# Abstract

The field of automated text was pioneered by Hans Peter Luhn a researcher at IBM, with the abundance of information today it is difficult to assess the relevance of a document without reading a large chunk of text. Summaries indicate the most salient points of the document and the overall topic it addresses.

To generate quality summaries the initial writer needs to summarise his document or an unbiased third-party reads the document and creates a condensed writeup.

With the number of documents available and with the number increasing this is infeasible. Machine learning offers a way to automate summarization of documents quickly and accurately in large corpora of data.

# Introduction

Machine learning is a type of computational algorithmics where the algorithm improves itself as it is run over the corpus of data. Machine learning for automated text summarization is split into main approaches abstractive and extractive, abstractive focuses on the individual words and re-generates text while extractive instead ranks sentences by prominence.

The results of the summarization are split into two types query based where a topical query is given the machine learning algorithm summarizes based on it and generic summaries that simply condense the most important details of the document.

Research in automated summarization has come to the consensus that summarization models work best on corpora with similar topical content in both results topical similarity helps the algorithm learn.

# Papers

## Paper 1

Samidha et al. propose an Extractive Text summarization model using Restricted Boltzmann Machine (RBM) in combination with fuzzy logic to generate a generic meaningful summary of large single documents. The model takes several word and sentence features which are used to generate two summaries with Fuzzy logic and RBM these summaries are then combined to make the final summary.

### Models

The feature extraction paradigm is based upon Luhn’s work which states the initial and final 7% of a document are the most meaningful and that longer sentences are more meaningful than shorter sentences.

The text is broken down into an ordered matrix of sentences and a bag of words, feature extraction processes are then carried out on the tokenized document: Sentence position rank, Sentence length ratio, Numerical token ratio, TF-ISF (Term frequency- Inverse Sentence Frequency),Cosine similarity to centroid (TF-ISF), Bi-Gram/Tri-gram (Calculated using the NLTK libraries), Proper Noun ratio, Thematic Word ratio.

These features are then combined to form a sentence feature matrix, the matrix is then normalized by dividing the values by the largest value and used as an input for the RBM.

RBM’s are stochastic generative networks where there is a clear division of visible and hidden nodes into two layers, where the visible notes are only connected to nodes in the next layer and not parallel nodes. The RBM does not process inputs linearly instead nodes are reset changing their states based on a Boltzmann distribution until the entire network is in equilibrium where the probability distribution has converged across the network.

In the RBM equilibrium is achieved through forward and backward passes from and to the visible nodes, a randomly generated bias is added to the hidden nodes on the forward and backward pass but it is only added to the visible node on the backward pass.

For the forward pass, the following equation is used to determine the probability whether a node will activate:

p(𝑆i |𝑠j ) = σ(∑ sj × wij + bi )

Where Si is the following node and sj is the preceding node, and the sigmoid equation is:

𝜎(𝑥) = 1/ (1+e- 𝑥)

For the backward pass, a different equation is used to calculate the values for the same sigmoid equation:

p(𝑠j |𝑆i ) = σ(∑mi=1 sj × wij + bi )

Through these equations the values of the node’s inputs are predicted this is known as Gibbs Sampling. The differences between the input values and the new values are used to gain the training loss using contrasted difference calculated by the following equations:

wij new = wij old + (LR × Δw)

Where LR is the learning rate, Δw the change in weights as given by the difference of the probabilities of the node outputs:

Δw = (𝑠j ⊗ 𝑝(𝑆i|𝑠j) − 𝑠j′ ⊗ 𝑝(𝑆i ′|𝑠j ′)

Once the network has reached equilibrium the improved feature matrix is the output, the values of these improved features are summed together to give a sentence score. To generate the summary the first sentence of the document is taken then the remaining sentences are ranked by score, the upper 50% are taken for the summary and then sorted according to their position in the original document to give the first summary.

The fuzzy logic summary is generated by converting the previous feature scores into percentages these percentages are then sorted using triangular membership functions: HIGH, MEDIUM and LOW. Fuzzy logic IF-THEN rules are applied to this set to de fuzz the sentences into new categories of Important, Average and Unimportant the Important are then used to create the second summary by sorting them into their occurrence in the original document.

To generate the final summary sentences common to both are taken from both, the remaining uncommon sentences are then sorted by their score and the top 50% are taken. These sentences like before are then sorted into their original position in the document.

### Experiments

Two main experimental paths were taken a control and the proposed method, the control would the RBM generated summary to act as a baseline to the results gained from the proposed method if it improved upon the RBM summary. To evaluate the results the ROUGE metric was used, using the precision, recall and F measure to compare the two summaries.

### Evaluation

Ten documents were summarized, and the evaluation results were compared between the two from the precision, recall and F-measure the new method showed better results by a few decimal points.

### Conclusions

While the results showed that the proposed method was better there was no review of what the contents of the documents to be evaluated were and how the size of the document, its sentences might affect the summarization techniques or how the nine features contribute to the model. The number of documents used were also particularly small.

The RBM used as a control was too similar to the proposed method a more mainstream version of automated summarization should have been used as a control.

## Paper 2

Yousefi et al use an Auto Encoder (AE) to refine the features in the term frequencies of a document for summarization, using local and global vocabularies. The paper investigates the effect of adding noise to the term frequency before processing it with the encoder, creating a set of AEs’ called the Ensemble Noisy Auto Encoder (ENAE). This ensemble adds random noise to the input term frequencies this changes the network from a feed forward model to a stochastic run model.

Due to the reconstructive ability of AE, the inputs are corrupted with random noise this corruption is undone by the network and this leads to dependencies and importance of the different segments of the input being revealed. Core features of this implementation that differentiates it from standard De-noising AE implementations are:

1. A small amount of random noise is added to all inputs, Standard AE’s use a random zero mask.

2. The resulting output is the same as the input

3.Noise is added to the training and test data while in standard implementations test data is not manipulated.

### Models

The AE neural network is feed forward network, the main feature of this network is the bottleneck in it hidden layer, its input and output layers have the same number of nodes and the network replicates its input as its output. What makes this network interesting is that its hidden late has a bottleneck a layer where the number of neurones is much smaller than that of any other layer, (Neuron counts in the Layers drop closer to the bottle next and increase after it) this makes the network recreate the input from sparse features and through this finding the most important features. This mean that the output is a very close approximation that it is recognisable, but not exact.

Their model has two training stages Pre-training and fine tