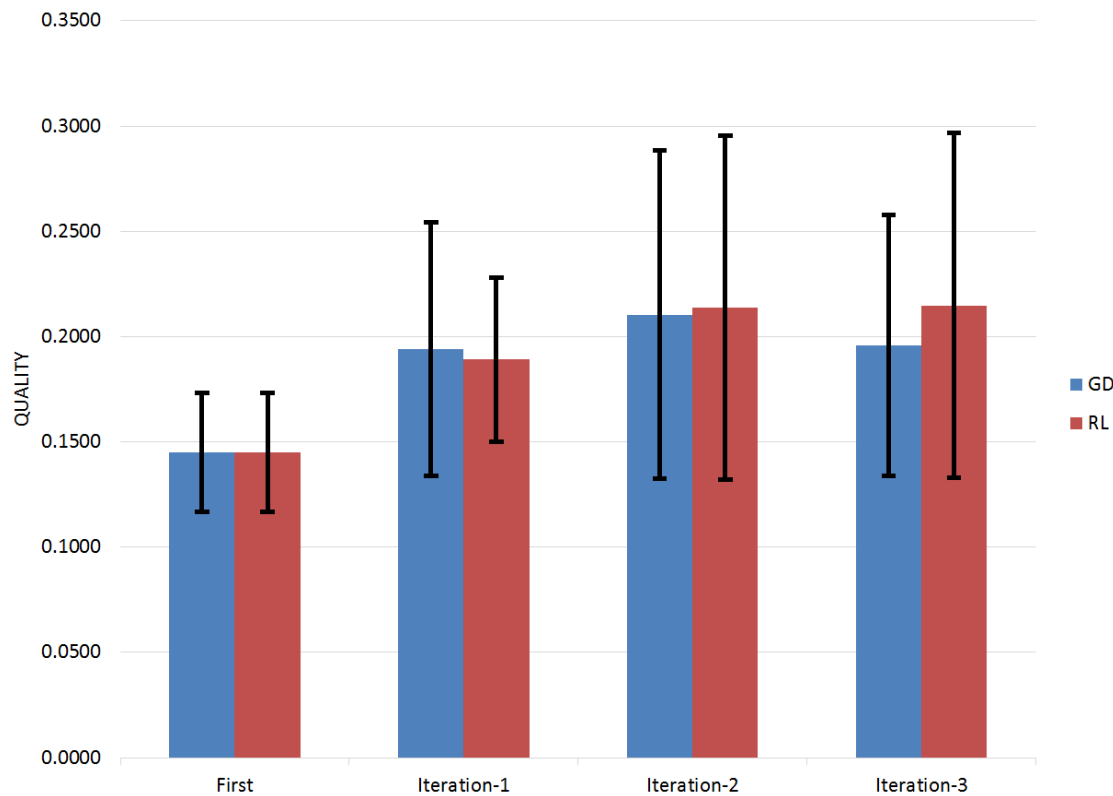


Figure 74: Recommender Evaluation – Amazon Database

Figure 75 shows the associated improvement.

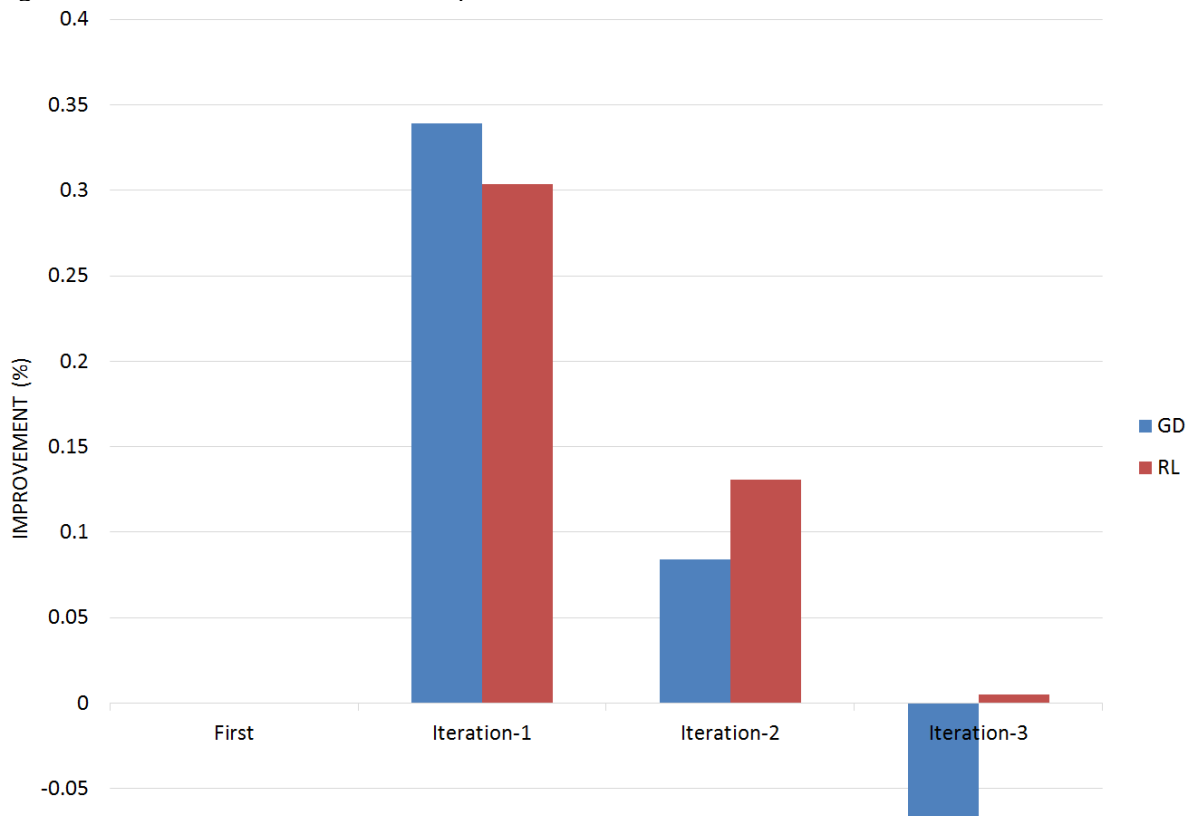


Figure 75: Recommender Evaluation - Improvement – Amazon Database

We can appreciate Gradient Descent learns mostly on its first iteration whereas Reinforcement Learning improvement is more dispersed among the different user iterations (Fig. 74, Fig. 75). The 95% Confidence Interval is greater in the amazon Dataset with a great result overlap. Gradient Descent outperforms Reinforcement Learning in the first iteration however Reinforcement Learning overtakes Gradient Descent due its continued learning rate. Gradient Descent Learning Relevant Centre has defocused on its third iteration producing a decrement on Quality.

Overall, the Recommender System evaluation has provided very similar results to the Web search and database evaluations with consistent conclusions. Gradient Descent mostly learns on its first iteration with a decrement rate learning in further stages. Reinforcement algorithm learns progressively with a learning rate dispersed within the learning stages until it reaches its maximum quality. Our algorithm provides very high quality with the Trip Advisor dataset at its first stage therefore the learning rate decreased significantly however it kept the same pattern: Gradient Descent performs better on its first iteration while Reinforcement Learning gradually keeps learning.

10 User Evaluation – Deep Learning

This section evaluates the increase of performance with the allocation of one Deep Learning Cluster per each Web Search Engine and the addition of a Deep Learning cluster to perform as a Management Cluster.

The proposed Deep Learning cluster structure (Fig. 76) included in ISA is evaluated against other Web search engines with open user queries. Deep Learning Clusters' Gradient Descent learning algorithm has been analysed based on result relevance and learning speed where our proposed method assigns the best performing cluster to the other Web Search Engines in order to improve result accuracy and relevance while reducing learning time. The Management Cluster unsupervised learning is analysed and its performance is compared against other search engines with a new proposed quality definition, which combines both relevance and rank.

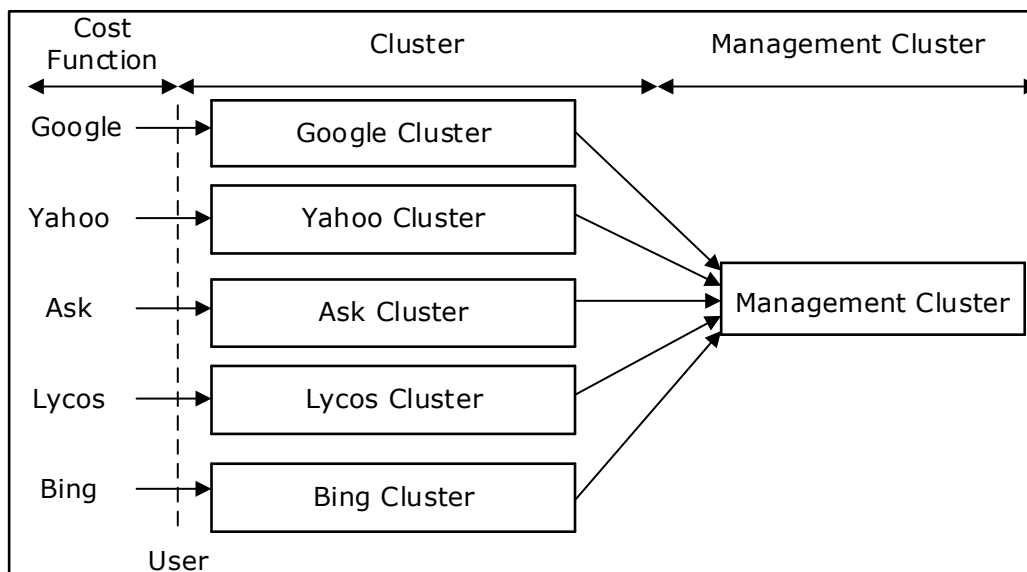


Figure 76: ISA Deep Learning Clusters model

In our evaluation we have asked users to select relevant results, not to rank them, as they normally do when using a Web search engine therefore we consider a result is either relevant or irrelevant. Validators are 15 personal friends from Imperial College students, researchers and London young professionals degree educated.

We have used available Web Search Engines in the current market. Although Yahoo search is powered by Bing and Ask.com has outsourced their search engine to an unknown third party; we have considered them independent and uncorrelated as they run on different Web pages; unless they provide the same results on the same order.

10.1 Implementation

The Intelligent Search Assistant we have proposed emulates how Web search engines work by using a very similar interface to introduce queries and display results. Our ISA acquires up to eight different dimensions values from the user however Web search engine and number of result options are fixed.

Our ISA has been programmed to retrieve snippets from five Web search engines (Google, Yahoo, Ask, Lycos, Bing). Our process is transparent; our ISA gets the query from the user and sends it to the different search engines selected without altering it (Fig. 77). ISA provides to the user with a reordered list of results clustered by Web Search Engine. The results are reordered based on our predetermined cost function defined; the user then selects relevant results and then our ISA provide with the final reordered list clustered by the different Web Search Engines.

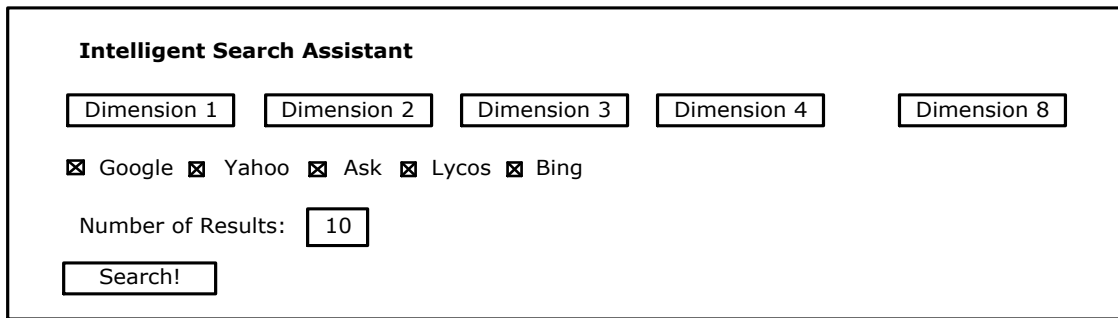


Figure 77: ISA Deep Learning cluster interface

10.2 Evaluation

A user in the experiment introduces a query. Our ISA assigns a Deep Learning cluster per Web Search Engine, acquires the first N results of each Web Search Engine and reorders them applying our cost function independently for each cluster. Finally our ISA shows a reordered list clustered by Web Search Engine. In our evaluation we have asked users to select Y relevant results per Web Search Engine, not to rank them, as they normally do when using a Web search engine therefore we consider a result is either relevant or irrelevant. The results are shown to the user on random order to avoid a biased result rank evaluation where users indirectly follow the order shown by the selected algorithm.

Each Deep Learning cluster learns each Web Search Engine Relevant Centre Point where the inputs i_u are the same values as the outputs y_c . Our ISA reorders each Web Search Engine cluster list according to the minimum error to the cluster Relevant Centre Point.

Once our ISA has established the best performing cluster; it uses its neural network weights to reorder all the retrieved results.

10.2.1 Deep Learning Cluster Evaluation

We define I as the quality improvement between a Web search engine and a reference:

$$I = \frac{QW - QR}{QR} \quad (1)$$

where I is the Improvement, QW is the quality of the Web search engine and QR is the quality reference. Figure 78 shows the Deep Learning Cluster Evaluation process.

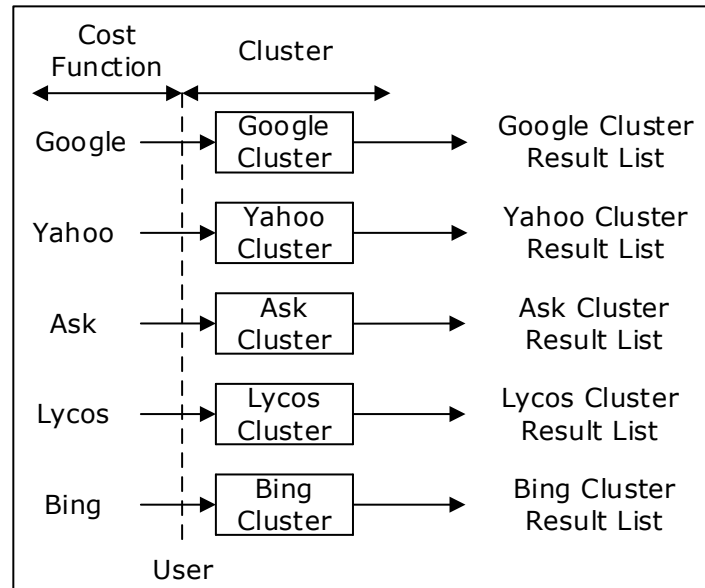


Figure 78: Deep Learning Cluster Evaluation

10.2.2 Management Cluster Evaluation

We define I as the quality improvement between a Deep Learning Cluster and a reference:

$$I = \frac{QC - QR}{QR} \quad (2)$$

where I is the Improvement, QC is the quality of the Deep Learning Cluster and QR is the quality reference.

We validate our management cluster with different values of $w^-(c)$:

$$w^-(c) = \left(\frac{1}{Q_c}\right)^\Phi \quad (3)$$

where Q_c is Quality of Cluster c and Φ the Management Cluster learning coefficient. Figure 79 shows the Management Cluster evaluation process.

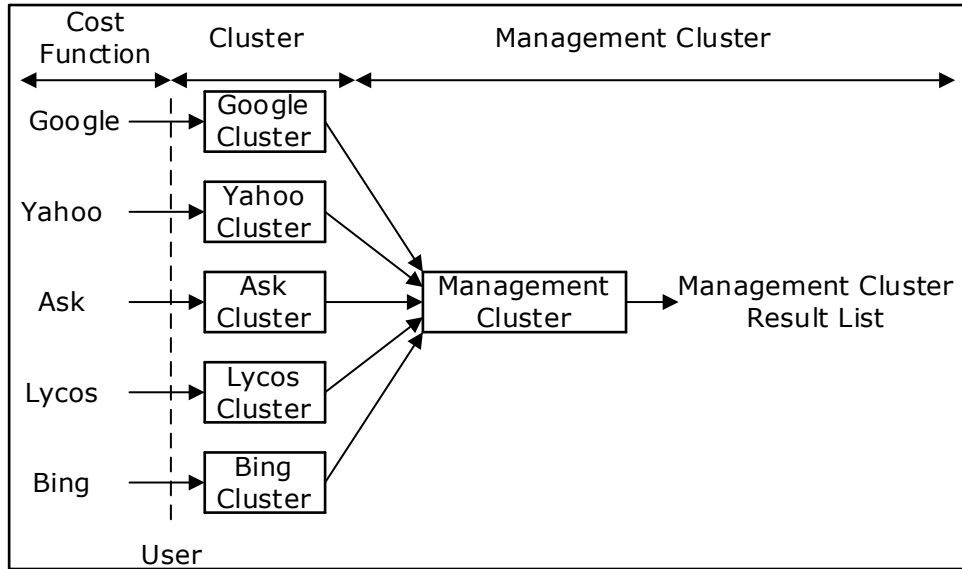


Figure 79: Management Cluster Evaluation

10.3 Experimental Results

We have asked our validators to introduce up to 8 dimensions per query. There are no rules in what users can search, however they have been advised their queries may be published. Our ISA acquires the first 10 results of each Web Search Engine, independently reorders them applying our cost function and finally shows a 50 result randomly reordered list joining the 10 result Web Search Engine cluster list. In our evaluation we have asked users to select 2 relevant results per Web Search Engine cluster list; this enables us to better compare performance between learning clusters.

10.3.1 Deep Learning Cluster

We have validated our proposed Deep Learning Cluster model with 45 different user queries. Table 29 show the Quality values with its associated 95% Confidence Range for the different Web search engines and Deep Learning Clusters before and after the user evaluation. We also show the Improvement between the quality of our Deep Learning clusters after and before the user evaluation. In addition; the Standard Deviation σ , the

variance σ^2 and the p value between the quality figures at different stages with different variance and a two tailed distribution are also shown to represent the statistical significance.

Table 29: Deep Learning Cluster Evaluation – Average Results

Metric	Google	Yahoo	Ask	Bing	Lycos
Q Web Search Engine	0.6846	0.6164	0.6573	0.6596	0.6421
σ	0.2044	0.2221	0.1946	0.1671	0.2098
CR 95%	0.0597	0.0649	0.0568	0.0488	0.0613
σ^2	0.0418	0.0493	0.0379	0.0279	0.0440
Q Cluster	0.5910	0.6164	0.6491	0.6167	0.6877
σ	0.1976	0.2037	0.2247	0.1926	0.1862
CR 95%	0.0577	0.0595	0.0657	0.0563	0.0544
σ^2	0.0390	0.0415	0.0505	0.0371	0.0347
Improvement Cluster-Web	-13.67%	0.00%	-1.25%	-6.50%	7.10%
p value	0.0298	1.0000	0.8538	0.2622	0.2784
Q Cluster Final	0.8201	0.8152	0.8047	0.7582	0.7556
σ	0.1731	0.2086	0.2137	0.2361	0.2156
CR 95%	0.0506	0.0610	0.0624	0.0690	0.0630
σ^2	0.0300	0.0435	0.0457	0.0557	0.0465
Improvement Cluster Final - Cluster	38.75%	32.26%	23.96%	22.95%	9.86%

With average values:

Web Search Engine	Quality web Search Engine	Quality Cluster	I	Quality Cluster Final	I
Average	0.6520	0.6322	-2.86%	0.79094	25.56%

Figure 79 shows the Quality at difference Stages for the different Deep Learning clusters with the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

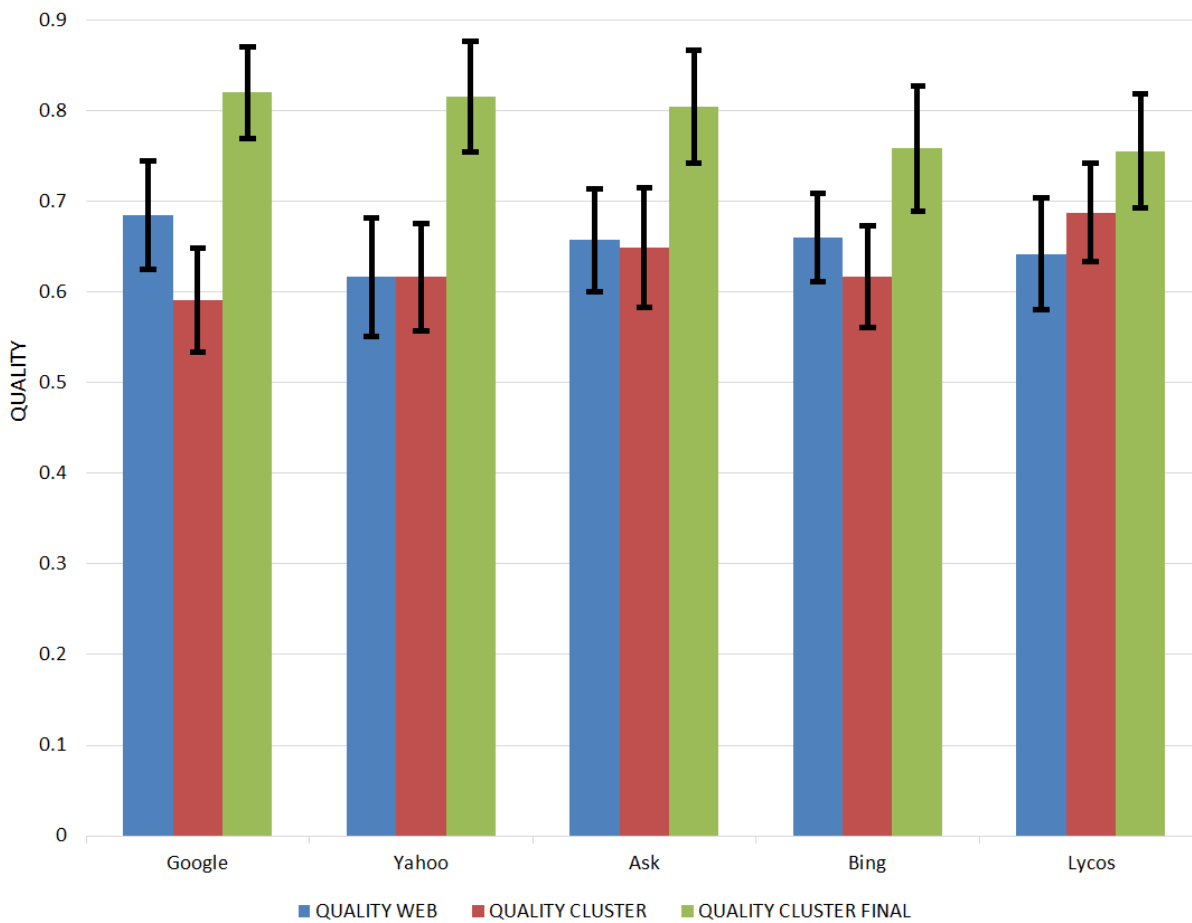


Figure 80: Deep Learning Cluster Evaluation – Average Results

On average; the Deep Learning clusters do not improve Web Search Engine performance before the user interaction (Fig. 79); however, they improve over 25% Quality after the first user iteration due to their capability to learn user relevance. The 95% Confidence Interval shows the results still overlap; although the increment of quality is clear. The p value shows that there is not significance between the Quality of the Web and the Quality of the clusters except for Google.

Table 30 show the Quality values of the different clusters when we select the neural weights of the best performing cluster and duplicate them to each Deep Learning cluster. We also show the Improvement between Learning Clusters. In addition; the Standard Deviation σ , the variance σ^2 and the p value between the quality figures at different stages with different variance and a two tailed distribution are also shown to represent the statistical significance.

Table 30: Best Performing Cluster Evaluation – Average Results

Metric	Google	Yahoo	Ask	Bing	Lycos
Q ISA Cluster	0.5910	0.6164	0.6491	0.6167	0.6877
σ	0.1976	0.2037	0.2247	0.1926	0.1862
CR 95%	0.0577	0.0595	0.0657	0.0563	0.0544
σ^2	0.0390	0.0415	0.0505	0.0371	0.0347
Q Best Performing Cluster	0.6554	0.6959	0.6421	0.6854	0.6351
σ	0.2081	0.2244	0.2645	0.2526	0.2567
CR 95%	0.0608	0.0656	0.0773	0.0738	0.0750
σ^2	0.0433	0.0504	0.0700	0.0638	0.0659
Improvement	10.88%	12.90%	-1.08%	11.14%	-7.65%
p value	0.1363	0.0819	0.8924	0.1506	0.2689

With average values:

Web Search Engine	Quality ISA Cluster	Quality ISA Best Performing Cluster	I
Average	0.6322	0.6628	5.24%

Figure 80 shows the Quality values of the different clusters when we select the neural weights of the best performing cluster and duplicate them to each Deep Learning cluster with the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

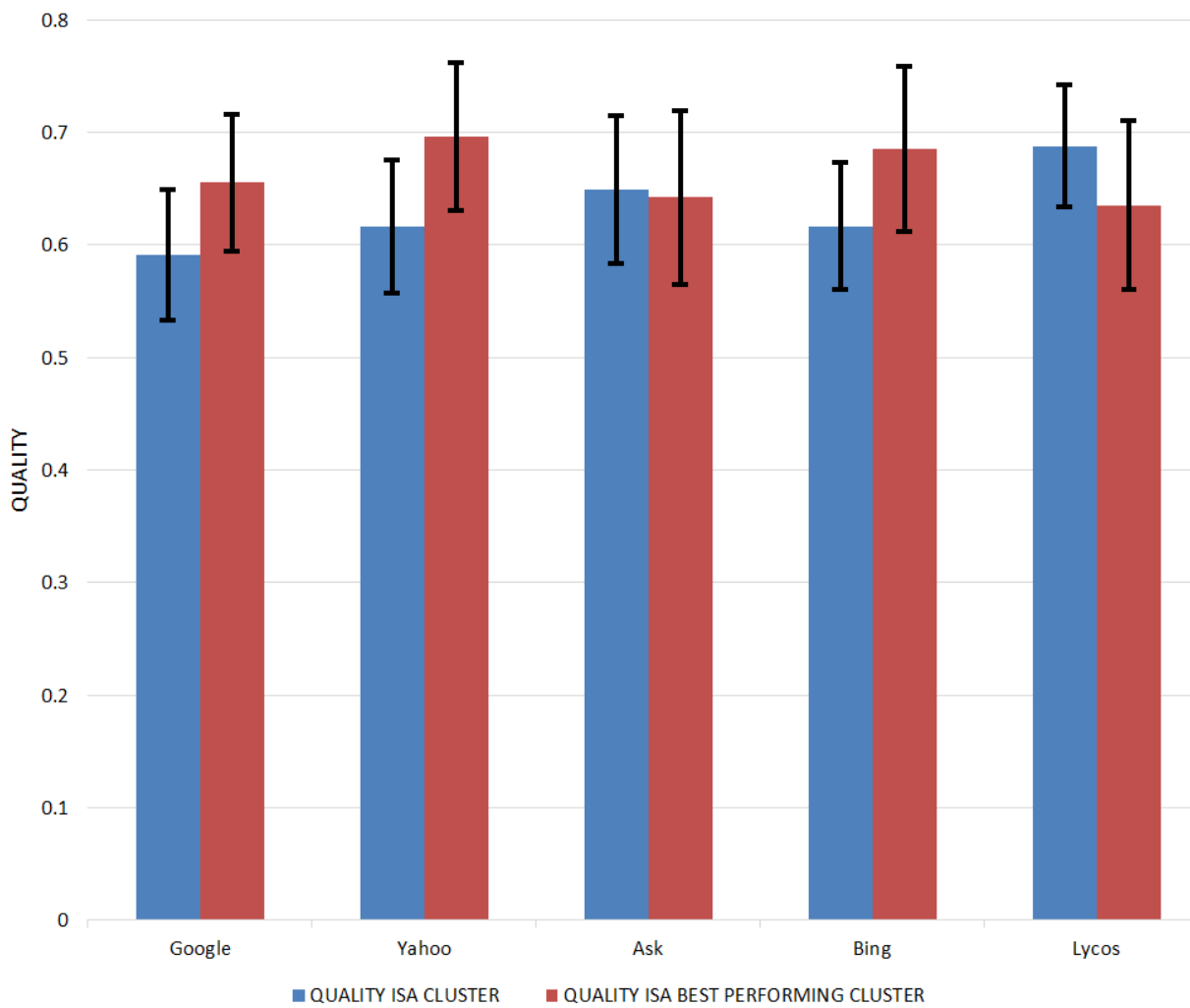


Figure 81: Best Performing Cluster Evaluation – Average Results

When the best performing Deep Learning cluster is applied to other Web Search Engines; the search Quality improves almost 5% on average (Fig. 80). The 95% Confidence Interval shows that results overlap where the p value shows that the results are not statistically significant. The best performing Deep Learning cluster can be used to teach or replace the other clusters; this improves result accuracy and relevance while reduces learning time and network weight's computation.

10.3.2 Management Cluster

We have validated our proposed Management Cluster model with 45 different user queries. Table 31 show the Quality values at the different stages: before the user evaluation (Cost Function), after the user evaluation (Best Cluster) and with the Management Cluster (Management Cluster) for five different Management Cluster

learning coefficients. We also show the Improvement between the quality of our Management cluster after and before the user evaluation.

Table 31: Management Cluster Evaluation – Average Results

Metric	$\Phi = 1/4$	$\Phi = 1/2$	$\Phi = 1.0$	$\Phi = 2.0$	$\Phi = 4.0$
Cost Function Q	0.2389	0.2389	0.2389	0.2389	0.2389
σ	0.0515	0.0515	0.0515	0.0515	0.0515
CR 95%	0.0151	0.0151	0.0151	0.0151	0.0151
σ^2	0.0027	0.0027	0.0027	0.0027	0.0027
Cluster Q	0.2767	0.2767	0.2767	0.2767	0.2767
σ	0.0735	0.0735	0.0735	0.0735	0.0735
CR 95%	0.0215	0.0215	0.0215	0.0215	0.0215
σ^2	0.0054	0.0054	0.0054	0.0054	0.0054
Management Cluster Q	0.2789	0.2776	0.2759	0.2712	0.2627
σ	0.0668	0.0671	0.0676	0.0691	0.0715
CR 95%	0.0195	0.0196	0.0197	0.0202	0.0209
σ^2	0.0045	0.0045	0.0046	0.0048	0.0051
MC vs Cluster I	0.81%	0.34%	-0.30%	-1.97%	-5.05%
p value	0.8798	0.9499	0.9561	0.7183	0.3638
MC vs Cost Function I	16.75%	16.20%	15.46%	13.53%	9.97%
p value	0.0021	0.0029	0.0046	0.0139	0.0738

Figure 81 shows the Quality values for the five different Management Cluster learning coefficients with the 95% Confidence Interval that corresponds to $Q \pm 95\%CR$.

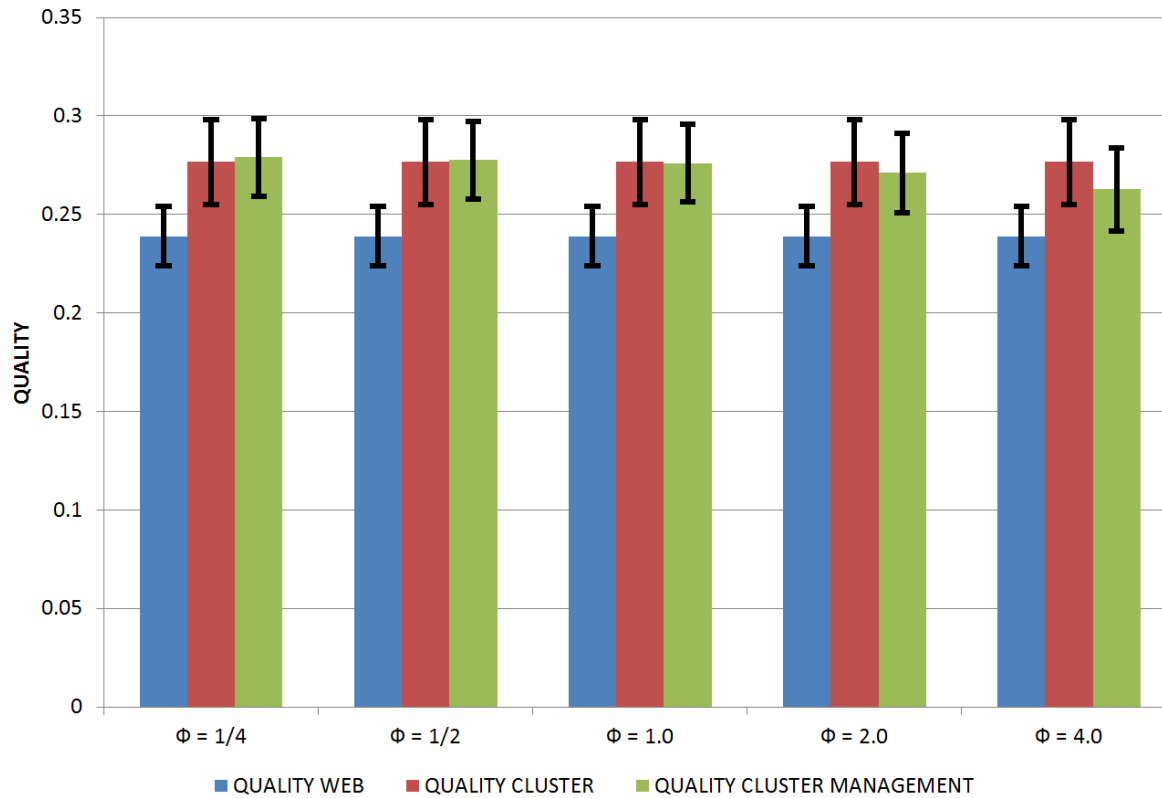


Figure 82: Management Cluster Evaluation – Average Results

The Quality improvement for the Deep Learning Management Cluster is very dependent with Φ with a greater chance to highly degrade Quality than slightly improve it (Fig. 81). The 95% Confidence Interval shows that the quality of the Cluster does not overlap with the Quality of the Web Search Engine; however; it overlaps when the Management Cluster is added. The results are statistically significant between the quality of the management cluster and the quality of the cost function; however; the quality between the quality of the management cluster and the deep learning cluster is not statistically significant.

11 Conclusions

We have proposed a novel approach to Web search and recommendation systems where the user iteratively trains the neural network while looking for relevant results. We have also defined a different process; the application of the Random Neural Network as a biological inspired algorithm to measure both user relevance and result ranking based on a predetermined cost function.

The main challenges faced by this research have been the technical Web Interaction with Web Search Engines and the secrecy from Web Search Engines and Recommender Systems to their algorithms and relevance measurements due their impact to their business model. This research has been based on the independency, unbiased and uncorrelated results between different Web Search Engines.

Another challenge presented during this research has been the retrieval of evaluation data; this research has obtained evaluation data from search experiments with real human validators rather than Web Search Engines or Recommender Systems commercial datasets with greater number of queries and evaluations as they were not publically available. The difficulty to obtain numerous user searches and experiments (in the order of 10s rather than 1000s) has increased our 95% Confidence Interval with large regions of overlap results. The results are shown as "proof of concept" rather than statistically significant or small confidence intervals however statistical significance to demonstrate estimation and hypothesis of the research data has been also reported within the evaluation sections.

Future work includes the application of more complex Deep Learning Cluster structures with different learnings as a mechanism to make relevant decisions in other fields such as Network Routing and Investment.

Our ISA has not scored better than the average of the other Web search engines, the main reason is the master result list has been evaluated by the score of the results provided by the Web search engines instead of the end user's evaluation. Our ISA has initially scored worse than Google, mainly because Google has provided the best result on its first position, however, after the first iteration with the user, our quality has improved and the first three results are shown within the 10 first positions. If precision is

used as a relevance metric instead of Quality then our ISA would have outperformed Google without user iteration and learning.

The results presented by the different Web search engines are retrieved from the same Web sites; however the order in which these are shown to the user constantly differs between them due to their own ranking algorithms. We conclude that there is not common approach between Web search engines regarding how to measure relevance when ordering results based on the same high level query.

We were expecting Metasearch engines to score better than Web search engines due their increased Web coverage however our research shows the opposite. Our proposed algorithm has scored better than both Metasearch engines proposed; however Metacrawler and Ixquick may have retrieved valid results from other sources (Online databases and Web directories) that we have not included in our result master list.

Our Intelligent Search Assistant performs slightly better than Google and other Web search engines however, this evaluation may be biased because users tend to concentrate on the first results provided which were the ones we showed in our algorithm. Our ISA adapts and learns from user previous relevance measurements increasing significantly its quality and improvement within the first iteration.

The order on which the results are presented to the user is relevant in the measurement of the Quality values. A worse performing algorithm can look better than its rivals if it is selected as a benchmark in the evaluation stage. Our ISA still outperforms when both ranking orders are combined in the evaluation. Our experiments show that we cannot determine which quality metric would have measured better the relevance of the different results in our user queries. The metric we have proposed, quality, is consistent with other metrics and it is very correlated with the Precision and Medium Average Precision measurements.

Reinforcement Learning algorithm performs better than Gradient Descent. Although Gradient Descent provides a better quality on the first iteration; Reinforcement Learning outperforms on the second one due its higher learning rate. Both of them have a residual learning on their third iteration. Gradient Descent would have been the preferred learning algorithm if only one iteration is required; however Reinforcement Learning would have been selected in the case of two iterations. It is not recommended three iterations

because learning is only residual. If we consider the property of a learning algorithm in relation to the adaptability to user changes; then Gradient Descent is more flexible to adapt to variable user interest than Reinforcement Learning. This is mostly because the rate of updating network weights.

In addition, we have presented a biological inspired learning algorithm the Random Neural Network in a Deep Learning cluster structure with a Management Cluster. We have validated it in a similar Big Data artificial environment where not all information can be processed due it is large amount the Web and users need to take relevant decisions: the Web. On average; our results prove that our Big Data Deep Learning clusters outperform other Web search engines with a significant improvement after the user iteration.

Cluster performance can be improved by learning from best performing clusters. Our model improves result accuracy and relevance while reduces learning time and network weigh's computation. On average; our results prove that our Deep Learning Clusters increase significantly the search Quality after the user's first interaction. Our Management cluster improves the Deep Learning Cluster Quality only when its learning coefficient is less than one; it has a detrimental effect if it is equal to or greater than one. Our Management Cluster needs to be the tuned to take the right decisions.

12 References

Intelligent Search Assistants

- [1] San Murugesan: Intelligent Agents On The Internet And Web: Applications and Prospects. Informatica. 23, 3 (1999)
- [2] Oren Etzioni, Daniel S. Weld: Intelligent Agents on the Internet: Fact, Fiction, and Forecast. IEEE Expert. 10,4, 44-49 (1995)
- [3] Nick Z. Zacharis, Themis Panayiotopoulos: A metagenetic algorithm for information filtering and collection from the World Wide Web. Expert Systems. 18,2, 99-108 (2001)
- [4] Michael J. Pazzani, Daniel Billsus: Learning and Revising User Profiles: The Identification of Interesting Web Sites. Machine Learning. 27,3, 313-331 (1997)
- [5] Henry Lieberman: Letizia: An Agent That Assists Web Browsing. International Joint Conference on Artificial Intelligence. 924-929 (1995)
- [6] Thorsten Joachims, Dayne Freitag, Tom M. Mitchell: Web Watcher: A Tour Guide for the World Wide Web. International Joint Conference on Artificial Intelligence, 770-777 (1997)
- [7] Bruce Krulwich: Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data. AI Magazine. 18,2, 37-45 (1997)

Web Search Engines

- [8] Amanda Spink, Bernard J. Jansen, Vinish Kathuria, Sherry Koshman: Overlap among major web search engines. Internet Research. 16,4, 419-426 (2006)
- [9] Alessandro Micarelli, Fabio Gaspiretti, Filippo Sciarrone, Susan Gauch: Personalized Search on the World Wide Web. The Adaptive Web. 195-230 (2007)
- [10] Nicolaas Matthijs, Filip Radlinski: Personalizing web search using long term browsing history. Web Search and Data Mining. 25-34 (2011)
- [11] Fang Liu, Clement T. Yu, Weiyi Meng: Personalized Web Search For Improving Retrieval Effectiveness. IEEE Transactions on Knowledge and Data Engineering. 16,1, 28-40 (2004)
- [12] Filip Radlinski, Susan T. Dumais: Improving personalized web search using result diversification. Special Interest Group on Information Retrieval. 691-692 (2006)
- [13] Lars Backstrom, Jon M. Kleinberg, Ravi Kumar, Jasmine Novak: Spatial variation in search engine queries. WWW. 357-366 (2008)

- [14] Fei Wu, Jayant Madhavan, Alon Y. Halevy: Identifying Aspects for Web-Search Queries. *Journal. Artificial. Intelligence Research* 40: 677-700 (2011)
- [15] Bernard J. Jansen, Tracy Mullen: Sponsored search: an overview of the concept, history, and technology. *International Journal of Electronic Business*. 6,2, 114-131 (2008)

Metasearch Engines

- [16] Weiyi Meng, Clement T. Yu, King-Lup Liu: Building efficient and effective metasearch engines. *ACM Computing Surveys*. 34,1, 48-89 (2002)
- [17] Manoj, M , Elizabeth, Jacob: Information retrieval on Internet using meta-search engines: A review. *Journal of Scientific & Industrial Research*, 67, 739-746 (2008)
- [18] Hossein Jadidoleslami: Search Result Merging and Ranking Strategies in Meta-Search Engines: A Survey. *International Journal of Computer Science*. 9, 239-251 (2012)
- [19] Antonio Gulli, Alessio Signorini: Building an open source meta-search engine. *WWW (Special interest tracks and posters)*. 1004-1005 (2005)
- [20] Zonghuan Wu, Vijay V. Raghavan, Hua Qian, Rama Vuyyuru, Weiyi Meng, Hai He, Clement T. Yu: Towards Automatic Incorporation of Search Engines into a Large-Scale Metasearch Engine. *Web Intelligence*. 658-661 (2003)
- [21] Javed A. Aslam, Mark H. Montague: Models for Metasearch. *Special Interest Group on Information Retrieval*. 275-284 (2001)
- [22] Cynthia Dwork, Ravi Kumar, Moni Naor, D. Sivakumar: Rank aggregation methods for the Web. *WWW*. 613-622 (2001)
- [23] Yiyao Lu, Weiyi Meng, Liangcai Shu, Clement T. Yu, King-Lup Liu: Evaluation of Result Merging Strategies for Metasearch Engines. *Web Information Systems Engineering*. 53-66 (2005)
- [24] R. Manmatha, Hayri Sever: A formal approach to score normalization for metasearch. *Human Language Technologies*. 98-103 (2002)
- [25] Leonidas Akritidis, Dimitrios Katsaros, Panayiotis Bozanis: Effective Ranking Fusion Methods for Personalized Metasearch Engines. *Panhellenic Conference on Informatics*. 39-43 (2008)
- [26] Weiyi Meng, Zonghuan Wu, Clement T. Yu, Zhuogang Li: A highly scalable and effective method for metasearch. *ACM Transactions Information Systems*. 19,3, 310-335 (2001)

- [27] B.T. Sampath Kumar, S.M. Pavithra: Evaluating the searching capabilities of search engines and metasearch engines: a comparative study. *Annals of Library and Information Studies*. 57, 87-97 (2010)

Web Result Clustering

- [28] Oren Zamir, Oren Etzioni: Web Document Clustering: A Feasibility Demonstration. *Special Interest Group on Information Retrieval*. 46-54 (1998)
- [29] Stanislaw Osinski, Dawid Weiss: A Concept-Driven Algorithm for Clustering Search Results. *IEEE Intelligent Systems*. 20,3, 48-54 (2005)
- [30] Claudio Carpineto, Stanislaw Osinski, Giovanni Romano, Dawid Weiss: A survey of Web clustering engines. *ACM Computing Surveys*. 41,3,17, 1-38 (2009)
- [31] Daniel Crabtree, Xiaoying Gao, Peter Andreae: Improving Web Clustering by Cluster Selection. *Web Intelligence*. 172-178 (2005)
- [32] Filippo Geraci, Marco Pellegrini, Marco Maggini, Fabrizio Sebastiani: Cluster Generation and Cluster Labelling for Web Snippets: A Fast and Accurate Hierarchical Solution. *String Processing Information Retrieval*. 25-36 (2006)
- [33] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, Jinwen Ma: Learning to cluster web search results. *Special Interest Group on Information Retrieval*. 210-217 (2004)
- [34] Jiyin He, Edgar Meij, Maarten de Rijke: Result diversification based on query-specific cluster ranking. *Journal of the Association for Information Science and Technology*. 62,3, 550-571 (2011)
- [35] Chi Lang Ngo, Hung Son Nguyen: A Method of Web Search Result Clustering Based on Rough Sets. *Web Intelligence*. 673-679 (2005)
- [36] Xuanhui Wang, ChengXiang Zhai: Learn from web search logs to organize search results. *Special Interest Group on Information Retrieval*. 87-94 (2007)
- [37] Zhao Li, Xindong Wu: A Phrase-Based Method for Hierarchical Clustering of Web Snippets. *Association for the Advancement of Artificial Intelligence*. 1947-1948 (2010)
- [38] Dell Zhang, Yisheng Dong: Semantic, Hierarchical, Online Clustering of Web Search Results. *Advanced Web Technologies and Applications*. 69-78 (2004)
- [39] Paolo Ferragina, Antonio Gulli: The Anatomy of a Hierarchical Clustering Engine for Web-page, News and Book Snippets. *International Conference on Data Mining*. 395-398 (2004)

- [40] Ricardo A. Baeza-Yates, Carlos A. Hurtado, Marcelo Mendoza: Query Clustering for Boosting Web Page Ranking. *Advances in Web Intelligence*. 164-175 (2004)

Travel Services

- [41] Athina Sismanidou, Miguel Palacios, Javier Tafur: Progress in airline distribution systems: The threat of new entrants to incumbent players. *Journal of Industrial Engineering and Management*. 251-272. (2009)
- [42] Nelson F. Granados, Robert J. Kauffman, Bradley King: The Emerging Role of Vertical Search Engines in Travel Distribution: A Newly-Vulnerable Electronic Markets Perspective. *Hawaii International Conference on System Sciences*. 389-399 (2008)
- [43] Hannes Werthner: Intelligent Systems in Travel and Tourism. *International joint conference on Artificial intelligence*. 1620-1625 (2003)
- [44] Bernard J. Jansen, Christopher C. Ciamacca, Amanda Spink: An Analysis of Travel Information Searching on the Web. *Journal of Information Technology and Tourism*. 10,2, 101-118 (2008)
- [45] Anindya Ghose, Panagiotis G. Ipeirotis, Beibei Li: Designing Ranking Systems for Hotels on Travel Search Engines to Enhance User Experience. *International Conference on Information Systems*. 113, 1-19 (2010)
- [46] Zheng Xiang, Daniel R. Fesenmaier: An Analysis of Two Search Engine Interface Metaphors for Trip Planning. *Journal of Information Technology and Tourism*. 7,2, 103-117 (2004)
- [47] Bernhard Kruepl, Wolfgang Holzinger, Yansen Darmaputra, Robert Baumgartner: A flight Metasearch engine with metamorph. *International conference on World Wide Web*. 1069-1070 (2009)
- [48] Nicole Mitsche: Understanding the Information Search Process within a Tourism Domain-specific Search Engine. *Information and Communication Technologies in Tourism*. 183-193 (2005)

Citation Analysis

- [49] Lokman I. Meho, Yvonne Rogers: Citation counting, citation ranking, and h-index of human-computer interaction researchers: A comparison of Scopus and Web of Science. *Journal of the American Society for Information Science and Technology*. 59,11, 1711-1726 (2008)
- [50] Judit Bar-Ilan: Which h-index? - A comparison of WoS, Scopus and Google Scholar. *Scientometrics*. 74,2, 257-271 (2008)

- [51] Judit Bar-Ilan: Informetrics at the beginning of the 21st century - A review. *Journal of Informetrics*. 2,1, 1-52 (2008)
- [52] Joeran Beel, Bela Gipp, and Erik Wilde. Academic Search Engine Optimization (ASEO): Optimizing Scholarly Literature for Google Scholar and Co. *Journal of Scholarly Publishing*. 41,2, 176-190 (2010)
- [53] Jorge E. Hirsch: An index to quantify an individual's scientific research output. *National Academy of Sciences of the United States of America*. 102, 46, 569-572 (2005)
- [54] Jöran Beel, Bela Gipp: Google Scholar's Ranking Algorithm: An Introductory Overview international society for scientometrics and infometrics. 439-446 (2009)
- [55] Dirk Lewandowski: Google Scholar as a tool for discovering journal articles in library and information science. *Online Information Review*. 34,2, 250-262 (2010)
- [56] William H. Walters: Comparative recall and precision of simple and expert searches in Google Scholar and eight other databases. *Libraries and the Academy*. 11,4, 971-1006 (2011)
- [57] Anne-Wil Harzing, Satu Alakangas: Google Scholar, Scopus and the Web of Science: a longitudinal and cross-disciplinary comparison. *Scientometrics*. 106,2, 787-804 (2016)
- [58] Bela Gipp, Jöran Beel, Christian Hentschel: Scienstein: A Research Paper Recommender System. *International Conference on Emerging Trends in Computing*. 309-315 (2009)

Ranking Algorithms

- [59] Ho Chung Wu, Robert Wing Pong Luk, Kam-Fai Wong, Kui-Lam Kwok: Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems*. 26,3,13, 1-37 (2008)
- [60] Charles Kemp, Kotagiri Ramamohanarao: Long-Term Learning for Web Search Engines. *Principles and Practice of Knowledge Discovery in Databases*. 263-274 (2002)
- [61] Peiquan Jin, Xiaowen Li, Hong Chen, Lihua Yue: CT-Rank: A Time-aware Ranking Algorithm for Web Search. *Journal of Convergence Information Technology*. 5,6, 99-111 (2010)

- [62] Taher H. Haveliwala: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*. 15,4, 784-796 (2003)
- [63] Sergey Brin, Lawrence Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*. 30,1,7, 107-117 (1998)
- [64] Jon M. Kleinberg: Authoritative Sources in a Hyperlinked Environment. *Symposium on Discrete Algorithms*. 668-677 (1998)
- [65] Ronny Lempel, Shlomo Moran: The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*. 33,1,6, 387-401 (2000)
- [66] Tao Wen-Xue, Zuo Wan-Li: Query-sensitive self-adaptable Web page ranking algorithm. *International Conference on Machine Learning and Cybernetics*. 413-418 (2003)
- [67] Ali Mohammad Zareh Bidoki, Nasser Yazdani: DistanceRank: An intelligent ranking algorithm for web pages. *Information Processing and Management*. 44,2, 877-892 (2008)
- [68] Renaud Delbru, Nickolai Toupikov, Michele Catasta, Giovanni Tummarello, Stefan Decker: Hierarchical Link Analysis for Ranking Web Data. *Extended Semantic Web Conference*. 2, 225-239 (2010)
- [69] Jian Jiao, Jun Yan, Haibei Zhao, Weiguo Fan: ExpertRank: An Expert User Ranking Algorithm in Online Communities. *New Trends in Information and Service Science*. 674-679 (2009)

Relevance Metrics

- [70] Mildrid Ljosland: Evaluation of Web search engines and the search for better ranking algorithms. *Special Interest Group on Information Retrieval, Workshop on Evaluation of Web Retrieval*. 1-6 (1999)
- [71] Michael D. Gordon, Praveen Pathak: Finding Information on the World Wide Web: The Retrieval Effectiveness of Search Engines. *Information Processing & Management*. 35,2, 141-180 (1999)
- [72] Liwen Vaughan: New measurements for search engine evaluation proposed and tested. *Information Processing & Management*. 40,4, 677-691 (2004)
- [73] Longzhuang Li, Yi Shang, Wei Zhang: Relevance Evaluation of Search Engines' Query Results. *World Wide Web Posters*. 1-2 (2001)

- [74] Amit Singhal, Marcin Kaszkiel: A case study in web search using TREC algorithms. *World Wide Web*. 708-716 (2001)
- [75] David Hawking, Nick Craswell, Paul B. Thistlewaite, Donna Harman: Results and Challenges in Web Search Evaluation. *Computer Networks*. 31,11,16, 1321-1330 (1999)
- [76] David Hawking, Nick Craswell, Peter Bailey, Kathleen Griffiths: Measuring Search Engine Quality. *Information Retrieval*. 4,1, 33-59 (2001)
- [77] Rashid Ali, Mirza Mohd. Sufyan Beg: An overview of Web search evaluation methods. *Computers & Electrical Engineering*. 37,6, 835-848 (2011)
- [78] Fazli Can, Rabia Nuray, Ayisigi B. Sevdik: Automatic performance evaluation of Web search engines. *Information Processes Management*. 40,3, 495-514 (2004)
- [79] Rashid Ali, Mirza Mohd, Sufyan Beg: Automatic Performance Evaluation of Web Search Systems using Rough Set based Rank Aggregation. *Intelligent Human Computer Interaction*. 344-358 (2009)
- [80] Jinbiao Hou: Research on Design of an Automatic Evaluation System of Search Engine. *Future Computer and Communication*. 16-18 (2009)
- [81] Abdur Chowdhury, Ian Soboroff: Automatic evaluation of world wide web search services. *Special Interest Group on Information Retrieval*. 421-422 (2002)
- [82] Mike Thelwall: Quantitative comparisons of search engine results. *Journal of the American Society for Information Science and Technology*. 59,11, 1702-1710 (2008)

Learning to Rank

- [83] Thorsten Joachims: Optimizing search engines using clickthrough data. *Knowledge Discovery and Data Mining*. 133-142 (2002)
- [84] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, Hang Li: Context-aware ranking in web search. *Special Interest Group on Information Retrieval*. 451-458 (2010)
- [85] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, Fernando Diaz: Towards recency ranking in web search. *Web Search and Data Mining*. 11-20 (2010)
- [86] Ananth Mohan, Zheng Chen, Kilian Q. Weinberger: Web-Search Ranking with Initialized Gradient Boosted Regression Trees. *Yahoo! Learning to Rank Challenge*. 77-89 (2011)

Recommender Systems

- [87] Lee Wei-Po, Liu Chih-Hung, Lu Cheng-Che: Intelligent agent-based systems for personalized recommendations in Internet commerce. *Expert Systems with Applications*. 22, 4, 275-284 (2002)
- [88] Dhoha Almazro, Ghadeer Shahatah, Lamia Albdulkarim, Mona Kherees, Romy Martinez, William Nzoukou: A Survey Paper on Recommender Systems. *CoRR abs/1006.5278* (2010)
- [89] Gediminas Adomavicius, Alexander Tuzhilin: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*. 17,6, 734-749 (2005)
- [90] Gediminas Adomavicius, Nikos Manouselis, YoungOk Kwon: Multi-Criteria Recommender Systems. *Recommender Systems Handbook*. 769-803 (2011)
- [91] Paolo Cremonesi, Antonio Tripodi, Roberto Turrin: Cross-Domain Recommender Systems. *International Conference on Data Mining Workshops*. 496-503 (2011)
- [92] Gunnar Schröder, Maik Thiele, Wolfgang Lehner: Setting Goals and Choosing Metrics for Recommender System Evaluations. *Conference on recommender systems*. 78-85 (2011)
- [93] Xiwang Yang, Yang Guo, Yong Liu, Harald Steck: A survey of collaborative filtering based social recommender systems. *Computer Communications*. 41, 1-10 (2014)

Neural Networks – Web Search

- [94] Franco Scarselli, Sweah Liang Yong, Marco Gori, Markus Hagenbuchner, Ah Chung Tsoi, Marco Maggini: Graph Neural Networks for Ranking Web Pages. *Web Intelligence*. 666-672 (2005)
- [95] Michael Chau, Hsinchun Chen: Incorporating Web Analysis Into Neural Networks: An Example in Hopfield Net Searching. *IEEE Transactions on Systems, Man, and Cybernetics. C*, 37,3, 352-358 (2007)
- [96] Sergio Bermejo, Javier Dalmau: Web Meta-search Using Unsupervised Neural Networks. *Artificial Neural Nets Problem Solving Methods*. 2, 711-718 (2003)
- [97] Bo Shu, Subhash C. Kak: A Neural Network-based Intelligent Metasearch Engine. *Information Sciences*. 120,1,4, 1-11 (1999)
- [98] Justin Boyan, Dayne Freitag, Thorsten Joachims: A Machine Learning Architecture for Optimizing Web Search Engines. *Association for the*

Advancement of Artificial Intelligence workshop on Internet-Based Information Systems. 1-8 (1996)

- [99] Shuo Wang, Kaiying Xu, Yong Zhang, Fei Li: Search Engine Optimization Based on Algorithm of BP Neural Networks. Computational Intelligence and Security. 390-394 (2011)
- [100] Yong S. Choi, Suk I. Yoo: Multi-agent Web Information Retrieval: Neural Network Based Approach. Advances in Intelligent Data Analysis. 499-512 (1999)

Neural Networks – Learning to Rank

- [101] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, Gregory N. Hullender: Learning to rank using gradient descent. International Conference Machine Learning. 89-96 (2005)
- [102] Matthew Richardson, Amit Prakash, Eric Brill: Beyond PageRank: machine learning for static ranking. World Wide Web. 707-715 (2006)
- [103] Eugene Agichtein, Eric Brill, Susan T. Dumais: Improving web search ranking by incorporating user behavior information. Special Interest Group on Information Retrieval. 19-26 (2006)
- [104] Leonardo Rigutini, Tiziano Papini, Marco Maggini, Franco Scarselli: SortNet: Learning to Rank by a Neural Preference Function. IEEE Transactions Neural Networks. 22,9, 1368-1380 (2011)
- [105] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, Hang Li: Learning to rank: from pairwise approach to listwise approach. International Conference on Machine Learning. 129-136 (2007)

Neural Networks – Recommender Systems

- [106] Smita Krishna Patil, Yogita Deepak Mane, Kanchan Rufus Dabre, Puneet Raman Dewan, Dhananjay R. Kalbande. An Efficient Recommender System using Collaborative Filtering Methods with K-separability Approach. International Journal of Engineering Research and Applications. 30-35 (2012)
- [107] Cheng-Chih Chang, Pei-Ling Chen, Fei-Rung Chiu, Yan-Kwang Chen: Application of neural networks and Kano's method to content recommendation in web personalization. Expert Systems with Applications. 36,3, 5310-5316 (2009)
- [108] Meehee Lee, Pyungseok Choi, Yongtae Woo: A Hybrid Recommender System Combining Collaborative Filtering with Neural Network. Adaptive Hypermedia and Adaptive Web-Based Systems. 531-534 (2002)

- [109] M. K. Kavitha Devi, R. Thirumalai Samy, S. Vinoth Kumar, P. Venkatesh: Probabilistic neural network approach to alleviate sparsity and cold start problems in collaborative recommender systems. *International Conference on Computational Intelligence and Computing Research*. 1-4 (2010)
- [110] Charalampos Vassiliou, Dimitris Stamoulis, Drakoulis Martakos, Sotiris Athanassopoulos: A recommender system framework combining neural networks & collaborative filtering. *International conference on Instrumentation, measurement, circuits and systems*. 285-290 (2006)
- [111] Kanokwan Kongsakun, Chun Che Fung: Neural Network Modeling for an Intelligent Recommendation System Supporting SRM for Universities in Thailand. *World Scientific and Engineering Academy and Society transactions on computers*. 2,11, 34-44 (2012)
- [112] Daniel Billsus, Michael J. Pazzani: Learning Collaborative Information Filters. *International Conference on Machine Learning*. 46-54 (1998)
- [113] Marko Krstic, Milan Bjelica: Context-aware personalized program guide based on neural network. *IEEE Transactions Consumer Electronics*. 58,4, 1301-1306 (2012)
- [114] Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, Alfonso Miola, Giuseppe Sansonetti: Context-aware movie recommendation based on signal processing and machine learning. *Challenge on Context-Aware Movie Recommendation*. 5-10 (2011)
- [115] Pao-Hua Chou, Pi-Hsiang Li, Kuang-Ku Chen, Menq-Jiun Wu: Integrating web mining and neural network for personalized e-commerce automatic service. *Expert Systems Applications*. 37,4, 2898-2910 (2010)

Deep Learning

- [116] Aliaksei Severyn, Alessandro Moschitti: Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. *Special Interest Group on Information Retrieval*. 373-382 (2015)
- [117] Baiyang Wang, Diego Klabjan: An Attention-Based Deep Net for Learning to Rank. *CoRR abs/1702.06106*. 1-8 (2017)
- [118] Li Deng, Xiaodong He, Jianfeng Gao: Deep stacking networks for information retrieval. *International Conference on Acoustics, Speech, and Signal Processing*. 3153-3157 (2013)

- [119] Hao Wang, Naiyan Wang, Dit-Yan Yeung: Collaborative Deep Learning for Recommender Systems. Special Interest Group on Knowledge Discovery and Data Mining. 1235-1244 (2015)
- [120] Ali Mamdouh Elkahky, Yang Song, Xiaodong He: A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. World Wide Web. 278-288 (2015)

G – Networks

- [121] Erol Gelenbe: Réseaux stochastiques ouverts avec clients négatifs et positifs, et réseaux neuronaux. Comptes-Rendus. Académie des Sciences de Paris. 309, II, 979-982 (1989)
- [122] Erol Gelenbe, Salvatore Tucci : Performances d'un systeme informatique dupliqué. Comptes-Rendus. Académie des Sciences. 312, II, 27-30 (1991)
- [123] Erol Gelenbe: Queueing networks with negative and positive customers. Journal of Applied Probability. 28, 656-663 (1991)
- [124] Erol Gelenbe, Rolf Schassberger: Stability of product form G-Networks. Probability in the Engineering and Informational Sciences. 6, 271-276 (1992)
- [125] Erol Gelenbe: G-Networks with signals and batch removal. Probability in the Engineering and Informational Sciences. 7, 335-342 (1993)
- [126] Erol Gelenbe: G-networks with triggered customer movement. Journal of Applied Probability. 30,3, 742-748 (1993)
- [127] Jean-Michel Fourneau, Erol Gelenbe: G-networks with resets. Special Interest Committee on Measurement and Evaluation. 29,3, 19-20 (2001)
- [128] Jean-Michel Fourneau, Erol Gelenbe: Flow equivalence and stochastic equivalence in G-networks. Computer Management Science. 1,2, 179-192 (2004)
- [129] Jean-Michel Fourneau, Erol Gelenbe, Rina Suros: G-Networks with Multiple Classes of Negative and Positive Customers. Theoretical Computer Science. 155,1, 141-156 (1996)
- [130] Erol Gelenbe, Ali Labed: G-networks with multiple classes of signals and positive customers. European Journal of Operational Research. 108,2, 293-305 (1998)
- [131] Erol Gelenbe: G-networks: a unifying model for neural and queueing networks. Annals of Operations Research. 48, 5, 433-461 (1994)
- [132] Erol Gelenbe: The first decade of G-networks. European Journal of Operational Research. 126, 2, 231-232 (2000)

- [133] Erol Gelenbe, Peter Glynn, Karl Sigman: Queues with negative arrivals. *Journal of Applied Probability*. 28, 245-250 (1991)
- [134] Erol Gelenbe, Hadas Shachnai: On G-networks and resource allocation in multimedia systems. *European Journal of Operational Research*. 126,2, 308-318 (2000)

Random Neural Network

- [135] Erol Gelenbe: Random Neural Networks with Negative and Positive Signals and Product Form Solution. *Neural Computation*. 1,4, 502-510 (1989)
- [136] Erol Gelenbe: Stability of the Random Neural Network Model. *Neural Computation*. 2,2, 239-247 (1990)
- [137] Erol Gelenbe: Learning in the Recurrent Random Neural Network. *Neural Computation*. 5,1, 154-164 (1993)
- [138] Erol Gelenbe, Jean-Michel Fourneau: Random Neural Networks with Multiple Classes of Signals. *Neural Computation*. 11,4, 953-963 (1999)
- [139] Erol Gelenbe, Khaled Hussain: Learning in the multiple class random neural network. *IEEE Transactions Neural Networks*. 13,6, 1257-1267 (2002)
- [140] Erol Gelenbe, Andreas Stafylopatis, Aristidis Likas: Associative memory operation of the random network model. *International Conference on Artificial Neural Networks*. 307-312 (1991)
- [141] Erol Gelenbe, Zhi-Hong Mao, Yan-Da Li: Function approximation with spiked random networks. *IEEE Transactions Neural Networks*. 10,1, 3-9 (1999)
- [142] Erol Gelenbe, Zhi-hong Mao, Yan-da Li: Function approximation by random neural networks with a bounded number of layers. *Differential Equations and Dynamical Systems*. 12, 143-170 (2004)
- [143] Erol Gelenbe, Stelios Timotheou: Random Neural Networks with Synchronized Interactions. *Neural Computation*. 20,9, 2308-2324 (2008)
- [144] Erol Gelenbe, Stelios Timotheou: Synchronized Interactions in Spiked Neuronal Networks. *Computation Journal*. 51,6, 723-730 (2008)
- [145] Erol Gelenbe, Yongha Yin: Deep learning with random neural networks. *International Joint Conference on Neural Networks*. 1633-1638 (2016)

Optimization

- [146] Erol Gelenbe, Frederic Batty: Minimum cost graph covering with the random neural network. *Computer Science and Operations Research*. 139-147 (1992)

- [147] Erol Gelenbe, Vassilada Koubi, Ferhan Pekergin: Dynamical random neural network approach to the traveling salesman problem. *Systems, Man and Cybernetics*. 630-635 (1993)
- [148] Jose Aguilar, Erol Gelenbe: Task Assignment and Transaction Clustering Heuristics for Distributed Systems. *Informatics and Computer Science*. 199-219 (1996)
- [149] Erol Gelenbe, Anoop Ghanwani, Vijay Srinivasan: Improved Neural Heuristics for Multicast Routing. *Selected Areas in Communications*. 15,2, 147-155 (1997)

Image and Video Compression

- [150] Volkan Atalay, Erol Gelenbe, Nese Yalabik: The Random Neural Network Model for Texture Generation. *International Journal of Pattern Recognition and Artificial Intelligence*. 6,1, 131-141 (1992)
- [151] Erol Gelenbe, Khaled Hussain, Hossam Abdelbaki : Random neural network texture model. *Applications of Artificial Neural Networks in Image Processing*. 104, 1-8 (2000)
- [152] Erol Gelenbe, Yutao Feng, Ranga Krishnan: Neural network methods for volumetric magnetic resonance imaging of the human brain. *Proceedings of the IEEE*. 84, 10, 1488-1496 (1996)
- [153] Erol Gelenbe, Taşkın Koçak, Rong Wang: Wafer surface reconstruction from top-down scanning electron microscope images. *Microelectronic Engineering*. 75,2, 216-233 (2004)
- [154] Erol Gelenbe, Yutao Feng: Image content classification methods, systems and computer programs using texture patterns. *U.S. Patent* 5,995,651 (1999)
- [155] Erol Gelenbe, Mert Sungur, Christopher Cramer, Pamir Gelenbe: Traffic and Video Quality with Adaptive Neural Compression. *Multimedia Systems*. 4,6, 357-369 (1996)
- [156] Christopher Eric Cramer, Erol Gelenbe, Hakan Bakircioglu: Low bit rate video compression with neural networks and temporal sub-sampling. *Proceedings of the IEEE*. 84,10, 1529-1543 (1996)
- [157] Christopher Cramer, Erol Gelenbe: Video quality and traffic QoS in learning-based subsampled and receiver-interpolated video sequences. *IEEE Journal on Selected Areas in Communications*. 18,2, 150-167 (2000)

Cognitive Packet Networks

- [158] Erol Gelenbe, Zhiguang Xu, Esin Seref: Cognitive Packet Networks. International Conference on Tools with Artificial Intelligence. 47-54 (1999)
- [159] Erol Gelenbe, Ricardo Lent, Zhiguang Xu: Measurement and performance of a cognitive packet network. Computer Networks. 37,6, 691-701 (2001)
- [160] Erol Gelenbe, Ricardo Lent, Zhiguang Xu: Towards networks with cognitive packets. Performance and QoS of Next Generation Networking. 3-17 (2002)
- [161] Erol Gelenbe, Ricardo Lent, Alfonso Montuori, Zhiguang Xu: Cognitive Packet Networks: QoS and Performance. Modelling, Analysis, and Simulation On Computer and Telecommunication Systems. 3-12 (2002)
- [162] Lan Wang, Erol Gelenbe: An Implementation of Voice Over IP in the Cognitive Packet Network. International Symposium on Computer and Information Sciences. 33-40 (2014)
- [163] Lan Wang, Erol Gelenbe: Demonstrating Voice over an Autonomic Network. International Conference on Autonomic Computing. 139-140 (2015)
- [164] Lan Wang, Erol Gelenbe: Real-Time Traffic over the Cognitive Packet Network. Computer Networks. 3-21 (2016)
- [165] Erol Gelenbe, Ricardo Lent, Arturo Nunez: Smart WWW traffic balancing. Communications Society Workshop on IP Operations and Management. 15-22 (2003)
- [166] Antoine Desmet, Erol Gelenbe: A Parametric Study of CPN's Convergence Process. International Symposium on Computer and Information Sciences. 13-20 (2014)
- [167] Erol Gelenbe, Ricardo Lent: Power-aware ad hoc cognitive packet networks. Ad Hoc Networks. 2,3, 205-216 (2004)
- [168] Ricardo Lent, Farhad Zonoozi: Power control in ad hoc cognitive packet networks. Texas Wireless Symposium. 65-69 (2005)
- [169] Huibo Bi, Erol Gelenbe: A Cooperative Emergency Navigation Framework Using Mobile Cloud Computing. International Symposium on Computer and Information Sciences. 41-48 (2014)
- [170] Erol Gelenbe: Cognitive Packet Network. U.S. Patent 6,804,201 (2004)

Self Aware Networks

- [171] Erol Gelenbe, Michael Gellman, Pu Su: Self-Awareness and Adaptability for Quality of Service. International Symposium on Computers and Communications. 3-9 (2003)
- [172] Erol Gelenbe, Ricardo Lent, Arturo Nunez: Self-aware networks and QoS. Proceedings of the IEEE. 92,9, 1478–1489 (2004)
- [173] Erol Gelenbe: Steps toward self-aware networks. Communications of the ACM. 52,7, 66-75 (2009)
- [174] Erol Gelenbe: A Software Defined Self-Aware Network: The Cognitive Packet Network. Network Cloud Computing and Applications. 9-14 (2014)

Software Defined Networks

- [175] Erol Gelenbe: A Software Defined Self-Aware Network: The Cognitive Packet Network. Semantics Knowledge and Grids. 1-5 (2013)
- [176] Erol Gelenbe, Zarina Kazhmaganbetova: Cognitive Packet Network for Bilateral Asymmetric Connections. IEEE Transactions on Industrial Informatics. 10,3, 1717-1725 (2014)
- [177] Frédéric François, Erol Gelenbe: Towards a cognitive routing engine for software defined networks. International Conference on Communications. 1-6 (2016)
- [178] Frédéric François, Erol Gelenbe: Optimizing Secure SDN-Enabled Inter-Data Centre Overlay Networks through Cognitive Routing. Modeling, Analysis, and Simulation On Computer and Telecommunication Systems. 283-288 (2016)
- [179] Olivier Brun, Lan Wang, Erol Gelenbe: Big Data for Autonomic Intercontinental Overlays. IEEE Journal on Selected Areas in Communications. 34,3, 575-583 (2016)

Smart Routing

- [180] Pu Su, Michael Gellman: Using adaptive routing to achieve quality of service. Performance Evaluation. 57,2, 105–119 (2004)
- [181] Erol Gelenbe, Michael Gellman, Ricardo Lent, Peixiang Liu, Pu Su: Autonomous Smart Routing for Network QoS. International Conference on Autonomic Computing. 232-239 (2004)
- [182] Erol Gelenbe, Peixiang Liu, Jeremy LainLaine: Genetic Algorithms for Route Discovery. IEEE Transactions on Systems, Man and Cybernetics. 36,6, 1247–1254 (2006)

- [183] Erol Gelenbe, Peixiang Liu, Jeremy LainLaine: Genetic algorithms for autonomic route discovery. *Distributed Intelligent Systems: Collective Intelligence and Its Applications*. 371–376 (2006)
- [184] Ricardo Lent, Peixiang Liu: Searching for low latency routes in CPN with reduced packet overhead. *International Symposium on Computer and Information Sciences*. 63-72 (2005)
- [185] Erol Gelenbe, Peixiang Liu: QoS and Routing in the Cognitive Packet Network. *World of Wireless, Mobile and Multimedia Networks*. 517-521 (2005)
- [186] Michael Gellman, Peixiang Liu: Random Neural Networks for the Adaptive Control of Packet Networks. *International Conference on Artificial Neural Networks*. 1, 313-320 (2006)
- [187] Erol Gelenbe, Michael Gellman: Oscillations in a Bio-Inspired Routing Algorithm. *Mobile Ad-hoc and Sensor Systems*. 1-7 (2007)
- [188] Laurence A. Hey: Reduced complexity algorithms for cognitive packet network routers. *Computer Communications*. 31,16, 3822-3830 (2008)
- [189] Laurence A. Hey, Peter Y. K. Cheung, Michael Gellman: FPGA Based Router for Cognitive Packet Networks. *Field-Programmable Technology*. 331-332 (2005)
- [190] Peixiang Liu, Erol Gelenbe: Recursive Routing in the Cognitive Packet Network. *Testbeds and Research Infrastructures for the Development of Networks and Communities*. 1-6 (2007)

Cybersecurity

- [191] Erol Gelenbe, Michael Gellman, George Loukas: Defending networks against denial of service attacks. *Optics Photonics in Security and Defense Unmanned/Unattended Sensors and Sensor Networks*. 5611, 233–243 (2004)
- [192] Erol Gelenbe, Michael Gellman, George Loukas: An Autonomic Approach to Denial of Service Defense. *World of Wireless Mobile and Multimedia Networks*. 537-541 (2005)
- [193] Erol Gelenbe, George Loukas: A self-aware approach to denial of service defense. *Computer Networks*. 51,5, 1299-1314 (2007)
- [194] Georgia Sakellari, Maurizio D'Arienzo, Erol Gelenbe: Admission Control in Self Aware Networks. *Global Telecommunications Conference*. 1-5 (2006)
- [195] Georgia Sakellari, Erol Gelenbe, Maurizio D'Arienzo: Admission of Packet Flows in a Self-Aware Network. *Mobile Ad-hoc and Sensor Systems*. 1-6 (2007)

- [196] Erol Gelenbe, Georgia Sakellari, Maurizio D'Arienzo: Admission of QoS aware users in a smart network. *ACM Transactions on Autonomous and Adaptive Systems*. 3,1,4, 1-28 (2008)
- [197] Erol Gelenbe, Georgia Sakellari, Maurizio D'Arienzo: Controlling Access to Preserve QoS in a Self-Aware Network. *Self-Adaptive and Self-Organizing Systems* . 205-213 (2007)
- [198] Georgios Loukas, Gülay Öke: A Biologically Inspired Denial of Service Detector Using the Random Neural Network. *Mobile Ad-hoc and Sensor Systems*. 1-6 (2007)
- [199] George Loukas, Gulay Oke: Likelihood ratios and recurrent random neural networks in detection of denial of service attacks. *Symposium Performance Evaluation of Computer and Telecommunication System*. 1-8 (2007)
- [200] Gülay Öke, George Loukas, Erol Gelenbe: Detecting Denial of Service Attacks with Bayesian Classifiers and the Random Neural Network. *Fuzzy Systems IEEE*. 1-6 (2007)
- [201] Gülay Öke, Georgios Loukas: A Denial of Service Detector based on Maximum Likelihood Detection and the Random Neural Network. *Computation Journal*. 50,6, 717-727 (2007)

Gene Regulatory Networks

- [202] Erol Gelenbe: Steady-state solution of probabilistic gene regulatory networks. *Physical review*. 76,3, 31903, 1-8 (2007)
- [203] Haseong Kim, Erol Gelenbe: Reconstruction of Large-Scale Gene Regulatory Networks Using Bayesian Model Averaging. *Bioinformatics and Biomedicine*. 202-207 (2011)
- [204] Haseong Kim, Taesung Park, Erol Gelenbe: Identifying disease candidate genes via large-scale gene network analysis. *International Journal of Data Mining and Bioinformatics*. 10,2, 175-188 (2014)
- [205] Haseong Kim, Rengul Atalay, Erol Gelenbe: G-Network Modelling Based Abnormal Pathway Detection in Gene Regulatory Networks. *International Symposium on Computer and Information Sciences*. 257-263 (2011)
- [206] Haseong Kim, Erol Gelenbe: Stochastic Gene Expression Modeling with Hill Function for Switch-Like Gene Responses. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 9,4, 973-979 (2012)

Web Search

- [207] Will Serrano, Erol Gelenbe: An Intelligent Internet Search Assistant Based on the Random Neural Network. Artificial Intelligence Applications and Innovations. 141-153 (2016)
- [208] Will Serrano: A Big Data Intelligent Search Assistant Based on the Random Neural Network. International Neural Network Society Conference on Big Data. 254-261 (2016)
- [209] Will Serrano, Erol Gelenbe: Intelligent Search with Deep Learning Clusters. Intelligent Systems Conference. 254-261 (2017)
- [210] Will Serrano, Erol Gelenbe: The Deep Learning Random Neural Network with a Management Cluster. International Conference on Intelligent Decision Technologies, 185-195 (2017)

Relevance Feedback

- [211] Diane Kelly, Jaime Teevan: Implicit feedback for inferring user preference: a bibliography. ACM Special Interest Group of Information Retrieval. 37,2 (2003)
- [212] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, Thomas White: Evaluating Implicit Measures to Improve Web Search. Journal ACM Transactions on Information Systems. 23, 2, 147-168 (2005)
- [213] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay: Accurately Interpreting Clickthrough Data as Implicit Feedback. Special Interest Group of Information Retrieval. 51, 14-11 (2017)
- [214] Filip Radlinski and Thorsten Joachims: Query chains, learning to rank from implicit feedback. ACM Special Interest Group on Knowledge discovery in data mining. 239-248 (2005)
- [215] Gawesh Jawaheer, Martin Szomszor, and Patty Kostkova: Comparison of implicit and explicit feedback from an online music recommendation service. International Workshop on Information Heterogeneity and Fusion in Recommender Systems. 47-51 (2010)
- [216] Ryen W. White, Ian Ruthven, Joemon M. Jose: The Use of Implicit Evidence for Relevance Feedback in Web Retrieval. European Conference on Information Retrieval, Advances in Information Retrieval. 93-109 (2002)
- [217] Douglas W. Oard and Jinmook Kim: Implicit Feedback for Recommender Systems. Proceedings of the Association for the Advancement of Artificial Intelligence Workshop on Recommender Systems. 81-83 (1998)

Datasets

- [218] Kavita Ganesan and Cheng Xiang Zhai: Opinion-based entity ranking. *Information retrieval*. 15, 2, 116—150 (2012)
- [219] Julian McAuley, Christopher Targett, Qinfeng Shi and Anton van den Hengel: Image-Based Recommendations on Styles and Substitutes. *Conference on Research and Development in Information Retrieval*. 43-52.
- [220] Julian McAuley, Rahul Pandey and Jure Leskovec. Inferring Networks of Substitutable and Complementary Products. *International Conference on Knowledge Discovery and Data Mining*. 785-794.

Appendix

A ISA Screen shots

Internet Search Assistant

Please introduce each of your search dimension values on each of the search boxes below.

Learning Type:

It may take some time while retrieving your results; please be patient for the next page.

Please comments or bugs; search your email and I will get back to you.

Figure 83: ISA Interface

ISA-RNN

space exploration research

Select your relevant results

6: Title: 'Beyond the Stars [VHS]' - Description: "A high school student obsessed with space exploration meets a troubled and withdrawn; retired astronaut who once explored the moon. The boy [Rating: 0.9 - Potential 0.4928473446991965]

1: Title: 'Spaceballs' - Description: "The farce is with you in this "uproarious salute to science fiction" (The Hollywood Reporter) that teams comedy legend Mel Brooks with an all-star cast of cutups including John Candy (Splash); Rick Moranis (Ghostbusters) and Bill Pullman (Ruthless People); When the evil Dark Helmet (Moranis) attempts to steal all the air from planet Druidia; a determined Druid Princess (Daphne Zuniga); a clueless rogue (Pullman) and a half-man-half-dog creature who [Rating: 0.8757255936675462 - Potential 0.32724814063556457]

2: Title: 'Office Space [VHS]' - Description: "Ever spend eight hours in a "Productivity Bin"? Ever had worries about layoffs? Ever had the urge to demolish a temperamental printer or fax machine? Ever had to endure a snarky; condescending boss? Then Office Space should hit pretty close to home for you. Peter (Ron Livingston) spends the day doing stupefyingly dull computer work in a cubicle. He goes home to an apartment sparsely furnished by IKEA and Target; then starts for a maddening commute to work again in the morning. His coworkers in the cube farm are an annoying lot; his boss is a snide; patronizing jerk; and his days are consumed with tedium. In desperation; he turns to career hypnotherapy; but when his hypno-induced relaxation takes hold; there [Rating: 0.9408829174664107 - Potential 0.3263250109505037]

3: Title: 'Lost in Space - Description: "Originally conceived as "The Swiss Family Robinson In Space" and airing from 1965 to 1968 this light-hearted adventure is a true sci-fi classic with an enormous following of devoted fans. Relive the fun and excitement in this specially selected series of the very best episodes. Volume 7: The Keeper Part 1. The Robinsons make the acquaintance of "The Keeper" (Michael Rennie) -- a member of an advanced alien race who collects specimens of creatures from around the galaxy. Urbane and courteous; the Robinsons accept him as a friend -- until they learn that he intends to include Will and Penny in his collection. The family makes it clear the children will never be surrendered; but it seems that The Keeper won [Rating: 1.0 - Potential 0.3258673693456302]

4: Title: 'Nova: Terror in Space [VHS]' - Description: "Despite the sensationalist title; this is a fantastically interesting and simultaneously frightening look at living in space. The melodramatic title only hints at the excitement and fascinating danger the astronauts on the Mir space station survived. From the terrifying reality of a fire in the confines of the station to the hidden dangers of failed experimental docking maneuvers; the challenges of the aging space station threaten the lives of the astronauts. Novadocuments this in an elegantly smart and unpretentious manner with a combination of interviews from the astronauts; real footage; and simulations. This video also serves to highlight the communication intricacies involved in guaranteeing the safety of both Russian and American astronauts in a hostile environment. As usual; Novadoc does a splendid job of entertaining; informing; and truly providing mind-altering television.--Amanda Powter [Rating: 1.0 - Potential 0.3246346555325904]

5: Title: 'Star Trek - Deep Space Nine - Description: "The "frontier town" aspect of Deep Space Nine always promised more conflict than previous Star Trek series; and "Dramatis Personae" presents a what-if scenario that pits all of the main characters against each other in a cauldron of deceit; betrayal; and conspiracy; brought on by a mysterious; telepathic energy that brings out the worst in everyone involved. At one extreme is Kira; who plots a Bajoran takeover of the station; and at the other is Dax; who sits passively in Ops; content to indulge dreamy reveries while chaos explodes around her. Even as a manipulated "play" caused by an alien force; this scenario allows the cast of DS9 to cut loose and shatter the barriers of proper decorum; and the result is a rare case of Star Trek anarchy; in stark contrast to the dutiful efficiency of a "normal" Federation crew. Edgy and fun; this is Trek with its hair down; leaving everyone embarrassed but admirably unapologetic.--Jeff Shannon [Rating: 0.7333333333333333 - Potential 0.3246231155778894]

6: Title: 'Star Trek - Deep Space Nine - Description: "Ensign Melora Pazlar (Daphne Ashbrook) arrives at the station on a mission to chart the Gamma Quadrant. Melora is from a planet with very low gravity and must use a wheelchair to get around in "normal" conditions. Though she resents any special treatment; a smitten Bashir (Siddiq El Fadil) outfits her quarters with a gravity device which allows her to "fly" around. Meanwhile; Quark (Armin Shimerman) welcomes Fallin Kot (Peter Crombie); an old acquaintance who announces that he intends to kill the Ferengi. Odo (Rene Auberjonois) accuses Quark he will not allow that to happen. When Melora is injured in a fall; Bashir develops a new technology which allow her to walk without crutches. But it will also prevent her from flying. While she struggles with the decision; Kot ignores Odo. [Rating: 0.9 - Potential 0.32387692387692384]

7: Title: 'Space Raiders [VHS]' - Description: "A plucky 10-year-old blasts off into a Star Wars world of intergalactic desperadoes; alien mercenaries and starship battles. Year: 1983 Director: Howard R. Cohen Starring: Vince Edwards; David Mendenhall; Patty Pease [Rating: 0.55 - Potential 0.3220125786163522]

8: Title: 'Innerspace [VHS]' - Description: "Distractingly loose but clever; this 1987 comedy by Joe Dante (Gremlins; Matinee) stars Martin Short as a hypochondriac and Dennis Quaid as a miniaturized test pilot who is accidentally injected into him. Taking a page or two (or a hundred) from the classic science fiction movie Fantastic Voyage; Innerspace details the adventures of traveling through the human body [Rating: 0.8313253012048193 - Potential 0.32186732186732187]

9: Title: 'Lost in Space - Description: "Originally conceived as "The Swiss Family Robinson In Space" and airing from 1965 to 1968 this light-hearted adventure is a true sci-fi classic with an enormous following of devoted fans. Relive the fun and excitement in this specially selected series of the very best episodes. "The Keeper" (Michael Rennie); a powerful alien from an advanced civilization; collects creatures from all over the galaxy. When Dr. Smith tries to steal The Keeper [Rating: 1.0 - Potential 0.32169746748802186]

10: Title: 'Star Trek - Deep Space Nine - Description: "After arriving at Deep Space Nine; B'udan (Steven James Carver); a Bajoran murderer recently released from prison; tangles with Security Chief Odo (Rene Auberjonois). When B'udan is murdered in a holosuite; the evidence points to a shape-shifter as the guilty party. Odo; who had vowed to remove B'udan from Deep Space Nine at any cost; is accused of the crime. Soon an angry group of Bajorans comes looking for justice and Commander Sisko (Avery Brooks) is forced to relieve Odo of duty. Although tests on a strand of DNA found at the crime scene promise to reveal the identity of the real killer; an impatient and violent mob chases down Odo and tries to take the law into its own hands. [Rating: 0.56 - Potential 0.3216216216216216]

11: Title: 'Space Jam' - Description: "Director Joe Pytko apparently has little to say about the making of Space Jam; he leaves most of the commentary to voice actors Billy West and D. Bradley Baker (who took over some of Mel Blanc [Rating: 0.8973568281938326 - Potential 0.3205128205128205]

12: Title: 'Flash Gordon - Space Soldiers' - Description: "Buster Crabbe stars as Flash Gordon in this classic 1930s serial presented for the first time on DVD. Humanity is doomed to destruction! A distant planet has broken its orbit and is headed straight toward the Earth.

Figure 84: ISA Result presentation

B Unsupervised Evaluation

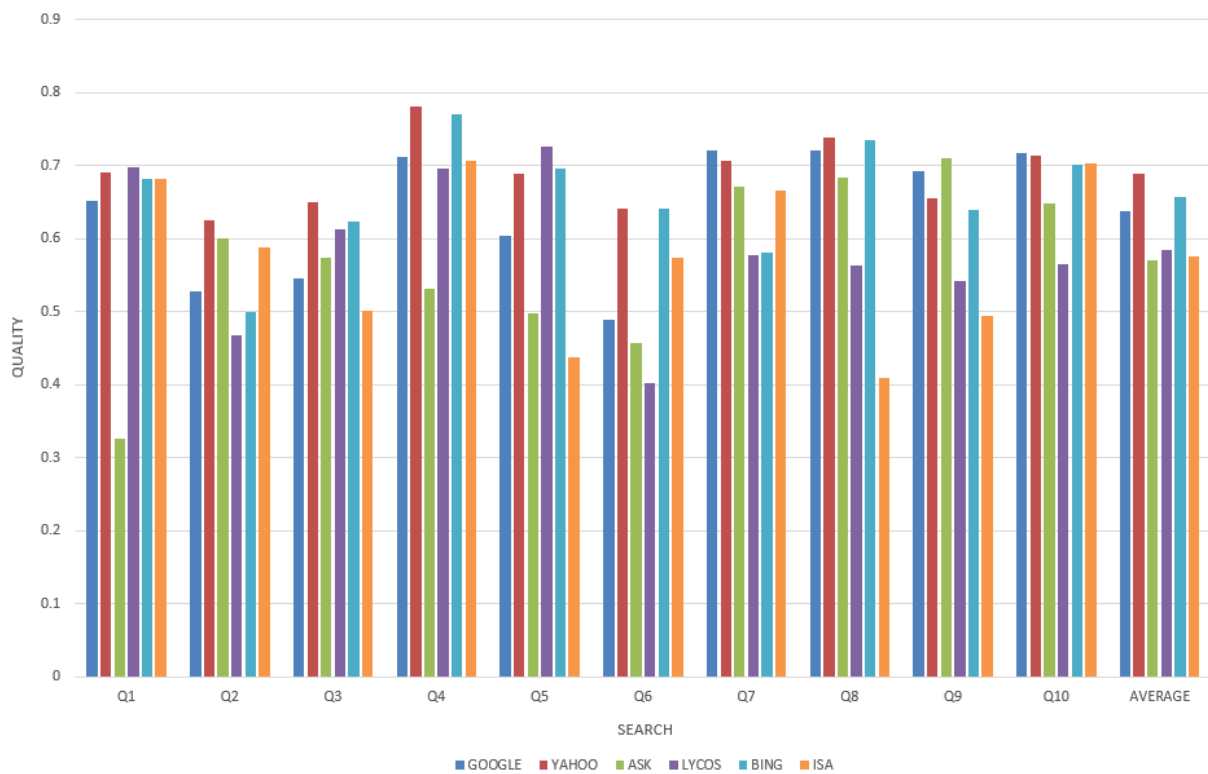


Figure 85: Web Search Evaluation

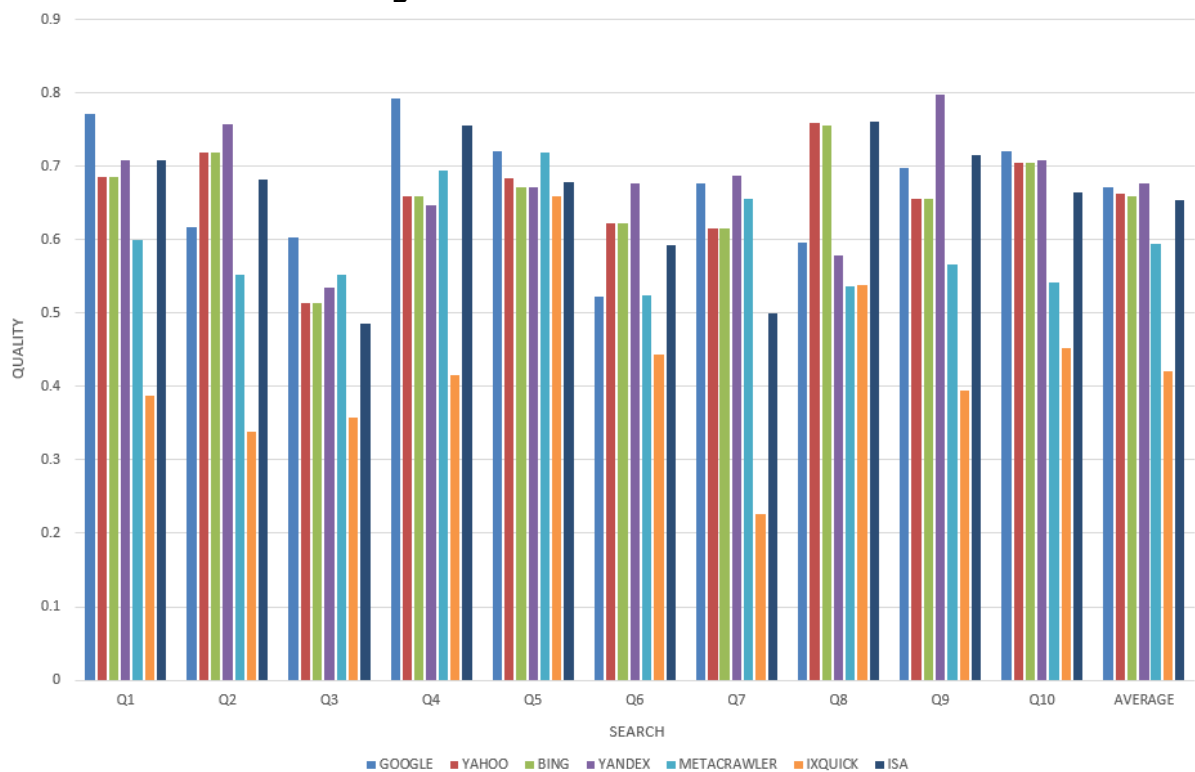


Figure 86: Meta Search Evaluation

C User Evaluation – First Iteration

Google Search – Result Cost Function

Results retrieved	Results selected	Precision	Quality Google	Quality ISA	Improvement ISA
Query Q01: Toulouse flight cheap					
20	04	0.2000	0.2619	0.2238	-14.55%
Query Q02: cross channel transport minibus					
09	01	0.1111	0.0888	0.1333	50.11%
Query Q03: accessibility image glare					
10	02	0.2000	0.2000	0.3111	55.55%
Query Q04: UK visa requirements					
09	03	0.3333	0.4000	0.5333	33.33%
Query Q05: flower market London					
17	04	0.2353	0.2745	0.3660	33.33%
Query Q06: pub darts London cheap					
29	14	0.4828	0.6552	0.5747	-12.29%
Query Q07: weather London UK					
20	04	0.2000	0.2381	0.3381	42.00%
Query Q08: champions league live pub London					
30	12	0.4000	0.5097	0.5118	0.41%
Query Q09: movie times east London thor					
10	05	0.5000	0.6545	0.6545	0.00%
Query Q10: free French language meetups London					
10	05	0.5000	0.6545	0.6363	-2.78%
Query Q11: sports bars Paris					
10	05	0.5000	0.6363	0.6727	5.72%
Query Q12: imperial college closure days					
17	03	0.1765	0.2941	0.2680	-8.87%
Query Q13: UK bank holidays					
09	01	0.1111	0.0222	0.2000	809.09%
Query Q14: Best burgers London					
19	07	0.3684	0.4316	0.3368	-21.96%
Query Q15: events November London					
20	03	0.1500	0.1095	0.2471	125.66%
Query Q16: online courses python					
20	02	0.1000	0.0952	0.1857	95.06%



Figure 87: Google Search Result Cost Function

Results retrieved	Results selected	Precision	Quality Google	Quality ISA	Improvement ISA
Query Q01: cheap flights Caribbean					
20	08	0.4000	0.3429	0.5238	52.76%
Query Q02: vegetarian restaurant central London					
10	06	0.6000	0.6727	0.5454	-18.92%
Query Q03: Varanasi India main mosque					
26	03	0.1154	0.1823	0.2906	59.41%
Query Q04: winter wonderland ride prices					
19	02	0.1053	0.1579	0.1789	13.30%
Query Q05: new restaurant opening hackney					
28	05	0.1786	0.2069	0.2906	40.45%
Query Q06: ballet classes London					
20	20	1.0000	1.0000	1.0000	0.00%
Query Q07: recipe lemon meringue pie					
17	17	1.0000	1.0000	1.0000	0.00%
Query Q08: musicals Shaftesbury avenue					
20	20	1.0000	1.0000	1.0000	0.00%
Query Q09: Vegetarian Restaurant Islington					
18	11	0.6111	0.6959	0.6784	-2.51%
Query Q10: Peterborough Brentford Football Result					
19	03	0.1579	0.2368	0.1684	-28.89%
Query Q11: Insurance iPhone Cheap					
20	15	0.7500	0.8905	0.9095	2.13%
Query Q12: Artist residence opportunity Cambridge					
20	05	0.2500	0.3429	0.2905	-15.28%
Query Q13: research funded fine art opportunity					
19	06	0.3158	0.2842	0.4579	61.12%

Query Q14: material culture research art Cambridge					
20	06	0.3000	0.2762	0.3810	37.94%
Query Q15: multiple sclerosis research blog London					
20	06	0.3000	0.4238	0.3286	-22.46%
Query Q16: Roman Baths Bath Opening Hours					
19	09	0.4737	0.5631	0.5263	-6.54%
Query Q17: Brompton folding bike second hand Hackney					
20	05	0.2500	0.3333	0.3095	-7.14%
Query Q18: pubs near Waterloo					
18	03	0.1667	0.1754	0.1406	-19.84%
Query Q19: second hand furniture markets London					
19	06	0.3158	0.3263	0.3947	20.96%
Query Q20: late night museums London					
15	05	0.3333	0.1416	0.3416	141.24%

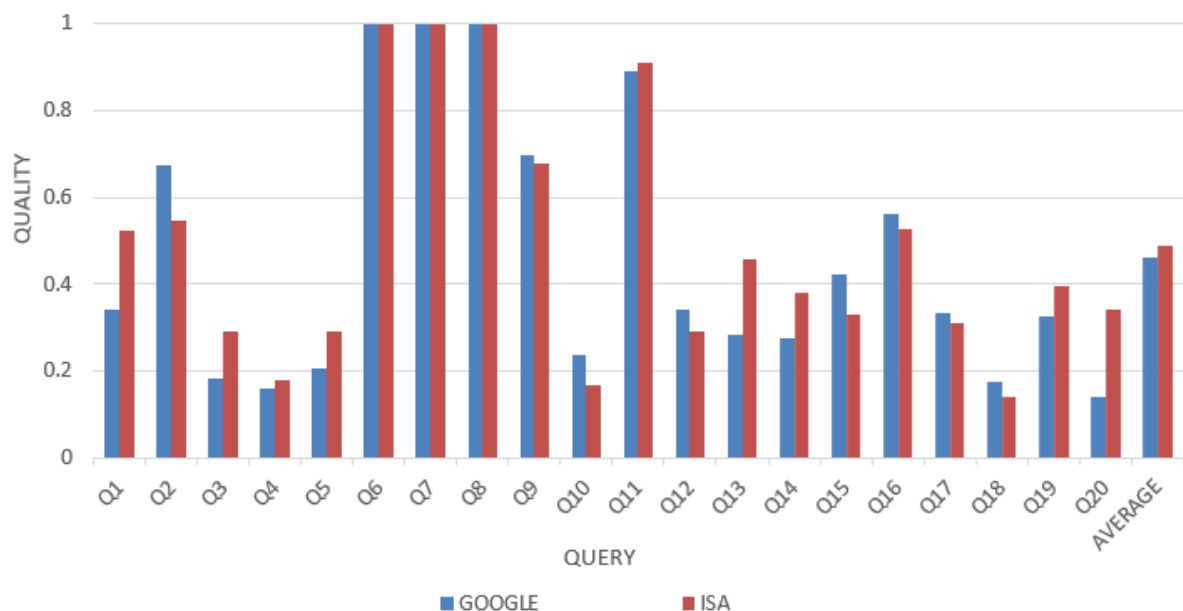


Figure 88: Google Search Result Cost Function

Web Search – Result Cost Function

Query Q01: movie times east London thor					
Results retrieved	Results displayed	Precision /Quality Google	Precision /Quality Ask	Precision /Quality Lycos	Precision /Quality ISA
10	26	0.4000 0.4118	0.8000 0.7272	0.5000 0.5091	0.6538 0.7692
Query Q02: free French language meetups					
Results retrieved	Results displayed	Precision/Quality Google		Precision/Quality Bing	Precision/Quality ISA
10	18	0.4000 0.4909		0.8000 0.8727	0.6667 0.7017

Appendix

Query Q03: Annapurna trek tour operator					
Results retrieved	Results displayed	Precision/Quality Google	Precision/Quality Lycos	Precision/Quality Bing	Precision/Quality ISA
20	54	0.7000 0.7429	0.5000 0.5524	0.5000 0.4857	0.6296 0.6189
Query Q04: champions league live pub London					
Results retrieved	Results displayed	Precision/Quality Google	Precision/Quality Yahoo	Precision/Quality Bing	Precision/Quality ISA
30	66	0.4333 0.4151	0.2333 0.2903	0.1333 0.1484	0.3636 0.4283
Query Q05: flowers watering system					
Results retrieved	Results displayed	Precision/Quality Google	Precision/Quality Yahoo	Precision/Quality Bing	Precision/Quality ISA
20	44	0.0000 0.0000	0.1500 0.1429	0.0000 0.0000	0.0682 0.1172
Query Q06: cheap opera tickets London					
Results retrieved	Results displayed	Precision/Quality Google	Precision/Quality Yahoo	Precision/Quality ISA	
20	40	0.1000 0.1286	0.2500 0.2952	0.1250 0.2404	
Query Q07: London fish market opening hours					
Results retrieved	Results displayed	Precision/Quality Google	Precision/Quality Yahoo	Precision/Quality ISA	
20	33	0.1500 0.0845	0.1000 0.1046	0.1515 0.2282	
Query Q08: Hampstead Christmas fair Danish					
Results retrieved	Results displayed	Precision/Quality Ask	Precision/Quality Bing	Precision/Quality ISA	
20	38	0.2000 0.2789	0.2500 0.3143	0.2368 0.2780	
Query Q09: steak London half price Monday					
Results retrieved	Results displayed	Precision/Quality Lycos	Precision/Quality Ixquick	Precision/Quality ISA	
20	36	0.3500 0.5619	0.2000 0.1952	0.3055 0.4414	

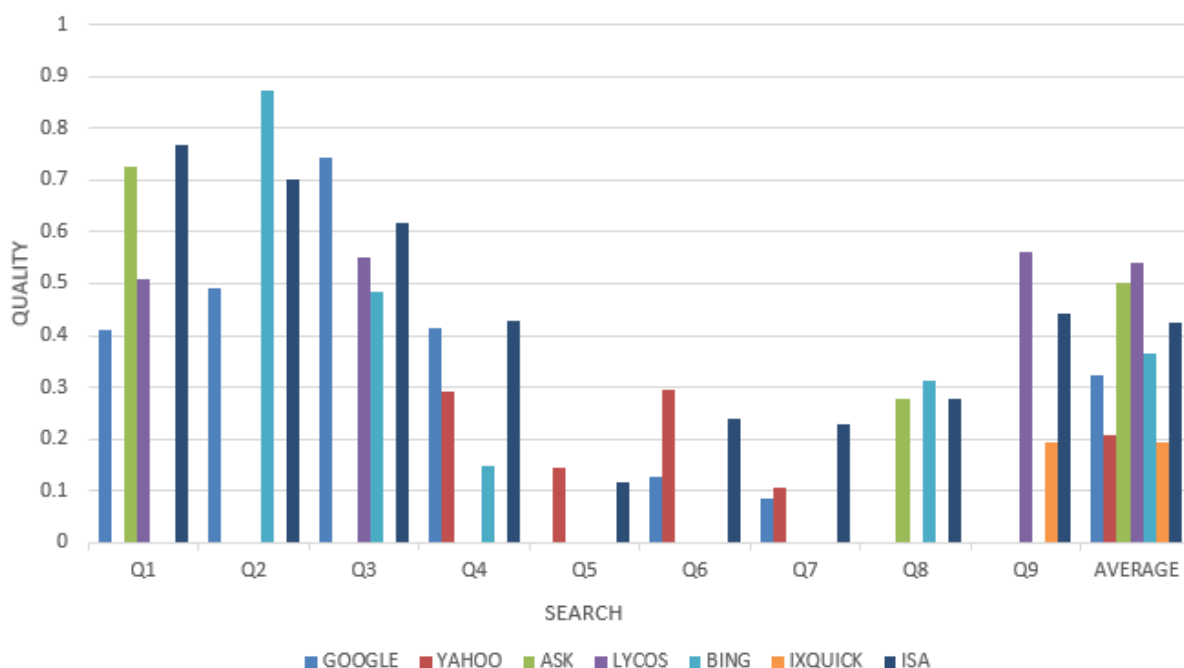


Figure 89: Web Search Result Cost Function

Query Q01: John Lewis lamp					
Results Retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Ask	Precision /Quality Lycos
10	60	0.1000 0.0910	0.1000 0.1455	0.0000 0.0000	0.0000 0.0000
Precision/Quality ISA		Precision/Quality Bing	Precision/Quality Ixquick	Precision/Quality MetaCrawler	Precision/Quality Yandex
0.0333 0.0650		0.0000 0.0000	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
Query Q02: cheap flight Toulouse					
Results Retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Ask	Precision /Quality Lycos
10	51	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000	0.2000 0.2182
Precision/Quality ISA		Precision/Quality Bing	Precision/Quality Ixquick	Precision/Quality MetaCrawler	Precision/Quality Yandex
0.1176 0.2179		0.4000 0.4364	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
Query Q03: Specsavers opening times Tottenham court road					
Results Retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Ask	Precision /Quality Lycos
10	52	0.0000 0.0000	0.1000 0.1818	0.0000 0.0000	0.1000 0.1455
Precision /Quality ISA		Precision /Quality Bing	Precision /Quality Ixquick	Precision /Quality MetaCrawler	Precision /Quality Yandex
0.0385 0.0123		0.0000 0.0000	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
Query Q04: Leeds Gallery Art					

Appendix

Results retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality ISA
10	18	0.3000 0.2182	0.3000 0.3455	0.3333 0.3567
Query Q05: Kingsley Amis Lucky Jim Review				
Results retrieved	Results displayed	Precision /Quality Yahoo	Precision /Quality Ask	Precision /Quality ISA
10	16	0.3000 0.2000	0.3000 0.4909	0.3750 0.4412
Query Q06: Stilton Cheese Recipe				
Results retrieved	Results displayed	Precision /Quality Bing		Precision /Quality ISA
10	10	0.5000 0.5091		0.5000 0.4364
Query Q07: hackney library opening hours				
Results retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Ask
10	34	0.6000 0.5091	0.3000 0.2182	0.4000 0.3818
		Precision /Quality Lycos	Precision /Quality Bing	Precision /Quality ISA
		0.4000 0.4182	0.4000 0.4727	0.6176 0.6286
Query Q08: Pilates Classes E8				
Results retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Ask
10	31	0.5000 0.3273	0.4000 0.4727	0.8000 0.6909
		Precision /Quality Lycos	Precision /Quality Bing	Precision /Quality ISA
		0.7000 0.7636	0.4000 0.2545	0.9032 0.9274
Query Q09: Tearoom Bath				
Results retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Ask
10	26	0.3000 0.2727	0.2000 0.2000	0.9000 0.8545
		Precision /Quality Lycos	Precision /Quality Bing	Precision /Quality ISA
		0.6000 0.5273	0.6000 0.5091	1.0000 1.0000
Query Q10: caricaturist wanted Cambridge artist portrait event				
Results retrieved	Results displayed	Precision /Quality Yahoo		Precision /Quality ISA
10	10	0.2000 0.1818		0.2000 0.3273

Appendix

Query Q11: predicting future method research magic					
Results retrieved	Results displayed	Precision /Quality Yahoo		Precision /Quality Bing	Precision /Quality ISA
10	14	0.0000 0.0000		0.1000 0.1273	0.0714 0.0095
Query Q12: Moment bargain Faust influences					
Results retrieved	Results displayed	Precision /Quality Lycos			Precision /Quality ISA
10	10	0.3000 0.4000			0.3000 0.3818
Query Q13: Homerton hospital diabetes					
Results Retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Bing	Precision /Quality ISA
10	21	0.2000 0.2727	0.4000 0.4727	0.3000 0.3818	0.4286 0.5324
Query Q14: Kate Pickett Bhutan Richard Wilkinson Wellbeing					
Results Retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Bing	Precision /Quality ISA
10	26	0.2000 0.1636	0.1000 0.0910	0.1000 0.1455	0.1539 0.1538
Query Q15: Filipino restaurant London					
Results Retrieved	Results displayed	Precision /Quality Google	Precision /Quality Yahoo	Precision /Quality Bing	Precision /Quality ISA
10	24	0.9000 0.8363	0.8000 0.8545	0.6000 0.4364	0.9583 0.9900
Query Q16: pubs near Waterloo					
Results Retrieved	Results displayed	Precision /Quality Bing			Precision /Quality ISA
10	10	0.3000 0.3091			0.9583 0.3455
Query Q17: furniture markets London					
Results Retrieved	Results displayed	Precision /Quality Bing			Precision /Quality ISA
10	10	0.5000 0.6000			0.5000 0.4909
Query Q18: late night museums London					
Results retrieved	Results displayed	Precision /Quality Bing			Precision /Quality ISA
10	10	0.6000 0.6182			0.6000 0.6909

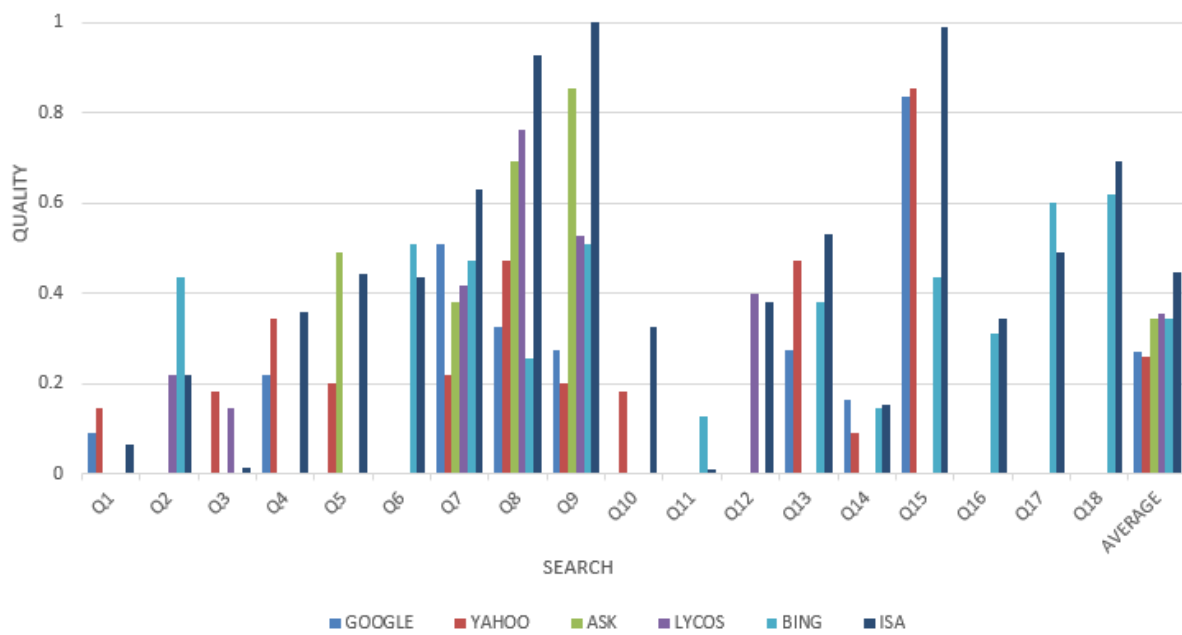


Figure 90: Web Search Result Cost Function

Results retrieved	Results selected	Precision	Quality Google	Quality ISA	Improvement ISA	Quality ISA Circle	Improvement ISA Circle
Query Q01: Annapurna trek tour operator							
10	04	0.4000	0.4727	0.3818	-19.23%	0.4545	-3.85%
Query Q02: sports bars Paris							
10	02	0.2000	0.2000	0.2182	9.10%	0.2364	18.20%
Query Q03: thor movie times east London							
10	04	0.4000	0.4000	0.4727	18.18%	0.3454	-13.65%
Query Q04: free French language meetups London							
10	05	0.5000	0.6182	0.4910	-20.58%	0.6545	5.87%
Query Q05: Wetherspoon London Holborn							
20	14	0.7000	0.7143	0.8380	17.32%	0.8523	19.32%

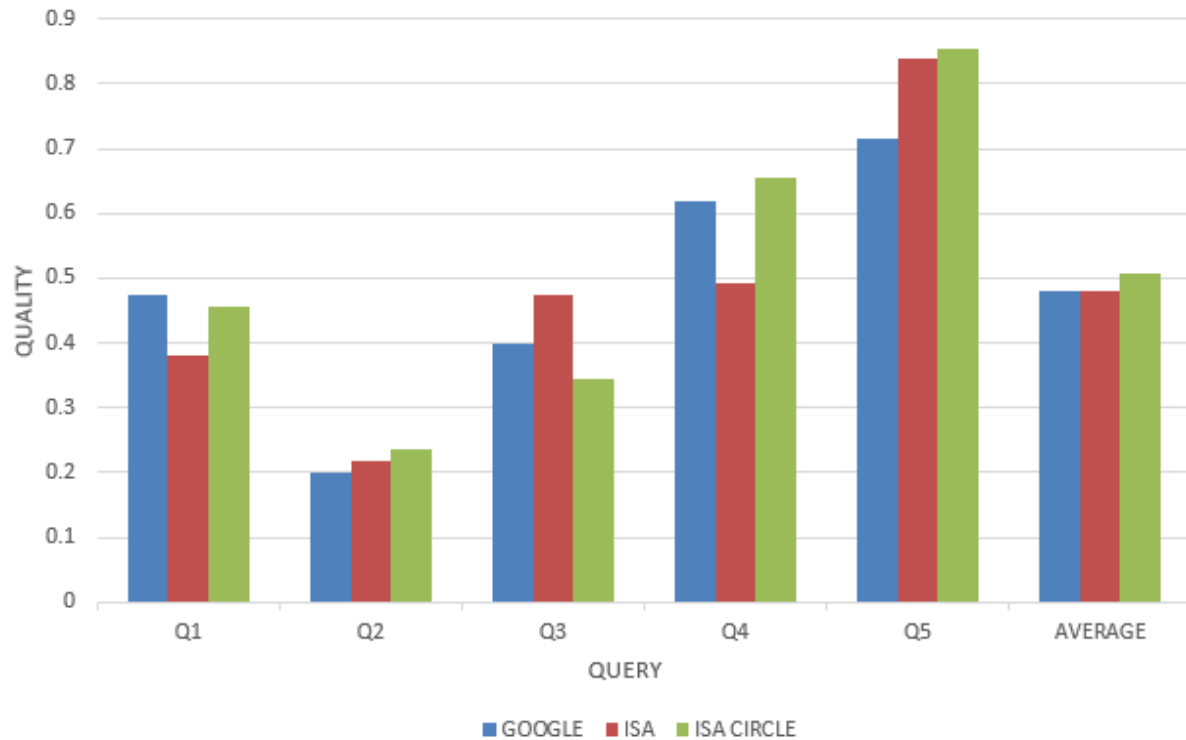


Figure 91: Google Search Relevant Centre Point

Results retrieved	Results selected	Precision	Quality Google	Quality ISA	Improvement ISA	Quality ISA Circle	Improvement ISA Circle
Query Q01: nutcracker ballet performance December							
20	17	0.8500	0.8571	0.8095	-5.55%	0.9048	5.57%
Query Q02: vintage dresses e8							
20	17	0.8500	0.8286	0.8857	6.89%	0.9048	9.20%
Query Q03: Christmas pudding recipe							
19	18	0.9474	0.9105	0.9000	-1.15%	0.9947	9.25%
Query Q04: how easy are chickens to keep?							
19	19	1.0000	1.0000	1.0000	0.00%	1.0000	0.00%
Query Q05: what are the best multivitamins?							
27	20	0.7407	0.6746	0.6984	3.53%	0.7222	7.06%
Query Q06: cheap places to live							
18	06	0.3333	0.3801	0.5205	36.94%	0.5322	40.02%
Query Q07: definition conditions play game							
20	06	0.3000	0.4190	0.3381	-19.31%	0.4190	0.00%
Query Q08: Ben Sherman Stockists London							
30	02	0.0667	0.1097	0.0970	-11.58%	0.0970	-11.58%
Query Q09: Claude Levi-strass system art							
20	06	0.3000	0.2952	0.1952	-33.88%	0.3333	12.91%

Appendix

Query Q10: advertising job Amsterdam							
30	02	0.0667	0.0368	0.1264	243.67%	0.1264	243.67%
Query Q11: Glastonbury festival line up 2014							
29	03	0.1034	0.1591	0.1484	-6.73%	0.1591	0.00%
Query Q12: nice bracelets London							
20	07	0.3500	0.3762	0.4190	11.38%	0.4381	16.45%
Query Q13: artist research opportunity							
20	08	0.4000	0.3666	0.3905	6.52%	0.4571	24.69%
Query Q14: Christmas Songs 1960s							
7	02	0.2857	0.2857	0.3928	37.49%	0.4284	49.95%
Query Q15: London Haunted Houses							
29	05	0.1724	0.2988	0.2781	-6.93%	0.2690	-9.97%
Query Q16: Christmas presents vegetarians							
20	02	0.1000	0.1238	0.1523	23.02%	0.1380	11.47%

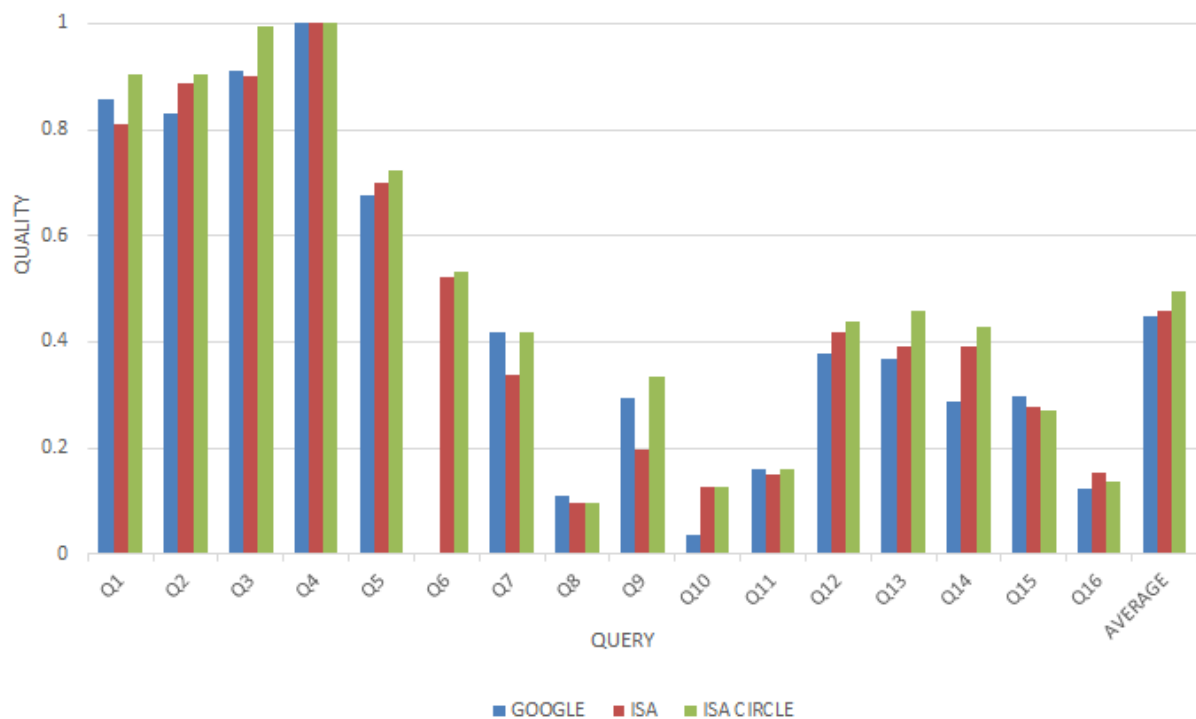


Figure 92: Google Search Relevant Centre Point

D User Evaluation – Learning Algorithms

Query: Best Beaches in the world – Gradient Descent - Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.4955	0.1162	1.0000	0.7488	0.6765	1.0000
ISA	1	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		101.83%	54.84%	0.00%	33.55%	47.81%	0.00%
Google	2	0.4250	0.1074	1.0000	0.7138	0.5716	1.0000
ISA	2	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Google	3	0.4684	0.1146	1.0000	0.7403	0.6222	1.0000
ISA	3	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: best beaches in world							
Universe: photos information listings pictures guides maps business insider trip destinations news travel rank places readers favourites holidays read breaks Seychelles archipelago Indian ocean popular resort photographed awards tripadvisor magazine							

Query: Rebounding Class London– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.3885	0.0644	0.8500	0.6835	0.5827	0.3333
ISA	1	0.9428	0.1682	0.8500	0.9853	0.8630	1.0000
I		142.68%	161.14%	0.00%	44.14%	48.09%	200.00%
Google	2	0.3767	0.0561	0.8500	0.6774	0.5790	0.2000
ISA	2	0.7793	0.1169	0.8500	0.8637	0.8395	0.5000
I		-17.34%	-30.48%	0.00%	-12.34%	-2.72%	-50.00%
Google	3	0.3767	0.0561	0.8500	0.6774	0.5790	0.2000
ISA	3	0.8430	0.1544	0.8500	0.9340	0.8432	1.0000
I		8.18%	32.07%	0.00%	8.15%	0.44%	100.00%
Relevant Dimensions: rebounding class							
Universe: look best tweet post longlegs joined location descriptions healthworks great first time experience level strength training developed taught discover enjoyed sweat laugh legs workout hip blog fitnesssoff popular nutrition plan food living optimal body							

Query: New Wave Cinema– Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.3709	0.0574	0.8500	0.6804	0.5568	0.2000
ISA	1	0.7997	0.1201	0.8500	0.8755	0.8444	0.5000
I		115.60%	109.38%	0.00%	28.67%	51.66%	150.00%
Google	2	0.3497	0.0538	0.9500	0.6653	0.5568	0.2000
ISA	2	0.8633	0.1299	0.9500	0.8986	0.9383	0.5000
I		7.95%	8.14%	11.76%	2.65%	11.11%	0.00%
Google	3	0.4039	0.0630	1.0000	0.6992	0.6309	0.2000
ISA	3	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		15.84%	38.50%	5.26%	11.28%	6.58%	100.00%
Relevant Dimensions: new wave cinema cine							
Universe: Japanese classics filmography join details French film guide best services production facilitation filming motion pictures feature visitor directors movement free dictionary abandoned traditional narrative techniques british openlearn university							

Query: Camping France Brittany– Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.5805	0.1288	1.0000	0.7935	0.7395	1.0000
ISA	1	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		72.26%	39.62%	0.00%	26.02%	35.23%	0.00%
Google	2	0.5805	0.1288	1.0000	0.7935	0.7395	1.0000
ISA	2	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Google	3	0.5805	0.1288	1.0000	0.7935	0.7395	1.0000
ISA	3	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: camping france brittany							
Universe: ferme available countryside farm rostrenen Normandy campsites rural guide selected small areas chalet holidays ferries perfect pursuits family western spacious static caravans people region landscape imaginable eurocampings coastline channel							

Query: Area of Outstanding Natural Beauty UK– Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.4637	0.0786	1.0000	0.7328	0.6951	0.3333
ISA	1	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		115.65%	128.83%	0.00%	36.45%	43.87%	200.00%
Google	2	0.4578	0.0717	1.0000	0.7237	0.7000	0.2500
ISA	2	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Google	3	0.4113	0.0643	1.0000	0.6964	0.6481	0.2500
ISA	3	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: area of outstanding natural beauty uk							
Universe: isle wight protecting landscape wildlife heritage information conservation visitors department northern Ireland view boundaries family protected working conserve enhance features events guide free England wales primary purpose							

Query: bulgakov margarita dante divine philosophy–Gradient Descent– Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.5563	0.0886	0.3000	0.7258	0.2889	1.0000
ISA	1	0.2325	0.0261	0.3000	0.5726	0.2716	0.2000
I		-58.21%	-70.53%	0.00%	-21.11%	-5.98%	-80.00%
Google	2	0.4099	0.0889	0.4500	0.6759	0.3420	1.0000
ISA	2	0.5559	0.0776	0.4500	0.7585	0.4667	0.5000
I		139.14%	197.41%	50.00%	32.47%	71.82%	150.00%
Google	3	0.5327	0.0812	0.2500	0.7042	0.2444	1.0000
ISA	3	0.6594	0.0929	0.2500	0.8355	0.2728	1.0000
I		18.61%	19.66%	-44.44%	10.15%	-41.53%	100.00%
Relevant Dimensions: bulgakov margarita dante divine master							
Universe: darkness visible Milton master detail analyzed various sources Shakespeare Goethe Russia Pontius pilate vagrant philosopher yeshua circle hell limbo souls veronica spira paper wandering charges levelled ethical norms expounded saint thomas comedy							

Query: levi strauss visual art language location drawing– Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.3514	0.0771	0.4000	0.6137	0.2765	1.0000
ISA	1	0.4257	0.0643	0.4000	0.6970	0.3889	0.5000
I		21.15%	-16.60%	0.00%	13.59%	40.63%	-50.00%
Google	2	0.3957	0.0950	0.7500	0.6730	0.4840	1.0000
ISA	2	0.7850	0.1435	0.7500	0.9062	0.7481	1.0000
I		84.38%	123.27%	87.50%	30.02%	92.38%	100.00%
Google	3	0.3850	0.0900	0.6000	0.6591	0.4111	1.0000
ISA	3	0.8519	0.1462	0.6000	0.9582	0.6235	1.0000
I		8.53%	1.87%	-20.00%	5.73%	-16.67%	0.00%
Relevant Dimensions: strauss							
Universe: anthropology aesthetics Cambridge navigation reconstructing internal logic thinking showing modern review proper poetry nineteenth century france signs claude bricolage subjectivity spiritual culture truth Colorado standards fluent music artistic							

Query: drum kit number dance movement mathematics rhythm repetition– Gradient Descent – Bing							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.1233	0.0103	0.1000	0.6077	0.0877	0.1667
ISA	1	0.8333	0.0667	0.1000	0.8155	0.1210	1.0000
I		575.68%	545.16%	0.00%	34.20%	38.03%	500.00%
Google	2	0.1233	0.0103	0.1000	0.6077	0.0877	0.1667
ISA	2	1.0000	0.0750	0.1000	1.0000	0.1222	1.0000
I		20.00%	12.50%	0.00%	22.63%	1.02%	0.00%
Google	3	0.1494	0.0170	0.2000	0.5471	0.1469	0.1667
ISA	3	0.7750	0.0913	0.2000	0.8828	0.2321	1.0000
I		-22.50%	21.67%	100.0%	-11.72%	89.90%	0.00%
Relevant Dimensions: drum mathematics rhythm African world news history							
Universe: African world news applied set history performance follow mali diansa							

Query: best road bike under 600– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.1863	0.0259	0.3500	0.5409	0.2494	0.2500
ISA	1	0.5136	0.0855	0.3500	0.6848	0.3605	1.0000
I		175.65%	230.41%	0.00%	26.61%	44.55%	300.00%
Google	2	0.2364	0.0384	0.5500	0.5920	0.3309	0.2500
ISA	2	0.7316	0.1226	0.5500	0.8330	0.5716	1.0000
I		42.44%	43.35%	57.14%	21.65%	58.56%	0.00%
Google	3	0.1918	0.0261	0.4500	0.5482	0.2864	0.2000
ISA	3	0.6872	0.1108	0.4500	0.8024	0.4790	1.0000
I		-6.07%	-9.63%	-18.18%	-3.67%	-16.20%	0.00%
Relevant Dimensions: best road bike under 600							
Universe: index general discussion review first grips wearing geared fixed single speed posts authors good tend start old models knocking stack exchange familiar forums mountain recreation feature trails equipment lighter faster suggestions							

Query: holiday cottage yorkshire £300 under– Gradient Descent –Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.4568	0.0895	0.3500	0.7414	0.2802	1.0000
ISA	1	0.6146	0.0995	0.3500	0.8105	0.3704	1.0000
I		34.56%	11.15%	0.00%	9.31%	32.16%	0.00%
Google	2	0.4708	0.0947	0.4000	0.7536	0.3247	1.0000
ISA	2	0.4201	0.0566	0.4000	0.6499	0.4000	0.5000
I		-31.65%	-43.10%	14.29%	-19.82%	8.00%	-50.00%
Google	3	0.3066	0.0690	0.4000	0.5678	0.2753	1.0000
ISA	3	0.3990	0.0439	0.4000	0.6623	0.4000	0.2000
I		-5.03%	-22.53%	0.00%	1.91%	0.00%	-60.00%
Relevant Dimensions: holiday cottage yorkshire age							
Universe: catering England results accommodation high season listings tripadvisor independent dales national trust visit room recommended homes breaks coastal staying Humberside rating reviews availability selection family character visiting							

Query: easy delicious dessert recipes – Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.1782	0.0209	0.3500	0.5493	0.2543	0.1250
ISA	1	0.6662	0.1063	0.3500	0.8517	0.3716	1.0000
I		273.97%	409.47%	0.00%	55.06%	46.12%	700.00%
Google	2	0.2722	0.0437	0.5000	0.6219	0.3272	0.2500
ISA	2	0.8057	0.1339	0.5000	0.9366	0.5284	1.0000
I		20.93%	26.02%	42.86%	9.97%	42.19%	0.00%
Google	3	0.4455	0.0605	0.3500	0.6806	0.3358	0.5000
ISA	3	0.6368	0.0982	0.3500	0.7667	0.3753	1.0000
I		-20.97%	-26.68%	-30.00%	-18.14%	-28.97%	0.00%
Relevant Dimensions: easy desserts recipes							
Universe: pudding simple quick food preparation complicated favourite woman home impress guests network topics results yogurt ideas fresh berry epicurious slideshow magazine classic cakes chocolate sweet perfect dinner party light bright delish menus							

Query: good quality english formal shoes – Gradient Descent – Bing							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.2189	0.0249	0.2500	0.5771	0.2173	0.2000
ISA	1	0.6042	0.0845	0.2500	0.7322	0.2765	1.0000
I		175.99%	238.86%	0.00%	26.88%	27.27%	400.00%
Google	2	0.1834	0.0194	0.2000	0.5702	0.1691	0.1667
ISA	2	0.6792	0.0642	0.2000	0.8182	0.2346	0.5000
I		12.40%	-24.10%	-20.00%	11.74%	-15.18%	-50.00%
Google	3	0.2878	0.0333	0.3000	0.6185	0.2728	0.2000
ISA	3	0.9762	0.1213	0.3000	0.9922	0.3506	1.0000
I		43.73%	89.05%	50.00%	21.28%	49.47%	100.00%
Relevant Dimensions: quality English shoes							
Universe: buy permanent style appropriate number made brands Britain british variety cheaper getting deal masterpiece finest craftsmen leathers established England manufacture unique materials traditional generations collections excellent value							

Query: best vegetarian parmesan substitutes – Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.3585	0.0580	0.3500	0.6681	0.2901	0.5000
ISA	1	0.6524	0.1042	0.3500	0.8382	0.3716	1.0000
I		81.97%	79.54%	0.00%	25.47%	28.09%	100.00%
Google	2	0.5436	0.1043	0.4000	0.8141	0.3383	1.0000
ISA	2	0.9306	0.1322	0.4000	0.9795	0.4519	1.0000
I		42.63%	26.91%	14.29%	16.85%	21.59%	0.00%
Google	3	0.6215	0.1103	0.4000	0.8516	0.3852	1.0000
ISA	3	0.9526	0.1331	0.4000	0.9846	0.4556	1.0000
I		2.37%	0.68%	0.00%	0.52%	0.82%	0.00%
Relevant Dimensions: parmesan cheese							
Universe: society campaign alternatives veggie approved popular foods ranked worst vegan omelette tofu ingredient substitutions healthy vegetable stock beans pasta bouillon nutritionist parma favourite supermarkets boards lovely definition prefer							

Query: best pub south Kensington world cup watch – Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.5087	0.0724	0.2000	0.6725	0.1815	1.0000
ISA	1	0.4515	0.0637	0.2000	0.5970	0.2099	1.0000
I		-11.23%	-12.00%	0.00%	-11.23%	15.65%	0.00%
Google	2	0.4620	0.0818	0.3000	0.6763	0.2395	1.0000
ISA	2	0.8269	0.1130	0.3000	0.9376	0.3370	1.0000
I		83.14%	77.38%	50.00%	57.07%	60.59%	0.00%
Google	3	0.6101	0.0931	0.3000	0.7560	0.2864	1.0000
ISA	3	0.9151	0.1171	0.3000	0.9636	0.3481	1.0000
I		10.66%	3.65%	0.00%	2.77%	3.30%	0.00%
Relevant Dimensions: pub south Kensington world cup							
Universe: builder's arms freshly cooked food speciality beers games screen Gloucester child friendly serving facilities covered smoking area sports traditional brew Kentish ale spitfire Whitstable bishops finger London fluid guide live coverage champions league							

Query: Mole recipe chicken mexican– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.4024	0.0661	1.0000	0.6879	0.6395	0.3333
ISA	1	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		148.50%	172.11%	0.00%	45.37%	56.37%	200.00%
Google	2	0.4063	0.0673	1.0000	0.6916	0.6358	0.3333
ISA	2	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Google	3	0.3354	0.0480	0.9000	0.6548	0.5358	0.1429
ISA	3	0.9940	0.1744	0.9000	0.9987	0.9198	1.0000
I		-0.60%	-3.02%	-10.00%	-0.13%	-8.02%	0.00%
Relevant Dimensions: Mole recipe chicken mexican							
Universe: make texas cooking turkey authentic food enjoy spicyness love dish unique chocolate sauce adobo quick easy spice weeknight meal ready minutes rating browned pieces simmer spicy tomato favourite lifestyle traditional blend dried chiles nuts seeds							

Query: colobus black and white monkey– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.3040	0.0699	0.4500	0.5698	0.3160	1.0000
ISA	1	0.5408	0.0918	0.4500	0.6961	0.4568	1.0000
I		77.89%	31.24%	0.00%	22.16%	44.53%	0.00%
Google	2	0.4411	0.0793	0.3000	0.6633	0.2667	1.0000
ISA	2	0.5217	0.0821	0.3000	0.6813	0.3160	1.0000
I		-3.53%	-10.53%	-33.33%	-2.11%	-30.81%	0.00%
Google	3	0.4575	0.0897	0.5000	0.6732	0.4469	1.0000
ISA	3	0.5114	0.0653	0.5000	0.7219	0.5012	0.3333
I		-1.97%	-20.43%	66.67%	5.96%	58.59%	-66.67%
Relevant Dimensions: colobus black and white monkey							
Universe: exploring nature detailed information grades students educators parents world stump eastern shaped cape species guereza Angolan angolensis Tanzania countries angola Burundi Kenya Rwanda Uganda Zaire geographically subspecies							

Query: Southend cafes coffee tea– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.5708	0.1211	0.8000	0.7828	0.6580	1.0000
ISA	1	0.9106	0.1621	0.8000	0.9727	0.8160	1.0000
I		59.53%	33.86%	0.00%	24.26%	24.02%	0.00%
Google	2	0.5110	0.1130	0.6500	0.7891	0.5309	1.0000
ISA	2	0.7455	0.1291	0.6500	0.8413	0.6605	1.0000
I		-18.13%	-20.34%	-18.75%	-13.51%	-19.06%	0.00%
Google	3	0.5110	0.1130	0.6500	0.7891	0.5309	1.0000
ISA	3	0.7455	0.1291	0.6500	0.8413	0.6605	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: southend tea							
Universe: essex shops united kingdom asked walk cafe's bars rooms bistro's estuary smileys emporium waters edge jangles barge captainstable riverside jolly utopia brentwood Chelmsford sea family emphasis healthy fresh food meals course brewed							

Query: southend jazz bars events june 2014 leigh– Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.1683	0.0188	0.4500	0.5434	0.2568	0.0667
ISA	1	0.3795	0.0518	0.4500	0.6590	0.4222	0.3333
I		125.53%	175.66%	0.00%	21.28%	64.42%	400.00%
Google	2	0.3062	0.0725	0.5500	0.5791	0.3568	1.0000
ISA	2	0.8910	0.1442	0.5500	0.9674	0.5926	1.0000
I		134.75%	178.11%	22.22%	46.80%	40.35%	200.00%
Google	3	0.3248	0.0767	0.6500	0.5979	0.4111	1.0000
ISA	3	0.9752	0.1576	0.6500	0.9937	0.7012	1.0000
I		9.45%	9.30%	18.18%	2.72%	18.33%	0.00%
Relevant Dimensions: southend jazz events june 2014 leigh							
Universe: homepage gazette found results type music blues social sports club local radio charity profit include classical tunes Thursday free entry sea parking available front dinner drinks greeted characters talent show palace private busy boat town							

Query: projector versus large screen television– Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.2792	0.0397	0.6000	0.6130	0.4099	0.2000
ISA	1	0.5584	0.1029	0.6000	0.7285	0.5753	1.0000
I		100.01%	159.04%	0.00%	18.84%	40.36%	400.00%
Google	2	0.3463	0.0540	0.7000	0.6618	0.4951	0.2500
ISA	2	0.9516	0.1600	0.7000	0.9885	0.7370	1.0000
I		70.39%	55.46%	16.67%	35.68%	28.11%	0.00%
Google	3	0.3699	0.0568	0.7500	0.6732	0.5469	0.2500
ISA	3	0.9750	0.1646	0.7500	0.9944	0.7877	1.0000
I		2.46%	2.89%	7.14%	0.59%	6.87%	0.00%
Relevant Dimensions: projector large screen							
Universe: answers become bigger brighter come high resolutions moreover compare using notice flat panel plasma worlds smaller tube predecessors video home sizes projection standard satellite second distance display devices posted living room							

Query: London zoo most famous exhibit – Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.1136	0.0098	0.1000	0.5991	0.0778	0.1667
ISA	1	0.0820	0.0056	0.1000	0.6177	0.0815	0.0588
I		-27.80%	-43.42%	0.00%	3.10%	4.76%	-64.71%
Google	2	0.1130	0.0133	0.2000	0.5138	0.1296	0.1667
ISA	2	0.6696	0.0853	0.2000	0.8324	0.2198	1.0000
I		716.21%	1430.08%	100.0%	34.76%	169.70%	1600.00%
Google	3	0.1189	0.0136	0.2000	0.5177	0.1383	0.1667
ISA	3	1.0000	0.1042	0.2000	1.0000	0.2395	1.0000
I		49.33%	22.16%	0.00%	20.13%	8.99%	0.00%
Relevant Dimensions: London zoo most exhibit human							
Universe: jump notable animals jumbo largest elephant jardin giant pandas terrifying cracked weird world ashamed popular fact sordid decided foods ingredients haunt dreams entitled human plantes paris time transferred zoo's four days late august eight							

Query: vegetarian curry recipe– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.4744	0.1167	0.7500	0.7838	0.5000	1.0000
ISA	1	0.8931	0.1588	0.7500	0.9706	0.7667	1.0000
I		88.25%	36.11%	0.00%	23.83%	53.33%	0.00%
Google	2	0.8303	0.1529	0.8500	0.9246	0.8370	1.0000
ISA	2	0.8862	0.1605	0.8500	0.9561	0.8519	1.0000
I		-0.78%	1.07%	13.33%	-1.49%	11.11%	0.00%
Google	3	0.8916	0.1670	0.9000	0.9691	0.8630	1.0000
ISA	3	0.9689	0.1730	0.9000	0.9929	0.9099	1.0000
I		9.34%	7.79%	5.88%	3.85%	6.81%	0.00%
Relevant Dimensions: vegetarian curry recipe							
Universe: food vegetable cook feast collection delicious best red online ditch takeaway Indian ideas dish type main course find peanut palak paneer egg lentil curries healthy cooking light slow cooker entrees side dishes salads great circles pressure rice							

Query: vegetarian restaurants dalston– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.5075	0.0908	0.8000	0.7613	0.6407	0.5000
ISA	1	0.9409	0.1645	0.8000	0.9824	0.8247	1.0000
I		85.39%	81.14%	0.00%	29.03%	28.71%	100.00%
Google	2	0.4140	0.0728	0.5500	0.7090	0.4617	0.5000
ISA	2	0.5250	0.0737	0.5500	0.7150	0.5420	0.5000
I		-44.20%	-55.18%	-31.25%	-27.22%	-34.28%	-50.00%
Google	3	0.4104	0.0743	0.6000	0.7093	0.4852	0.5000
ISA	3	0.7872	0.1401	0.6000	0.9308	0.6148	1.0000
I		49.94%	89.93%	9.09%	30.18%	13.44%	100.00%
Relevant Dimensions: vegetarian							
Universe: vortex gillet street live music jazz London tastecard gives meals food join members taking free trial trusted compare find businesses view profiles trade association memberships vegan healthy organic directory natural health stores							

Query: Memes history– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.1313	0.0169	0.4000	0.5163	0.1728	0.1111
ISA	1	0.3594	0.0458	0.4000	0.6348	0.3852	0.3333
I		173.75%	171.62%	0.00%	22.95%	122.86%	200.00%
Google	2	0.2523	0.0393	0.6000	0.5967	0.3593	0.2500
ISA	2	0.4832	0.0613	0.6000	0.7090	0.5691	0.2500
I		34.42%	33.74%	50.00%	11.68%	47.76%	-25.00%
Google	3	0.2444	0.0374	0.6000	0.5883	0.3667	0.2500
ISA	3	0.7821	0.1390	0.6000	0.9262	0.6160	1.0000
I		61.87%	126.86%	0.00%	30.63%	8.24%	300.00%
Relevant Dimensions: memes history							
Universe: obsessed internet rise fall boxy gender relationships found resources cataloguing explaining currently golden age current events trending topics notion flourished uncovered dish world hospitable environment view sabotage times specific							

Query: hayfever remedies– Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.4484	0.0801	1.0000	0.7348	0.6568	0.3333
ISA	1	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		123.01%	124.60%	0.00%	36.09%	52.26%	200.00%
Google	2	0.9974	0.1773	0.9500	0.9995	0.9605	1.0000
ISA	2	0.9974	0.1773	0.9500	0.9995	0.9605	1.0000
I		-0.26%	-1.46%	-5.00%	-0.05%	-3.95%	0.00%
Google	3	0.9974	0.1773	0.9500	0.9995	0.9605	1.0000
ISA	3	0.9974	0.1773	0.9500	0.9995	0.9605	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: hayfever remedies							
Universe: opticrom range counter treatments help provide relief homeopathic natural therapy lasting community social general discussion dying literally ruining weekend itching eyes runny nose sneezes tried branded medication shop news research health							

Query: the globe theatre – Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.4894	0.0834	1.0000	0.7488	0.7062	0.3333
ISA	1	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		104.34%	115.74%	0.00%	33.54%	41.61%	200.00%
Google	2	0.5445	0.1232	0.9500	0.7763	0.6951	1.0000
ISA	2	0.9740	0.1757	0.9500	0.9937	0.9519	1.0000
I		-2.60%	-2.32%	-5.00%	-0.63%	-4.81%	0.00%
Google	3	0.6035	0.1378	1.0000	0.8432	0.7494	1.0000
ISA	3	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		2.67%	2.37%	5.26%	0.63%	5.06%	0.00%
Relevant Dimensions: the globe theatre							
Universe: visit site dedicated providing information history construction actors accurate facts oxford dictionaries definition Shakespeare brief guide London council museums arts venues redevelopment grade listed priorities high street view plan trip							

Query: real ale festivals london – Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.4395	0.0849	0.7500	0.7389	0.5519	0.5000
ISA	1	0.8380	0.1420	0.7500	0.8777	0.7667	1.0000
I		90.64%	67.14%	0.00%	18.78%	38.93%	100.00%
Google	2	0.4975	0.0934	0.7500	0.7730	0.5753	0.5000
ISA	2	0.8672	0.1555	0.7500	0.9574	0.7630	1.0000
I		3.49%	9.49%	0.00%	9.08%	-0.48%	0.00%
Google	3	0.5110	0.0962	0.8000	0.7798	0.6160	0.5000
ISA	3	0.8721	0.1586	0.8000	0.9587	0.8049	1.0000
I		0.56%	1.99%	6.67%	0.14%	5.50%	0.00%
Relevant Dimensions: real ale fest							
Universe: travel ciders british beer Olympia promise perries pubs perfect pint shows related events serving list free information summer brew fest fields fine food royal treat beautiful borough parks historic houses theatres craft rising returns promising braver							

Query: shoe repair Ladbroke grove – Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.7278	0.1351	0.8000	0.8405	0.7457	1.0000
ISA	1	0.9476	0.1656	0.8000	0.9864	0.8247	1.0000
I		30.21%	22.64%	0.00%	17.36%	10.60%	0.00%
Google	2	0.7863	0.1589	0.9000	0.9369	0.8000	1.0000
ISA	2	0.9886	0.1742	0.9000	0.9976	0.9173	1.0000
I		4.33%	5.16%	12.50%	1.14%	11.23%	0.00%
Google	3	0.7660	0.1583	0.9000	0.9339	0.7852	1.0000
ISA	3	0.9682	0.1729	0.9000	0.9927	0.9099	1.0000
I		-2.06%	-0.72%	0.00%	-0.49%	-0.81%	0.00%
Relevant Dimensions: shoe repair Ladbroke grove							
Universe: touch local find get contact details special offers yell business photos including opening hours shops station list directory information Kensington books music video countries sell confidential broker road conventional established give premium							

Query: destiny traveller bungie – Reinforcement Learning – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.2074	0.0358	0.3000	0.5300	0.2062	0.5000
ISA	1	0.2559	0.0311	0.3000	0.5890	0.2765	0.2500
I		23.39%	-13.08%	0.00%	11.13%	34.13%	-50.00%
Google	2	0.2566	0.0483	0.5500	0.5814	0.3370	0.5000
ISA	2	0.8084	0.1344	0.5500	0.9159	0.5802	1.0000
I		215.92%	331.79%	83.33%	55.52%	109.82%	300.00%
Google	3	0.2544	0.0492	0.6000	0.5835	0.3346	0.5000
ISA	3	0.6814	0.1237	0.6000	0.8526	0.6037	1.0000
I		-15.71%	-7.97%	9.09%	-6.92%	4.04%	0.00%
Relevant Dimensions: destiny traveller bungie game							
Universe: news believe bungie's guardians posts authors traveller large spherical construct white surface community upcoming game thoughts think writer history sites countdown live play footage sony's conference collapse living earth saved software							

Query: milan cheap flight ticket – Gradient Descent – Google							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Google	1	0.1705	0.0179	0.1500	0.5604	0.1370	0.2500
ISA	1	1.0000	0.0917	0.1500	1.0000	0.1815	1.0000
I		486.67%	412.17%	0.00%	78.46%	32.43%	300.00%
Google	2	0.1705	0.0179	0.1500	0.5604	0.1370	0.2500
ISA	2	1.0000	0.0917	0.1500	1.0000	0.1815	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Google	3	0.1875	0.0156	0.1000	0.6250	0.1012	0.2500
ISA	3	1.0000	0.0750	0.1000	1.0000	0.1222	1.0000
I		0.00%	-18.18%	-33.33%	0.00%	-32.65%	0.00%
Relevant Dimensions: milan cheap flight find							
Universe: Bangladesh low cost find best offer easyjet book tourism information sightseeing shopping restaurants nightlife stylish sophisticated destination London edreams early often price							

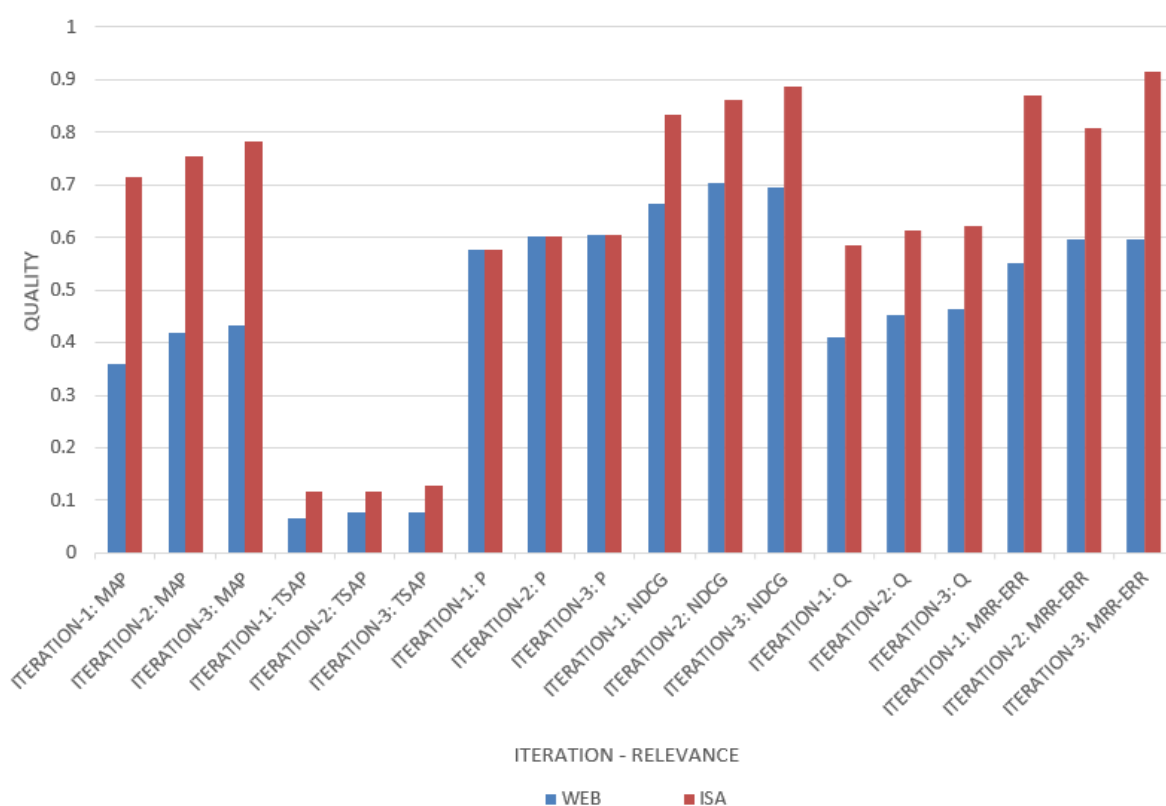


Figure 93: Web Search Learning evaluation – Gradient Descent - Average

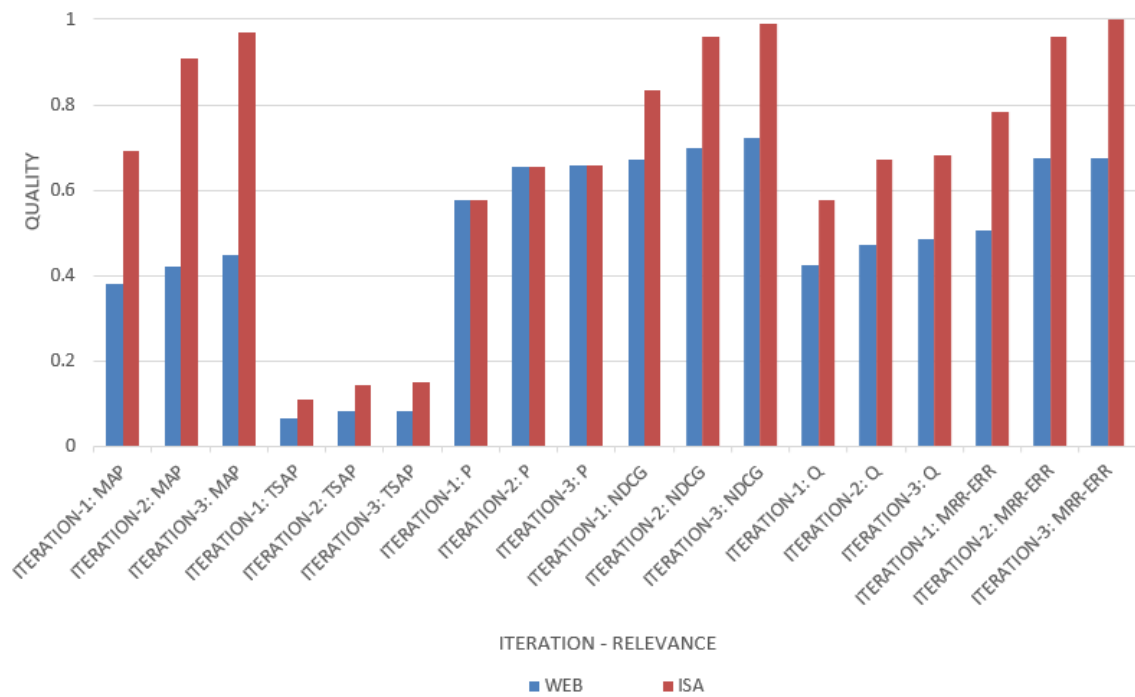


Figure 94: Web Search Learning evaluation – Reinforcement Learning - Average

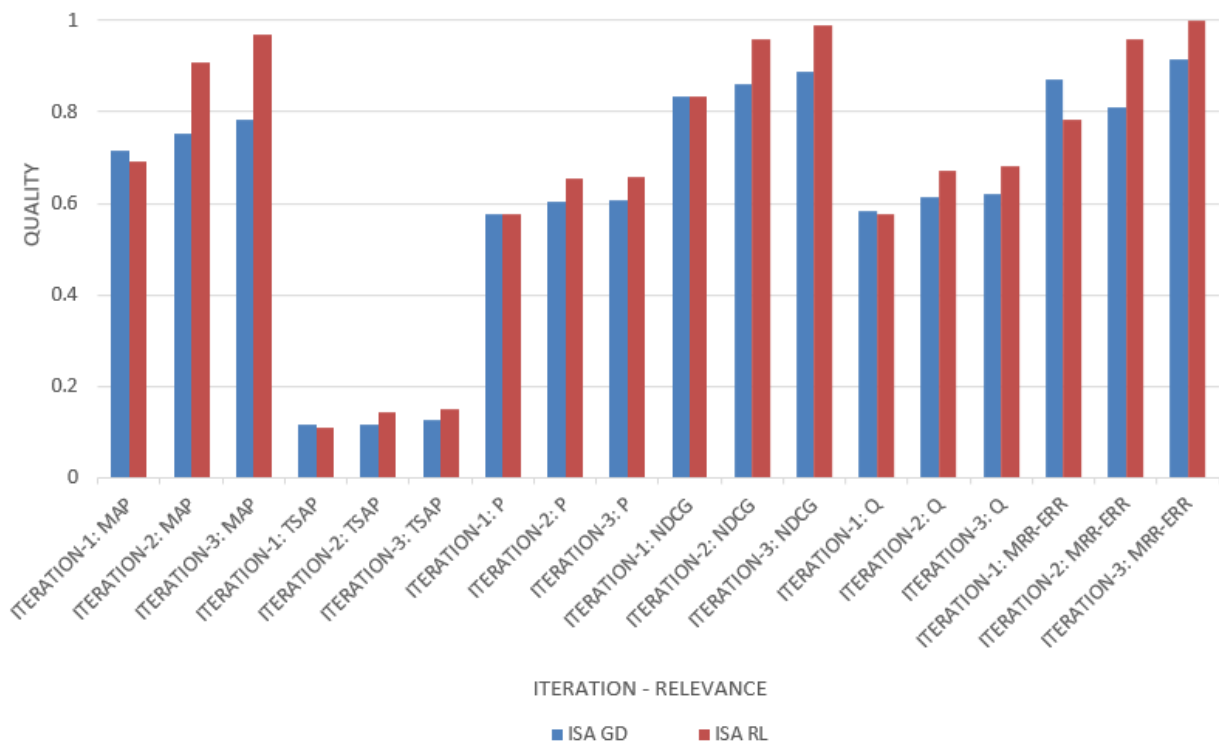


Figure 95: Web Search Learning evaluation – Evaluation between learnings

Query: Random Neural Network – Gradient Descent – Google Scholar							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.7580	0.1408	0.6000	0.9285	0.5531	1.0000
ISA	1	0.7923	0.1411	0.6000	0.9351	0.6136	1.0000
I		4.52%	0.15%	0.00%	0.71%	10.94%	0.00%
GS	2	0.3357	0.0538	0.5000	0.6575	0.3975	0.3333
ISA	2	0.7324	0.1265	0.5000	0.8991	0.5185	1.0000
I		-7.56%	-10.29%	-16.67%	-3.85%	-15.49%	0.00%
GS	3	0.2496	0.0355	0.4000	0.5933	0.3148	0.2500
ISA	3	0.6478	0.1100	0.4000	0.8478	0.4136	1.0000
I		-11.55%	-13.05%	-20.00%	-5.71%	-20.24%	0.00%
Relevant Dimensions: Random Neural Network							
Universe: negative positive signals gelenbe computation conclusions circulate neurons quality multiple classes recurrent texture generation systems dynamical approach optimization theoretical practical stability inhibitory excitatory learning artificial							

Query: Random Neural Network – Gradient Descent – IEEE Xplore							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
IEEE	1	0.3082	0.0490	0.5000	0.6340	0.3951	0.3333
ISA	1	0.7359	0.1256	0.5000	0.8901	0.5198	1.0000
I		138.76%	156.42%	0.00%	40.39%	31.56%	200.00%
IEEE	2	0.4592	0.0904	0.6500	0.6663	0.5605	1.0000
ISA	2	0.9586	0.1566	0.6500	0.9893	0.6975	1.0000
I		30.26%	24.66%	30.00%	11.14%	34.20%	0.00%
IEEE	3	0.4843	0.0953	0.7000	0.6852	0.6037	1.0000
ISA	3	0.9636	0.1607	0.7000	0.9915	0.7407	1.0000
I		0.52%	2.62%	7.69%	0.22%	6.19%	0.00%
Relevant Dimensions: Random Neural Network							
Universe: decoder error connecting codes patents filter detection radio synchronized interactions computation multiple classes signals gelenbe learning process cognitive packet spiked nonlinearity model color pattern recognition algorithm image processing							

Query: Random Neural Network – Gradient Descent – CiteSeerX							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
CSX	1	0.8044	0.1169	0.3500	0.9219	0.3914	1.0000
ISA	1	0.7144	0.1092	0.3500	0.8705	0.3827	1.0000
I		-11.19%	-6.54%	0.00%	-5.57%	-2.21%	0.00%
CSX	2	0.7775	0.1184	0.4000	0.8963	0.4407	1.0000
ISA	2	0.6297	0.1003	0.4000	0.7606	0.4272	1.0000
I		-11.85%	-8.21%	14.29%	-12.62%	11.61%	0.00%
CSX	3	0.8828	0.1339	0.4500	0.9604	0.4988	1.0000
ISA	3	0.7216	0.1127	0.4500	0.8129	0.4864	1.0000
I		14.59%	12.41%	12.50%	6.88%	13.87%	0.00%
Relevant Dimensions: Random Neural Network							
Universe: emergency management challenges modelling simulation analysis network optimization algorithmic solution security distributed database systems technology prototype dynamic multicast training newton methods error function weights							

Query: Random Neural Network – Gradient Descent – Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
MA	1	0.4737	0.1135	0.7000	0.7795	0.4938	1.0000
ISA	1	0.8114	0.1492	0.7000	0.9431	0.7012	1.0000
I		71.28%	31.46%	0.00%	20.99%	42.00%	0.00%
MA	2	0.4740	0.1188	0.8000	0.7878	0.5321	1.0000
ISA	2	0.7910	0.1173	0.8000	0.8717	0.8037	0.5000
I		-2.51%	-21.38%	14.29%	-7.57%	14.61%	-50.00%
MA	3	0.4230	0.1075	0.7500	0.7476	0.4864	1.0000
ISA	3	0.7943	0.1382	0.7500	0.8624	0.7556	1.0000
I		0.41%	17.76%	-6.25%	-1.07%	-5.99%	100.00%
Relevant Dimensions: Random Neural Network							
Universe: synchronized interactions distributed systems neural artificial propagate information distances physical signalling distinguish pulses recognition shaped objects stromg clutter detect targets stability enlargement fusion intelligence generation							

Query: Web Search Engine – Gradient Descent – Google Scholar							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.2378	0.0307	0.4500	0.5845	0.3284	0.1667
ISA	1	0.6749	0.1160	0.4500	0.8595	0.4667	1.0000
I		183.82%	278.26%	0.00%	47.04%	42.11%	500.00%
GS	2	0.3415	0.0785	0.4500	0.6147	0.3086	1.0000
ISA	2	0.4258	0.0525	0.4500	0.6856	0.4420	0.2500
I		-36.91%	-54.75%	0.00%	-20.23%	-5.29%	-75.00%
GS	3	0.3492	0.0789	0.4000	0.6268	0.2642	1.0000
ISA	3	0.3775	0.0434	0.4000	0.6455	0.3938	0.2500
I		-11.35%	-17.32%	-11.11%	-5.86%	-10.89%	0.00%
Relevant Dimensions: Web Search Engine							
Universe: graphic snapshots indexing page contents user selection criteria find report inventive associated crawler function display supervised possibility activities internet algorithms methods effective results relevant techniques clustering automatic mining							

Query: Web Search Engine – Gradient Descent – IEEE Xplore							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
IEEE	1	0.1974	0.0239	0.4000	0.5614	0.2765	0.1250
ISA	1	0.5082	0.0925	0.4000	0.7144	0.3901	1.0000
I		157.45%	287.20%	0.00%	27.25%	41.07%	700.00%
IEEE	2	0.2249	0.0292	0.5500	0.5804	0.3420	0.1250
ISA	2	0.6937	0.1182	0.5500	0.8128	0.5679	1.0000
I		36.51%	27.88%	37.50%	13.79%	45.57%	0.00%
IEEE	3	0.3437	0.0819	0.7000	0.6199	0.4469	1.0000
ISA	3	0.7002	0.1267	0.7000	0.8227	0.6901	1.0000
I		0.94%	7.16%	27.27%	1.22%	21.52%	0.00%
Relevant Dimensions: Web Search Engine							
Universe: program analysis personalised fuzzy concept network link structure toward feedback based advanced information networking applications extracting knowledge results tools artificial intelligence design implementation technology engineering							

Query: Web Search Engine – Gradient Descent – CiteSeerX							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
CSX	1	0.4522	0.0738	0.5000	0.7230	0.4432	0.5000
ISA	1	0.6747	0.1123	0.5000	0.7951	0.5222	1.0000
I		49.21%	52.04%	0.00%	9.97%	17.83%	100.00%
CSX	2	0.4373	0.1003	0.6500	0.7069	0.4877	1.0000
ISA	2	0.7658	0.1409	0.6500	0.9208	0.6531	1.0000
I		13.51%	25.54%	30.00%	15.81%	25.06%	0.00%
CSX	3	0.4373	0.1003	0.6500	0.7069	0.4877	1.0000
ISA	3	0.7658	0.1409	0.6500	0.9208	0.6531	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: Web Search Engine							
Universe: query systems intelligence technology viewing automatic evaluation long term learning data mining knowledge principles discovery impact subject searching analyses results similarity computer science							

Query: Web Search Engine – Gradient Descent – Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
MA	1	0.2683	0.0392	0.4500	0.5880	0.3605	0.3333
ISA	1	0.6762	0.1146	0.4500	0.8520	0.4704	1.0000
I		152.02%	192.13%	0.00%	44.90%	30.48%	200.00%
MA	2	0.3806	0.0645	0.6000	0.7009	0.4568	0.3333
ISA	2	0.6688	0.1246	0.6000	0.8568	0.5951	1.0000
I		-1.09%	8.71%	33.33%	0.56%	26.51%	0.00%
MA	3	0.3735	0.0648	0.7000	0.6893	0.4938	0.3333
ISA	3	0.7565	0.1107	0.7000	0.8605	0.7136	0.5000
I		13.12%	-11.18%	16.67%	0.44%	19.92%	-50.00%
Relevant Dimensions: Web Search Engine							
Universe: request isolation results query identical independent next generation increasing information implicit explicit functionality restricted high level discussion problems retrieval generic architecture crawling storage indexing analysis performance							

Query: Meta Search Engine – Gradient Descent – Google Scholar							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.5533	0.1175	0.7000	0.7767	0.5617	1.0000
ISA	1	0.8706	0.1541	0.7000	0.9636	0.7185	1.0000
I		57.37%	31.11%	0.00%	24.07%	27.91%	0.00%
GS	2	0.3624	0.0577	0.7500	0.6747	0.5136	0.2500
ISA	2	0.7603	0.1093	0.7500	0.8472	0.7605	0.5000
I		-12.67%	-29.05%	7.14%	-12.08%	5.84%	-50.00%
GS	3	0.3810	0.0603	0.8000	0.6844	0.5531	0.2500
ISA	3	0.7655	0.1119	0.8000	0.8506	0.8000	0.5000
I		0.67%	2.41%	6.67%	0.41%	5.19%	0.00%
Relevant Dimensions: Meta Search Engine							
Universe: architecture google system data sources controller query information retrieval method snippet database results index categories functionality distributed services ranking algorithms function innovative reduced extract							

Query: Meta Search Engine – Gradient Descent – IEEE Xplore							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
IEEE	1	0.1912	0.0255	0.5500	0.5549	0.2889	0.1429
ISA	1	0.4695	0.0554	0.5500	0.6994	0.5321	0.2000
I		145.62%	117.14%	0.00%	26.03%	84.19%	40.00%
IEEE	2	0.2625	0.0359	0.6000	0.5961	0.4000	0.2000
ISA	2	0.6938	0.1279	0.6000	0.8760	0.6012	1.0000
I		47.77%	130.93%	9.09%	25.26%	12.99%	400.00%
IEEE	3	0.3153	0.0436	0.7000	0.6300	0.4852	0.2000
ISA	3	0.7441	0.1387	0.7000	0.8998	0.6951	1.0000
I		7.24%	8.47%	16.67%	2.72%	15.61%	0.00%
Relevant Dimensions: Meta Search Engine							
Universe: personalised design metadata cross applications web technologies evaluation merge algorithms redundancy servers distributed advanced networking automation software user profile framework agent							

Query: Meta Search Engine – Gradient Descent – CiteSeerX							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
CSX	1	0.5056	0.1168	0.7000	0.7968	0.5469	1.0000
ISA	1	0.7806	0.1432	0.7000	0.9183	0.7037	1.0000
I		54.38%	22.56%	0.00%	15.26%	28.67%	0.00%
CSX	2	0.5741	0.1224	0.8000	0.7871	0.6593	1.0000
ISA	2	0.8782	0.1599	0.8000	0.9637	0.8049	1.0000
I		12.50%	11.66%	14.29%	4.94%	14.39%	0.00%
CSX	3	0.6642	0.1435	0.8000	0.8934	0.6728	1.0000
ISA	3	0.8833	0.1603	0.8000	0.9654	0.8062	1.0000
I		0.58%	0.28%	0.00%	0.18%	0.15%	0.00%
Relevant Dimensions: Meta Search Engine							
Universe: information retrieval multiple domains resources page ranking techniques adaptive interfaces service methods query specific annotation results framework mechanism model forms interactions applications management large scale weight							

Query: Meta Search Engine – Gradient Descent – Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
MA	1	0.5884	0.1286	0.7500	0.8435	0.6284	1.0000
ISA	1	0.7517	0.1283	0.7500	0.8194	0.7494	1.0000
I		27.76%	-0.20%	0.00%	-2.86%	19.25%	0.00%
MA	2	0.5688	0.1237	0.8000	0.7908	0.6432	1.0000
ISA	2	0.8673	0.1563	0.8000	0.9500	0.8074	1.0000
I		15.37%	21.86%	6.67%	15.94%	7.74%	0.00%
MA	3	0.5688	0.1237	0.8000	0.7908	0.6432	1.0000
ISA	3	0.8673	0.1563	0.8000	0.9500	0.8074	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: Meta Search Engine							
Universe: query citations multiple parallel periods downtime deficiencies coverage inconsistent inefficient interfaces databases poor ranking relevancy precision difficulties spamming techniques computer networks flexible efficient performance development							

Query: Learning Rank Algorithm – Reinforcement Learning – Google Scholar							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.5051	0.1051	0.6500	0.7268	0.5370	1.0000
ISA	1	0.8667	0.1476	0.6500	0.9497	0.6827	1.0000
I		71.58%	40.39%	0.00%	30.68%	27.13%	0.00%
GS	2	0.4542	0.1030	0.7500	0.7076	0.5444	1.0000
ISA	2	0.9424	0.1623	0.7500	0.9849	0.7815	1.0000
I		8.74%	9.95%	15.38%	3.71%	14.47%	0.00%
GS	3	0.4596	0.1017	0.7000	0.7064	0.5296	1.0000
ISA	3	0.9430	0.1591	0.7000	0.9852	0.7383	1.0000
I		0.07%	-1.93%	-6.67%	0.03%	-5.53%	0.00%
Relevant Dimensions: Learning Rank Algorithm							
Universe: stochastic application contextual advertising problem model objects framework ranking optimized information retrieval measures normalized efficient margin capture relationship time parameters property consistency unsupervised							

Query: Learning Rank Algorithm – Reinforcement Learning – IEEE Xplore							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
IEEE	1	0.8362	0.1453	0.6000	0.9511	0.5840	1.0000
ISA	1	0.7674	0.0987	0.6000	0.8652	0.6346	0.3333
I		-8.23%	-32.11%	0.00%	-9.03%	8.67%	-66.67%
IEEE	2	0.7599	0.1476	0.7000	0.9288	0.6000	1.0000
ISA	2	0.9871	0.1625	0.7000	0.9989	0.7444	1.0000
I		28.62%	64.71%	16.67%	15.46%	17.32%	200.00%
IEEE	3	0.7160	0.1503	0.8000	0.9166	0.6333	1.0000
ISA	3	0.9797	0.1680	0.8000	0.9955	0.8321	1.0000
I		-0.75%	3.38%	14.29%	-0.34%	11.77%	0.00%
Relevant Dimensions: Learning Rank							
Universe: ranking positive unlabelled examples fuzzy systems knowledge sparse efficient dual primal machine intelligence pattern analysis function signal large scale clustering meta approach neural networks novel reinforcement artificial signal							

Query: Learning Rank Algorithm – Reinforcement Learning – CiteSeerX							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
CSX	1	0.0735	0.0071	0.2000	0.5058	0.0877	0.0556
ISA	1	0.1865	0.0163	0.2000	0.5713	0.1889	0.1111
I		153.76%	128.31%	0.00%	12.95%	115.49%	100.00%
CSX	2	0.2186	0.0417	0.5000	0.5478	0.2802	0.5000
ISA	2	0.7094	0.1252	0.5000	0.8923	0.5136	1.0000
I		280.42%	667.50%	150.0%	56.18%	171.90%	800.00%
CSX	3	0.3980	0.0996	0.6500	0.7255	0.4296	1.0000
ISA	3	0.8695	0.1510	0.6500	0.9642	0.6728	1.0000
I		22.57%	20.69%	30.00%	8.06%	31.01%	0.00%
Relevant Dimensions: Learning Algorithm							
Universe: gradient document recognition multilayer neural networks propagation architecture complex decision classify reinforcement supervised genetic artificial control theory trial error variation selection improved boosting machine improvements							

Query: Learning Rank Algorithm – Reinforcement Learning – Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
MA	1	0.6592	0.1440	0.7000	0.9352	0.5481	1.0000
ISA	1	0.8830	0.1535	0.7000	0.9618	0.7259	1.0000
I		33.95%	6.59%	0.00%	2.85%	32.43%	0.00%
MA	2	0.6090	0.1438	0.9000	0.8752	0.6580	1.0000
ISA	2	0.9971	0.1746	0.9000	0.9994	0.9210	1.0000
I		12.92%	13.73%	28.57%	3.91%	26.87%	0.00%
MA	3	0.5960	0.1463	1.0000	0.8710	0.6840	1.0000
ISA	3	1.0000	0.1799	1.0000	1.0000	1.0000	1.0000
I		0.29%	3.02%	11.11%	0.06%	8.58%	0.00%
Relevant Dimensions: Learning Rank Algorithm							
Universe: stochastic application neural application accurate generalization listwise theoretical machine design parameter tuning sequence labelling reinforcement labelling structured perceptron margins simple linear query dependent variables benchmark							

Query: Google Scholar – Reinforcement Learning – Google Scholar							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.8131	0.1543	0.7500	0.9493	0.7173	1.0000
ISA	1	0.8041	0.1164	0.7500	0.8758	0.7679	0.5000
I		-1.12%	-24.54%	0.00%	-7.74%	7.06%	-50.00%
GS	2	0.6665	0.1314	0.6000	0.8879	0.5556	1.0000
ISA	2	0.9086	0.1501	0.6000	0.9761	0.6358	1.0000
I		13.01%	28.97%	-20.00%	11.45%	-17.20%	100.00%
GS	3	0.4937	0.0839	0.6000	0.7559	0.5321	0.5000
ISA	3	0.9095	0.1502	0.6000	0.9765	0.6358	1.0000
I		0.09%	0.05%	0.00%	0.03%	0.00%	0.00%
Relevant Dimensions: Google Scholar							
Universe: generation indexes locating potentially relevant subject identifying published important feature source bibliometrics knowledge particular social index reports impact factors alternative metric measure systematic comparison design methodology							

Query: Google Scholar – Reinforcement Learning – IEEE Xplore							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
IEEE	1	1.0208	0.1042	0.2000	1.0000	0.2395	1.0000
ISA	1	1.0000	0.1042	0.2000	1.0000	0.2395	1.0000
I		-2.04%	0.00%	0.00%	0.00%	0.00%	0.00%
IEEE	2	0.7233	0.0991	0.2500	0.8840	0.2444	1.0000
ISA	2	0.7669	0.1014	0.2500	0.9048	0.2753	1.0000
I		-23.31%	-2.62%	25.00%	-9.52%	14.95%	0.00%
IEEE	3	0.7941	0.1116	0.3000	0.9240	0.3025	1.0000
ISA	3	1.0000	0.1225	0.3000	1.0000	0.3519	1.0000
I		30.39%	20.76%	20.00%	10.52%	27.80%	0.00%
Relevant Dimensions: Google Scholar							
Universe: ranking algorithm impact citation counts empirical research challenges information age technology generations patterns service oriented system engineering open access journals analysis promotion resources effect cooperation study science							

Query: Google Scholar – Reinforcement Learning – CiteSeerX							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
CSX	1	0.5065	0.0778	0.7500	0.7464	0.6506	0.3333
ISA	1	0.9154	0.1600	0.7500	0.9758	0.7753	1.0000
I		80.75%	105.66%	0.00%	30.74%	19.17%	200.00%
CSX	2	0.5069	0.0824	0.8500	0.7546	0.6765	0.3333
ISA	2	0.9682	0.1702	0.8500	0.9929	0.8691	1.0000
I		5.76%	6.42%	13.33%	1.75%	12.10%	0.00%
CSX	3	0.4883	0.0796	0.8000	0.7466	0.6358	0.3333
ISA	3	0.9702	0.1675	0.8000	0.9933	0.8284	1.0000
I		0.21%	-1.63%	-5.88%	0.05%	-4.69%	0.00%
Relevant Dimensions: Google Scholar							
Universe: ranking algorithm introductive overview research progress challenges academic search engines articles performed reverse engineering present results coverage field evaluates databases index subject area impact empirical study journal							

Query: Google Scholar – Reinforcement Learning – Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
MA	1	0.4822	0.1035	0.8500	0.7066	0.6198	1.0000
ISA	1	0.8472	0.1570	0.8500	0.9429	0.8407	1.0000
I		75.69%	51.66%	0.00%	33.45%	35.66%	0.00%
MA	2	0.5255	0.1177	0.9000	0.7550	0.6444	1.0000
ISA	2	0.9809	0.1737	0.9000	0.9957	0.9148	1.0000
I		15.78%	10.65%	5.88%	5.60%	8.81%	0.00%
MA	3	0.5255	0.1177	0.9000	0.7550	0.6444	1.0000
ISA	3	0.9809	0.1737	0.9000	0.9957	0.9148	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: Google Scholar							
Universe: citations web multi discipline exploratory analysis data gathering method basis compare traditional patterns articles open access coverage field databases reference selected management generation locating features determine frequency							

Query: Travel Search Engine – Reinforcement Learning – Google Scholar							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.2933	0.0536	0.6000	0.6073	0.4086	0.5000
ISA	1	0.9421	0.1520	0.6000	0.9852	0.6469	1.0000
I		221.24%	183.78%	0.00%	62.23%	58.31%	100.00%
GS	2	0.3175	0.0558	0.6000	0.6197	0.4383	0.5000
ISA	2	0.9538	0.1528	0.6000	0.9886	0.6469	1.0000
I		1.25%	0.53%	0.00%	0.34%	0.00%	0.00%
GS	3	0.3175	0.0558	0.6000	0.6197	0.4383	0.5000
ISA	3	0.9538	0.1528	0.6000	0.9886	0.6469	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: Travel Search Engine							
Universe: framework planning important element marketing efforts destination organizations dynamic queries tourism communication technologies growing importance better understand behaviour aspect goal study identify patterns tourist							

Query: Travel Search Engine – Reinforcement Learning – IEEE Xplore							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
IEEE	1	0.2763	0.0369	0.2000	0.5777	0.1852	0.5000
ISA	1	0.4432	0.0639	0.2000	0.5980	0.2074	1.0000
I		60.42%	73.37%	0.00%	3.51%	12.00%	100.00%
IEEE	2	0.2423	0.0379	0.2500	0.5584	0.1901	0.5000
ISA	2	0.9667	0.1125	0.2500	0.9877	0.2951	1.0000
I		118.12%	76.00%	25.00%	65.16%	42.26%	0.00%
IEEE	3	0.2494	0.0435	0.3500	0.5708	0.2432	0.5000
ISA	3	1.0000	0.1296	0.3500	1.0000	0.4062	1.0000
I		3.45%	15.24%	40.00%	1.25%	37.66%	0.00%
Relevant Dimensions: Travel Search Engine							
Universe: emerging vertical distribution newly vulnerable electronic markets planning optimization energy time constrains application internet marketing motel computer automation designing gathering intelligent agent personalized engineering information							

Query: Travel Search Engine – Reinforcement Learning – CiteSeerX							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
CSX	1	0.2500	0.0125	0.0500	1.0000	0.0580	0.2500
ISA	1	1.0000	0.0500	0.0500	1.0000	0.0617	1.0000
I		300.00%	300.00%	0.00%	0.00%	6.38%	300.00%
CSX	2	0.2500	0.0125	0.0500	1.0000	0.0580	0.2500
ISA	2	1.0000	0.0500	0.0500	1.0000	0.0617	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CSX	3	0.2500	0.0125	0.0500	1.0000	0.0580	0.2500
ISA	3	1.0000	0.0500	0.0500	1.0000	0.0617	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Relevant Dimensions: Travel Search Engine							
Universe: designing ranking algorithms systems hotels mining user generated							

Query: Travel Search Engine – Reinforcement Learning – Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
MA	1	0.4429	0.0924	0.4000	0.7393	0.3160	1.0000
ISA	1	0.3947	0.0463	0.4000	0.6616	0.3975	0.2500
I		-10.88%	-49.88%	0.00%	-10.51%	25.78%	-75.00%
MA	2	0.4045	0.1016	0.6500	0.7329	0.4296	1.0000
ISA	2	0.9269	0.1546	0.6500	0.9806	0.6901	1.0000
I		134.84%	233.99%	62.50%	48.22%	73.60%	300.00%
MA	3	0.4061	0.1000	0.6000	0.7325	0.4062	1.0000
ISA	3	0.8733	0.1446	0.6000	0.9516	0.6395	1.0000
I		-5.79%	-6.47%	-7.69%	-2.96%	-7.33%	0.00%
Relevant Dimensions: Travel Search Engine							
Universe: emerging role vertical distribution vulnerable electronic markets perspective digital intermediaries sector agents online increasing vulnerability technologies design evaluation tourism potential change landscape innovation optimization limitations							

Average Values – Gradient Descent – Google Scholar – 3 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.5164	0.0963	0.5833	0.7632	0.4811	0.7222
ISA	1	0.7793	0.1370	0.5833	0.9194	0.5996	1.0000
I		50.92%	42.24%	0.00%	20.46%	24.64%	38.46%
Web	2	0.3465	0.0633	0.5667	0.6489	0.4066	0.5278
ISA	2	0.6395	0.0961	0.5667	0.8106	0.5737	0.5833
I		-17.93%	-29.86%	-2.86%	-11.83%	-4.32%	-41.67%
Web	3	0.3266	0.0582	0.5333	0.6348	0.3774	0.5000
ISA	3	0.5969	0.0885	0.5333	0.7813	0.5358	0.5833
I		-6.66%	-7.97%	-5.88%	-3.62%	-6.60%	0.00%

Average Values – Gradient Descent – IEEE Xplore – 3 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.2322	0.0328	0.4833	0.5834	0.3202	0.2004
ISA	1	0.5712	0.0911	0.4833	0.7679	0.4807	0.7333
I		145.94%	177.98%	0.00%	31.62%	50.13%	265.94%
Web	2	0.3156	0.0518	0.6000	0.6143	0.4342	0.4417
ISA	2	0.7820	0.1342	0.6000	0.8927	0.6222	1.0000
I		36.91%	47.27%	24.14%	16.25%	29.45%	36.36%
Web	3	0.3811	0.0736	0.7000	0.6451	0.5119	0.7333
ISA	3	0.8026	0.1420	0.7000	0.9047	0.7086	1.0000
I		2.63%	5.81%	16.67%	1.34%	13.89%	0.00%

Average Values – Gradient Descent – Cite SeerX – 3 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.5874	0.1025	0.5167	0.8139	0.4605	0.8333
ISA	1	0.7232	0.1216	0.5167	0.8613	0.5362	1.0000
I		23.12%	18.58%	0.00%	5.83%	16.44%	20.00%
Web	2	0.5963	0.1137	0.6167	0.7968	0.5292	1.0000
ISA	2	0.7579	0.1337	0.6167	0.8817	0.6284	1.0000
I		4.80%	9.98%	19.35%	2.37%	17.19%	0.00%
Web	3	0.6615	0.1259	0.6333	0.8536	0.5531	1.0000
ISA	3	0.7902	0.1380	0.6333	0.8997	0.6486	1.0000
I		4.27%	3.22%	2.70%	2.04%	3.21%	0.00%

Average Values – Gradient Descent – Microsoft Academic – 3 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.4435	0.0938	0.6333	0.7370	0.4942	0.7778
ISA	1	0.7464	0.1307	0.6333	0.8715	0.6403	1.0000
I		68.32%	39.40%	0.00%	18.25%	29.56%	28.57%
Web	2	0.4745	0.1023	0.7333	0.7598	0.5440	0.7778
ISA	2	0.7757	0.1328	0.7333	0.8928	0.7354	0.8333
I		3.92%	1.56%	15.79%	2.45%	14.85%	-16.67%
Web	3	0.4551	0.0987	0.7500	0.7426	0.5412	0.7778
ISA	3	0.8060	0.1351	0.7500	0.8910	0.7588	0.8333
I		3.91%	1.73%	2.27%	-0.21%	3.19%	0.00%

Average Values – Reinforcement Learning – Google Scholar – 3 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.5372	0.1043	0.6667	0.7611	0.5543	0.8333
ISA	1	0.8709	0.1387	0.6667	0.9369	0.6992	0.8333
I		62.13%	32.92%	0.00%	23.10%	26.13%	0.00%
Web	2	0.4794	0.0967	0.6500	0.7384	0.5128	0.8333
ISA	2	0.9350	0.1551	0.6500	0.9832	0.6881	1.0000
I		7.35%	11.83%	-2.50%	4.94%	-1.59%	20.00%
Web	3	0.4236	0.0805	0.6333	0.6940	0.5000	0.6667
ISA	3	0.9355	0.1540	0.6333	0.9834	0.6737	1.0000
I		0.05%	-0.66%	-2.56%	0.02%	-2.09%	0.00%

Average Values – Reinforcement Learning – IEEE Xplore – 3 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.7111	0.0955	0.3333	0.8429	0.3362	0.8333
ISA	1	0.7369	0.0889	0.3333	0.8211	0.3605	0.7778
I		3.62%	-6.85%	0.00%	-2.60%	7.22%	-6.67%
Web	2	0.5751	0.0949	0.4000	0.7904	0.3449	0.8333
ISA	2	0.9069	0.1255	0.4000	0.9638	0.4383	1.0000
I		23.07%	41.12%	20.00%	17.39%	21.58%	28.57%
Web	3	0.5865	0.1018	0.4833	0.8038	0.3930	0.8333
ISA	3	0.9932	0.1400	0.4833	0.9985	0.5300	1.0000
I		9.52%	11.61%	20.83%	3.60%	20.94%	0.00%

Average Values – Reinforcement Learning – Cite SeerX – 3 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.2767	0.0325	0.3333	0.7507	0.2654	0.2130
ISA	1	0.7006	0.0754	0.3333	0.8491	0.3420	0.7037
I		153.25%	132.25%	0.00%	13.10%	28.84%	230.43%
Web	2	0.3252	0.0455	0.4667	0.7675	0.3383	0.3611
ISA	2	0.8925	0.1151	0.4667	0.9617	0.4815	1.0000
I		27.39%	52.64%	40.00%	13.27%	40.79%	42.11%
Web	3	0.3788	0.0639	0.5000	0.8240	0.3745	0.5278
ISA	3	0.9466	0.1228	0.5000	0.9859	0.5210	1.0000
I		6.05%	6.69%	7.14%	2.51%	8.21%	0.00%

Average Values – Reinforcement Learning – Microsoft Academic – 3 Queries							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
Web	1	0.5281	0.1133	0.6500	0.7937	0.4947	1.0000
ISA	1	0.7083	0.1189	0.6500	0.8554	0.6547	0.7500
I		34.12%	4.97%	0.00%	7.78%	32.36%	-25.00%
Web	2	0.5130	0.1210	0.8167	0.7877	0.5774	1.0000
ISA	2	0.9683	0.1676	0.8167	0.9919	0.8420	1.0000
I		36.71%	40.96%	25.64%	15.95%	28.60%	33.33%
Web	3	0.5092	0.1213	0.8333	0.7862	0.5782	1.0000
ISA	3	0.9514	0.1661	0.8333	0.9825	0.8514	1.0000
I		-1.75%	-0.94%	2.04%	-0.95%	1.12%	0.00%

Query: Random Neural Network – GS IEEE Xplore Cite SeerX Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.8775	0.1440	0.5500	0.9658	0.5852	1.0000
ISA	1	0.6329	0.1158	0.5500	0.7984	0.5309	1.0000
I		-27.88%	-19.56%	0.00%	-17.34%	-9.28%	0.00%
IEEE	1	0.6366	0.1178	0.6500	0.7924	0.6321	1.0000
ISA	1	0.2997	0.0447	0.6500	0.6335	0.4432	0.2000
I		-52.92%	-62.05%	0.00%	-20.05%	-29.88%	-80.00%
CSX	1	0.5241	0.0681	0.3500	0.7368	0.3741	0.5000
ISA	1	0.4852	0.0807	0.3500	0.6544	0.3556	1.0000
I		-7.42%	18.51%	0.00%	-11.18%	-4.95%	100.00%
MA	1	0.6934	0.1222	0.5500	0.8581	0.5630	1.0000
ISA	1	0.4459	0.0459	0.5500	0.6366	0.4444	0.2500
I		-35.69%	-62.43%	0.00%	-25.81%	-21.05%	-75.00%

Query: Web Search Engine – GS IEEE Xplore Cite SeerX Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.5068	0.0885	0.4000	0.6885	0.4012	1.0000
ISA	1	0.2647	0.0383	0.4000	0.6073	0.2975	0.2500
I		-47.78%	-56.68%	0.00%	-11.79%	-25.85%	-75.00%
IEEE	1	0.5014	0.0868	0.3500	0.6971	0.3519	1.0000
ISA	1	0.3385	0.0692	0.3500	0.5779	0.2765	1.0000
I		-32.49%	-20.28%	0.00%	-17.09%	-21.40%	0.00%
CSX	1	0.7763	0.1412	0.6500	0.9222	0.6580	1.0000
ISA	1	0.3573	0.0486	0.6500	0.6613	0.5000	0.1667
I		-53.97%	-65.57%	0.00%	-28.29%	-24.02%	-83.33%
MA	1	0.5355	0.0940	0.4000	0.7239	0.4025	1.0000
ISA	1	0.2732	0.0333	0.4000	0.5987	0.3420	0.2000
I		-48.98%	-64.54%	0.00%	-17.30%	-15.03%	-80.00%

Query: Metasearch Engine – GS IEEE Xplore Cite SeerX Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.6972	0.1249	0.6500	0.8230	0.6481	1.0000
ISA	1	0.5076	0.0891	0.6500	0.7692	0.5383	0.5000
I		-27.20%	-28.66%	0.00%	-6.54%	-16.95%	-50.00%
IEEE	1	0.6288	0.1135	0.6000	0.7806	0.5938	1.0000
ISA	1	0.2709	0.0528	0.6000	0.6001	0.3531	0.5000
I		-56.92%	-53.45%	0.00%	-23.13%	-40.54%	-50.00%
CSX	1	0.7023	0.1225	0.6000	0.8217	0.6099	1.0000
ISA	1	0.4977	0.1032	0.6000	0.7295	0.5259	1.0000
I		-29.13%	-15.73%	0.00%	-11.22%	-13.77%	0.00%
MA	1	0.5592	0.0867	0.6500	0.7651	0.6210	0.5000
ISA	1	0.4329	0.0897	0.6500	0.6612	0.5358	1.0000
I		-22.58%	3.41%	0.00%	-13.59%	-13.72%	100.00%

Query: Learning Rank Algorithm – GS IEEE Xplore Cite SeerX Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.6425	0.1143	0.6500	0.7733	0.6395	1.0000
ISA	1	0.4945	0.0996	0.6500	0.7038	0.5494	1.0000
I		-23.04%	-12.86%	0.00%	-8.98%	-14.09%	0.00%
IEEE	1	0.7189	0.1237	0.5500	0.8656	0.5716	1.0000
ISA	1	0.8913	0.1451	0.5500	0.9706	0.5852	1.0000
I		23.98%	17.28%	0.00%	12.13%	2.38%	0.00%
CSX	1	0.1750	0.0186	0.2000	0.5516	0.1802	0.2000
ISA	1	0.2117	0.0316	0.2000	0.5293	0.1457	0.5000
I		21.00%	69.72%	0.00%	-4.05%	-19.18%	150.00%
MA	1	0.6731	0.1136	0.5000	0.8021	0.5185	1.0000
ISA	1	0.5038	0.1000	0.5000	0.7262	0.4173	1.0000
I		-25.15%	-11.96%	0.00%	-9.47%	-19.52%	0.00%

Query: Google Scholar – GS IEEE Xplore Cite SeerX Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.8946	0.1490	0.6000	0.9715	0.6358	1.0000
ISA	1	0.6924	0.1006	0.6000	0.8359	0.6148	0.5000
I		-22.60%	-32.48%	0.00%	-13.95%	-3.30%	-50.00%
IEEE	1	1.0000	0.1042	0.2000	1.0000	0.2395	1.0000
ISA	1	1.0000	0.1042	0.2000	1.0000	0.2395	1.0000
I		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CSX	1	0.7886	0.1326	0.6500	0.8569	0.6728	1.0000
ISA	1	0.6391	0.1256	0.6500	0.8471	0.6099	1.0000
I		-18.95%	-5.32%	0.00%	-1.15%	-9.36%	0.00%
MA	1	0.6992	0.1163	0.6000	0.7900	0.6160	1.0000
ISA	1	0.6413	0.1274	0.6000	0.8701	0.5617	1.0000
I		-8.28%	9.57%	0.00%	10.14%	-8.82%	0.00%

Query: Travel Search Engine – GS IEEE Xplore Cite SeerX Microsoft Academic							
Search	Iteration	MAP	TSAP	P	NDCG	Q	MRR- ERR
GS	1	0.3308	0.0458	0.3000	0.6049	0.2938	0.5000
ISA	1	0.6052	0.0930	0.3000	0.7562	0.2951	1.0000
I		82.96%	103.12%	0.00%	25.02%	0.42%	100.00%
IEEE	1	0.3207	0.0351	0.1500	0.6085	0.1568	0.5000
ISA	1	0.5076	0.0608	0.1500	0.6167	0.1642	1.0000
I		58.26%	73.29%	0.00%	1.35%	4.72%	100.00%
CSX	1	0.2159	0.0170	0.1000	0.6445	0.1074	0.2500
ISA	1	1.0000	0.0750	0.1000	1.0000	0.1222	1.0000
I		363.16%	340.00%	0.00%	55.15%	13.79%	300.00%
MA	1	0.7500	0.0896	0.2000	0.8688	0.2309	1.0000
ISA	1	0.2438	0.0227	0.2000	0.6021	0.2037	0.1667
I		-67.50%	-74.66%	0.00%	-30.70%	-11.76%	-83.33%

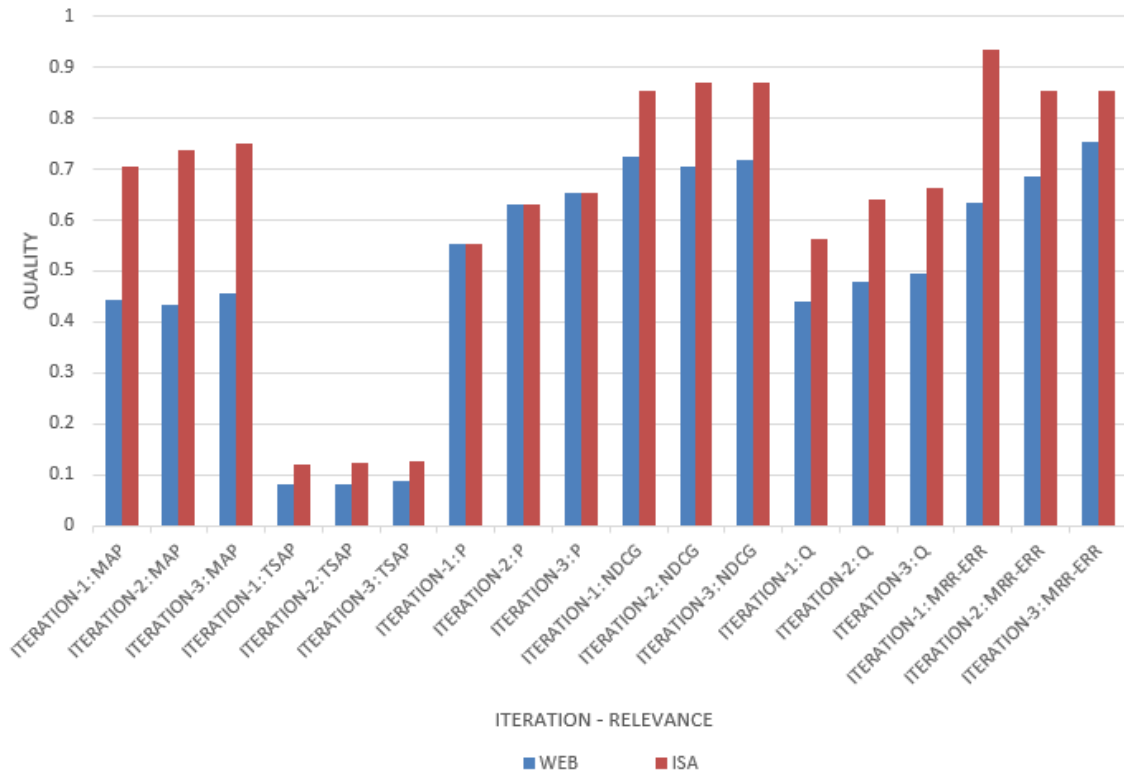


Figure 96: Database Learning evaluation – Gradient Descent - Average

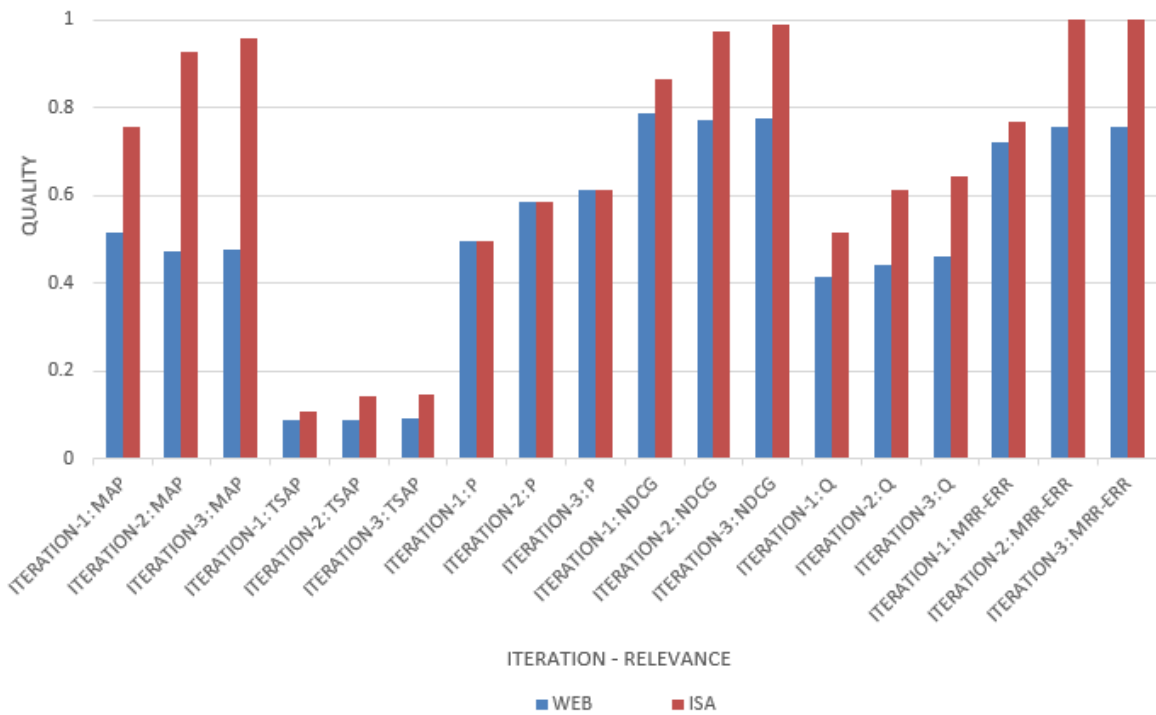


Figure 97: Database Learning evaluation – Reinforcement Learning - Average

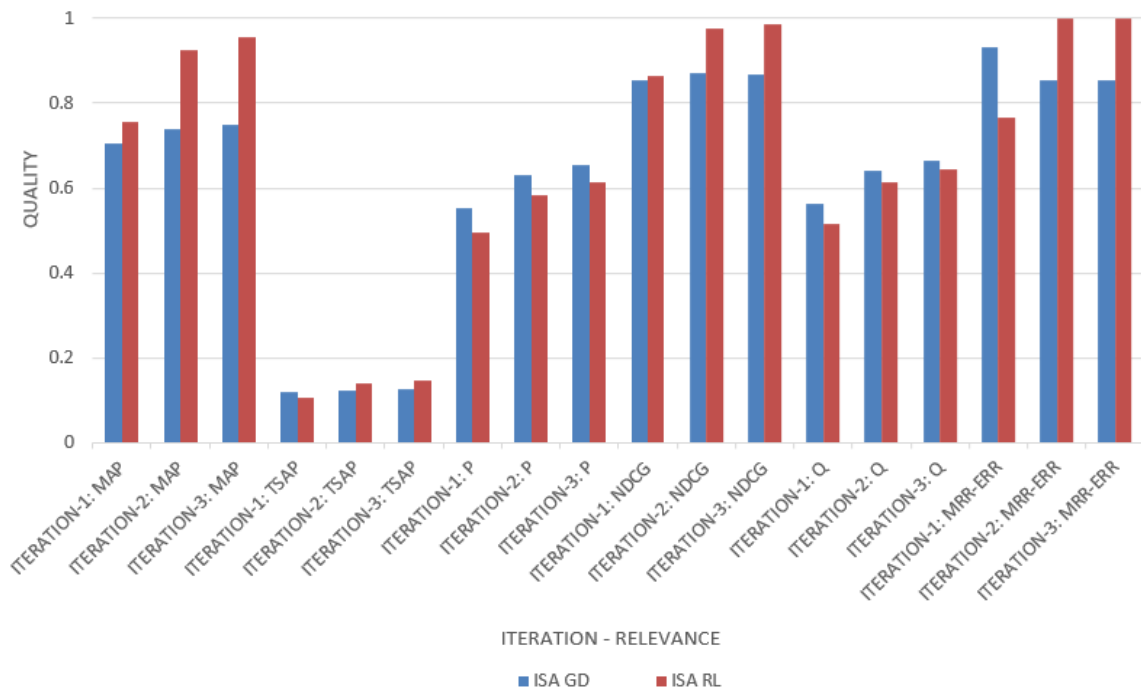


Figure 98: Database Learning evaluation – Evaluation between learnings

E User Evaluation – Deep Learning

Q Google	Q Google Cluster	Q Google Cluster Final	Q Yahoo	Q Yahoo Cluster	Q Yahoo Cluster Final	Q Ask	Q Ask Cluster	Q Ask Cluster Final	Q Bing	Q Bing Cluster	Q Bing Cluster Final	Q Lycos	Q Lycos Cluster	Q Lycos Cluster Final
Random Neural Network														
0.58	0.32	0.89	0.47	0.79	1.00	0.74	0.47	0.68	0.47	0.68	0.79	0.79	0.79	0.58
Art Exhibitions London														
0.74	0.63	1.00	0.37	0.79	0.89	0.79	0.74	0.79	0.29	1.00	1.00	0.37	0.58	0.79
Art Galleries Berlin														
0.53	0.74	0.79	0.95	0.74	1.00	0.68	0.79	0.89	0.71	0.88	1.00	0.95	0.58	1.00
Night Clubs London														
0.84	0.42	0.68	0.89	0.37	1.00	0.84	0.79	0.68	0.80	0.20	1.00	0.53	0.58	0.37
Night Clubs Berlin														
1.00	0.63	0.68	0.58	0.16	1.00	0.89	0.53	0.47	0.93	0.40	0.87	0.89	0.58	1.00
Vegetarian restaurant Stoke newington														
0.47	0.74	0.68	0.68	0.89	0.89	0.89	0.84	1.00	0.84	0.89	0.79	0.68	1.00	1.00
Techno Festivals Europe														
1.00	0.95	0.89	0.74	0.89	1.00	0.53	0.95	1.00	0.74	0.89	1.00	0.74	0.84	1.00
Indie Rock London														
0.74	0.37	1.00	0.53	0.63	0.68	0.84	0.89	1.00	0.68	0.47	0.79	0.63	0.63	1.00
Flights Tokyo														
0.84	1.00	1.00	0.84	0.63	0.89	0.79	1.00	0.89	0.85	0.54	0.38	0.84	0.68	0.79
Gentrification World														
1.00	0.53	1.00	0.58	0.89	0.79	0.95	0.84	0.89	0.58	0.84	0.89	0.89	0.89	1.00
Best documentaries 2016														
0.89	0.21	0.89	0.58	0.58	0.74	0.47	0.32	0.89	0.63	0.68	0.37	0.42	1.00	1.00
Book shops soho														
0.68	0.89	1.00	0.74	0.79	1.00	0.84	0.68	0.47	0.65	0.53	0.76	0.47	1.00	0.89
Boutique hotels mexico city														
0.68	0.58	1.00	0.26	0.74	0.89	0.37	0.84	0.89	0.77	0.23	1.00	0.68	0.68	0.68
Brownie cafe London														
0.84	0.74	0.89	1.00	0.37	0.79	0.79	0.58	1.00	0.79	0.74	0.89	0.26	0.58	0.79
Haute couture catwalks 2016														
0.59	0.35	0.88	0.26	0.21	0.79	0.68	0.47	0.47	0.74	0.32	0.26	0.26	0.58	0.47

Appendix

Q Google	Q Google Cluster	Q Google Cluster Final	Q Yahoo	Q Yahoo Cluster	Q Yahoo Cluster Final	Q Ask	Q Ask Cluster	Q Ask Cluster Final	Q Bing	Q Bing Cluster	Q Bing Cluster Final	Q Lycos	Q Lycos Cluster	Q Lycos Cluster Final
Mid Century Patterns														
0.77	0.54	0.69	0.79	0.58	0.58	0.68	0.53	1.00	0.58	0.95	0.89	0.68	0.53	0.58
Recycling Center London														
0.47	0.53	0.47	0.63	0.32	1.00	0.58	0.37	0.47	0.58	0.68	1.00	0.63	0.74	1.00
Vintage Arm Chair														
0.35	0.41	0.88	0.47	0.47	0.63	0.42	0.47	1.00	0.33	0.27	0.73	0.53	0.26	1.00
Wedding Design Gifts Shops														
0.53	0.41	0.41	0.37	0.74	1.00	0.32	0.37	0.89	0.23	0.85	1.00	0.16	0.74	0.89
Weekend Paris Break														
0.79	0.47	0.79	0.16	0.58	1.00	0.68	0.74	0.89	0.62	0.69	0.85	0.26	0.32	0.89
Tory Party Conference														
0.54	0.54	0.85	0.53	0.84	1.00	0.37	0.47	0.79	0.63	0.53	0.58	0.58	0.79	0.63
restaurants Forest Gate														
0.74	0.32	0.89	0.47	0.68	0.58	0.42	0.42	1.00	0.84	0.42	0.89	0.47	0.74	0.47
compare energy prices														
0.58	0.37	0.79	0.79	0.32	0.89	0.89	0.79	0.68	0.69	0.46	0.85	0.89	0.79	0.79
mid century furniture														
0.29	0.53	0.65	0.21	0.42	0.58	0.68	0.47	0.89	0.71	0.59	0.18	0.58	0.89	0.79
reclaimed wood London														
0.32	0.26	0.58	0.89	0.53	0.79	0.42	0.16	1.00	0.58	0.47	0.89	0.63	0.37	0.58
Best dinner date London														
1.00	0.53	0.84	0.26	0.95	1.00	0.37	0.63	0.37	0.95	0.63	1.00	0.63	0.63	0.89
Best holiday destinations														
0.53	0.74	0.68	0.74	0.47	0.37	0.32	0.32	1.00	0.53	0.67	0.73	0.89	0.79	0.89
Holiday deals packages														
0.79	0.74	1.00	0.74	0.79	1.00	0.68	1.00	1.00	0.62	0.46	0.69	0.95	0.74	0.58
Best rooftop bars London														
0.84	0.63	1.00	0.63	0.63	1.00	0.89	0.32	0.58	0.63	0.53	0.26	0.42	0.32	0.89
cheap flights Shanghai														
0.68	0.53	1.00	0.79	0.89	0.68	0.42	0.95	0.68	0.47	0.67	0.87	0.74	0.89	0.68

Appendix

Q Google	Q Google Cluster	Q Google Cluster Final	Q Yahoo	Q Yahoo Cluster	Q Yahoo Cluster Final	Q Ask	Q Ask Cluster	Q Ask Cluster Final	Q Bing	Q Bing Cluster	Q Bing Cluster Final	Q Lycos	Q Lycos Cluster	Q Lycos Cluster Final
Film clubs Edinburgh														
0.32	0.74	0.26	0.74	0.53	0.37	0.37	0.74	0.58	0.63	0.53	0.42	0.79	0.53	0.37
fundraising jobs arts heritage Edinburgh														
0.89	0.95	0.95	0.37	0.32	0.89	1.00	0.89	0.79	0.79	0.74	0.63	1.00	0.84	1.00
Holiday cottages rent Stoke on Trent 10 people														
0.89	0.84	0.79	0.84	0.74	1.00	0.68	0.89	1.00	0.55	0.64	1.00	0.42	0.95	0.79
Home remedies sore throats														
0.74	0.68	0.79	0.84	0.79	1.00	0.79	0.89	1.00	0.79	0.58	0.89	0.79	0.58	0.79
Volunteering opportunities Edinburgh														
0.47	0.63	0.79	0.95	0.79	0.89	0.74	0.58	0.68	0.53	0.58	0.89	0.58	0.63	0.37
Brexit investments														
0.59	0.88	0.88	0.68	0.37	0.37	0.47	0.74	0.89	0.63	0.42	0.26	0.84	0.74	0.68
Bristol breweries														
0.95	0.63	0.79	0.68	0.74	0.89	0.84	0.79	0.68	0.68	0.68	0.89	0.68	0.74	0.89
Harmonica shops London														
0.42	0.53	1.00	0.68	0.74	0.68	0.53	0.47	0.16	0.47	0.47	1.00	0.68	0.84	0.47
Narrow boats sale London														
1.00	0.53	0.89	0.58	0.53	0.68	0.47	0.79	1.00	0.74	0.53	0.58	0.95	0.58	0.79
How get Irish passport														
0.63	0.84	0.89	0.95	0.53	0.89	0.68	0.68	0.63	1.00	0.63	0.68	0.42	0.37	0.89
Best Pizza New York														
0.84	0.32	0.68	0.68	0.74	1.00	0.79	0.74	0.58	0.74	0.95	0.89	0.84	0.95	1.00
Islington History														
0.41	0.82	0.76	0.58	0.79	0.37	0.74	0.74	1.00	0.42	0.84	0.68	0.53	0.68	0.37
Southern rail overground														
0.63	0.53	0.58	0.58	0.58	0.79	0.68	0.37	0.89	0.58	0.68	0.89	0.58	0.58	0.79
US presidential election forecast														
0.58	0.58	1.00	0.32	0.58	0.37	0.89	0.89	1.00	0.84	0.63	0.58	0.74	0.63	0.37
Yorkshire dales walks														
0.79	0.47	1.00	0.32	0.32	1.00	0.63	0.26	1.00	0.84	0.68	0.58	0.58	0.79	0.47

Appendix

Q Google Cluster Learning	Improvement	Q Yahoo Cluster Learning	Improvement	Q Ask Cluster Learning	Improvement	Q Bing Cluster Learning	Improvement	Q Lycos Cluster Learning	Improvement
Random Neural Network									
0.32	0.00%	1.00	26.67%	0.53	11.11%	0.63	-7.69%	0.79	0.00%
Art Exhibitions London									
0.79	25.00%	1.00	26.67%	0.58	-21.4%	1.00	0.00%	0.68	18.18%
Art Galleries Berlin									
0.84	14.29%	0.47	-35.71%	0.32	-60.0%	0.65	-26.67%	1.00	72.73%
Night Clubs London									
0.68	62.50%	0.74	100.00%	0.16	-80.0%	1.00	400.00%	0.37	-36.36%
Night Clubs Berlin									
0.84	33.33%	0.79	400.00%	0.53	0.00%	0.87	116.67%	1.00	72.73%
Vegetarian restaurant Stoke Newington									
0.58	-21.43%	0.95	5.88%	0.68	-18.7%	0.95	5.88%	1.00	0.00%
Techno Festivals Europe									
0.21	-77.78%	1.00	11.76%	0.42	-55.6%	1.00	11.76%	1.00	18.75%
Indie Rock London									
0.42	14.29%	0.63	0.00%	0.68	-23.5%	0.32	-33.33%	1.00	58.33%
Flights Tokyo									
1.00	0.00%	0.68	8.33%	0.26	-73.7%	0.54	0.00%	0.84	23.08%
Gentrification World									
0.82	55.56%	0.68	-23.53%	0.53	-37.5%	0.95	12.50%	1.00	11.76%
Best documentaries 2016									
0.21	0.00%	0.58	0.00%	0.53	66.67%	0.68	0.00%	1.00	0.00%
Book shops soho									
0.79	-11.76%	1.00	26.67%	0.74	7.69%	0.59	11.11%	0.42	-57.89%
Boutique hotels mexico city									
0.58	0.00%	0.42	-42.86%	0.26	-68.7%	1.00	333.33%	0.47	-30.77%
Brownie cafe London									
0.47	-35.71%	0.68	85.71%	1.00	72.73%	0.42	-42.86%	0.32	-45.45%
Haute couture catwalks 2016									
0.88	150.00%	0.68	225.00%	0.74	55.56%	0.74	133.33%	0.53	-9.09%

Appendix

Q Google Cluster Learning	Improvement	Q Yahoo Cluster Learning	Improvement	Q Ask Cluster Learning	Improvement	Q Bing Cluster Learning	Improvement	Q Lycos Cluster Learning	Improvement
Mid Century Patterns									
0.54	0.00%	0.74	27.27%	1.00	90.00%	0.21	-77.78%	0.74	40.00%
Recycling Center London									
0.63	20.00%	0.84	166.67%	0.63	71.43%	1.00	46.15%	0.58	-21.43%
Vintage Arm Chair									
0.65	57.14%	0.74	55.56%	0.42	-11.11%	0.87	225.00%	1.00	280.00%
Wedding Design Gifts Shops									
0.59	42.86%	0.58	-21.43%	0.37	0.00%	1.00	18.18%	0.47	-35.71%
Weekend Paris Break									
0.68	44.44%	1.00	72.73%	0.42	-42.86%	0.54	-22.22%	0.63	100.00%
Tory Party Conference									
0.62	14.29%	1.00	18.75%	0.37	-22.22%	0.53	0.00%	0.53	-33.33%
restaurants Forest Gate									
0.84	166.67%	0.58	-15.38%	1.00	137.50%	0.74	75.00%	0.47	-35.71%
compare energy prices									
0.58	57.14%	0.89	183.33%	0.37	-53.33%	0.62	33.33%	0.42	-46.67%
mid century furniture									
0.71	33.33%	0.42	0.00%	0.89	88.89%	0.18	-70.00%	0.26	-70.59%
reclaimed wood London									
0.79	200.00%	0.63	20.00%	1.00	533.33%	0.47	0.00%	0.53	42.86%
Best dinner date London									
0.63	20.00%	0.16	-83.33%	0.47	-25.00%	1.00	58.33%	0.84	33.33%
Best holiday destinations									
0.37	-50.00%	0.58	22.22%	1.00	216.67%	0.60	-10.00%	0.37	-53.33%
Holiday deals packages									
0.74	0.00%	0.74	-6.67%	1.00	0.00%	0.38	-16.67%	0.68	-7.14%
Best rooftop bars London									
0.58	-8.33%	1.00	58.33%	0.32	0.00%	0.53	0.00%	0.26	-16.67%
cheap flights Shanghai									
1.00	90.00%	0.21	-76.47%	0.16	-83.33%	0.60	-10.00%	0.26	-70.59%

Appendix

Q Google Cluster Learning	Improvement	Q Yahoo Cluster Learning	Improvement	Q Ask Cluster Learning	Improvement	Q Bing Cluster Learning	Improvement	Q Lycos Cluster Learning	Improvement
Film clubs Edinburgh									
0.68	-7.14%	0.42	-20.00%	0.58	-21.43%	0.42	-20.00%	0.42	-20.00%
fundraising jobs arts heritage Edinburgh									
0.89	-5.56%	0.53	66.67%	0.63	-29.41%	0.68	-7.14%	1.00	18.75%
Holiday cottages rent Stoke on Trent 10 people									
0.58	31.25%	0.58	-21.43%	0.47	-47.06%	1.00	57.14%	0.26	-72.22%
Home remedies sore throats									
0.74	7.69%	0.89	13.33%	1.00	11.76%	0.53	-9.09%	0.58	0.00%
Volunteering opportunities Edinburgh									
0.47	25.00%	0.47	-40.00%	0.68	18.18%	0.89	54.55%	0.47	-25.00%
Brexit investments									
0.35	60.00%	1.00	171.43%	0.89	21.43%	0.84	100.00%	0.74	0.00%
Bristol breweries									
0.95	50.00%	0.89	21.43%	0.63	-20.00%	0.95	38.46%	0.89	21.43%
Harmonica shops London									
0.95	80.00%	0.58	-21.43%	0.84	77.78%	1.00	111.11%	0.53	-37.50%
Narrow boats sale London									
0.58	10.00%	0.58	10.00%	1.00	26.67%	0.26	-50.00%	0.53	-9.09%
How get Irish passport									
0.37	56.25%	0.74	40.00%	0.58	-15.38%	0.47	-25.00%	0.89	142.86%
Best Pizza New York									
0.32	0.00%	0.79	7.14%	0.53	-28.57%	0.95	0.00%	1.00	5.56%
Islington History									
0.71	14.29%	0.37	-53.33%	1.00	35.71%	0.32	-62.50%	0.26	-61.54%
Southern rail overground									
0.89	70.00%	0.58	0.00%	0.68	85.71%	0.89	30.77%	0.47	-18.18%
US presidential election forecast									
0.89	54.55%	0.47	-18.18%	1.00	11.76%	0.53	-16.67%	0.58	-8.33%
Yorkshire dales walks									
0.74	55.56%	1.00	216.67%	1.00	280.00%	0.53	-23.08%	0.47	-40.00%

Appendix

Quality General	Quality General Final	Quality Management $\Phi^{1/4}$	Improvement	Quality Management $\Phi^{1/2}$	Improvement	Quality Management Φ	Improvement	Quality Management Φ^2	Improvement	Quality Management Φ^4	Improvement
Random Neural Network											
0.21	0.23	0.23	-0.34%	0.23	-0.34%	0.23	0.00%	0.23	-1.02%	0.23	-2.04%
Art Exhibitions London											
0.40	0.56	0.54	-2.92%	0.54	-3.07%	0.55	-2.19%	0.54	-2.63%	0.55	-1.75%
Art Galleries Berlin											
0.24	0.35	0.34	-3.50%	0.34	-3.50%	0.34	-3.50%	0.34	-3.73%	0.34	-4.20%
Night Clubs London											
0.25	0.29	0.29	-1.73%	0.29	-1.73%	0.29	-0.58%	0.29	-0.29%	0.29	-0.29%
Night Clubs Berlin											
0.24	0.35	0.34	-2.88%	0.34	-2.64%	0.33	-5.53%	0.33	-7.93%	0.28	-21.39%
Vegetarian restaurant Stoke newington											
0.25	0.29	0.27	-5.72%	0.27	-5.18%	0.27	-4.90%	0.27	-5.18%	0.28	-2.18%
Techno Festivals Europe											
0.26	0.32	0.31	-1.24%	0.31	-0.99%	0.31	-0.99%	0.31	-1.24%	0.31	-1.49%
Indie Rock London											
0.22	0.27	0.28	1.74%	0.28	1.74%	0.28	2.90%	0.28	2.32%	0.27	0.58%
Flights Tokyo											
0.25	0.27	0.27	-0.33%	0.26	-2.31%	0.25	-5.61%	0.21	-22.77%	0.17	-36.30%
Gentrification World											
0.25	0.31	0.30	-4.47%	0.29	-5.26%	0.29	-5.53%	0.28	-9.21%	0.25	-18.95%
Best documentaries 2016											
0.28	0.21	0.21	0.38%	0.21	0.38%	0.21	0.38%	0.21	3.44%	0.23	14.12%
Book shops soho											
0.28	0.32	0.29	-10.63%	0.29	-11.39%	0.28	-12.41%	0.26	-19.75%	0.19	-40.76%
Boutique hotels mexico city											
0.23	0.23	0.22	-3.09%	0.22	-3.47%	0.22	-3.09%	0.22	-4.25%	0.22	-3.86%
Brownie cafe London											
0.21	0.30	0.30	0.79%	0.30	1.06%	0.30	0.79%	0.30	1.85%	0.30	1.32%
Haute couture catwalks 2016											
0.17	0.20	0.21	6.94%	0.21	7.35%	0.22	8.57%	0.22	8.57%	0.22	8.98%

Appendix

Quality General	Quality General Final	Quality Management $\Phi^{1/4}$	Improvement	Quality Management $\Phi^{1/2}$	Improvement	Quality Management Φ	Improvement	Quality Management Φ^2	Improvement	Quality Management Φ^4	Improvement
Mid Century Patterns											
0.22	0.24	0.25	2.20%	0.25	2.20%	0.25	2.20%	0.25	1.47%	0.24	0.00%
Recycling Center London											
0.20	0.18	0.19	8.00%	0.19	7.11%	0.18	2.67%	0.17	-2.67%	0.17	-4.44%
Vintage Arm Chair											
0.17	0.29	0.30	2.11%	0.30	2.11%	0.30	2.11%	0.30	2.11%	0.30	2.42%
Wedding Design Gifts Shops											
0.20	0.19	0.23	21.95%	0.23	20.49%	0.22	16.10%	0.21	11.71%	0.21	8.29%
Weekend Paris Break											
0.33	0.44	0.44	1.42%	0.44	1.42%	0.44	1.01%	0.44	1.42%	0.44	-0.40%
Tory Party Conference											
0.21	0.27	0.29	5.84%	0.29	5.52%	0.29	4.55%	0.28	2.60%	0.28	0.97%
restaurants Forest Gate											
0.21	0.25	0.26	2.80%	0.25	-1.87%	0.24	-4.98%	0.22	-	0.16	-
compare energy prices											
0.21	0.24	0.24	-0.74%	0.24	0.00%	0.24	-0.74%	0.24	-0.74%	0.23	-3.35%
mid century furniture											
0.33	0.22	0.23	1.52%	0.23	0.76%	0.22	0.38%	0.22	-3.04%	0.22	-3.80%
reclaimed wood London											
0.16	0.20	0.22	8.14%	0.21	4.65%	0.21	1.55%	0.19	-6.20%	0.20	0.08%
Best dinner date London											
0.18	0.34	0.34	0.00%	0.34	0.00%	0.34	-0.23%	0.34	-0.46%	0.34	-0.46%
Best holiday destinations											
0.21	0.22	0.23	6.32%	0.23	6.32%	0.23	7.91%	0.23	9.09%	0.24	11.46%
Holiday deals packages											
0.30	0.32	0.30	-6.42%	0.30	-5.87%	0.30	-5.03%	0.30	-3.91%	0.31	-1.40%
Best rooftop bars London											
0.18	0.21	0.22	4.07%	0.22	2.59%	0.22	1.85%	0.22	2.59%	0.22	2.96%
cheap flights Shanghai											
0.32	0.34	0.34	0.25%	0.34	0.00%	0.34	-0.50%	0.34	-1.99%	0.30	-

Appendix

Quality General	Quality General Final	Quality Management $\Phi^{1/4}$	Improvement	Quality Management $\Phi^{1/2}$	Improvement	Quality Management Φ	Improvement	Quality Management Φ^2	Improvement	Quality Management Φ^4	Improvement
Film clubs Edinburgh											
0.20	0.15	0.15	3.72%	0.15	3.72%	0.16	10.11%	0.17	12.23%	0.17	13.30%
fundraising jobs arts heritage Edinburgh											
0.23	0.25	0.25	0.62%	0.26	0.93%	0.26	1.55%	0.26	3.41%	0.27	6.19%
Holiday cottages rent Stoke on Trent 10 people											
0.27	0.31	0.30	-3.59%	0.30	-3.29%	0.30	-2.99%	0.30	-2.10%	0.31	0.00%
Home remedies sore throats											
0.26	0.32	0.32	-1.22%	0.31	-2.68%	0.31	-3.41%	0.31	-3.41%	0.30	-7.06%
Volunteering opportunities Edinburgh											
0.21	0.28	0.28	-2.49%	0.28	-2.49%	0.28	-2.49%	0.28	-1.94%	0.28	0.28%
Brexit investments											
0.21	0.24	0.25	3.68%	0.25	4.01%	0.25	2.68%	0.25	2.01%	0.25	0.67%
Bristol breweries											
0.30	0.22	0.26	18.71%	0.26	18.71%	0.26	19.42%	0.26	19.42%	0.26	20.50%
Harmonica shops London											
0.21	0.23	0.24	5.56%	0.24	4.51%	0.23	2.43%	0.23	0.35%	0.20	-11.81%
Narrow boats sale London											
0.20	0.23	0.26	14.58%	0.26	14.58%	0.26	14.24%	0.26	14.93%	0.27	19.44%
How get Irish passport											
0.21	0.20	0.21	3.14%	0.20	2.35%	0.20	0.78%	0.19	-3.14%	0.20	-2.35%
Best Pizza New York											
0.29	0.36	0.36	0.66%	0.36	0.22%	0.36	-0.44%	0.35	-2.64%	0.31	-12.53%
Islington History											
0.35	0.36	0.35	-0.91%	0.35	-0.91%	0.35	-0.91%	0.35	-0.91%	0.35	-0.91%
Southern rail overground											
0.19	0.25	0.29	13.40%	0.28	12.46%	0.28	10.90%	0.27	8.72%	0.26	3.43%
US presidential election forecast											
0.21	0.27	0.26	-5.46%	0.25	-8.62%	0.23	-14.37%	0.22	-20.98%	0.21	-21.55%
Yorkshire dales walks											
0.24	0.33	0.31	-6.37%	0.30	-8.49%	0.29	-12.50%	0.27	-19.58%	0.19	-41.51%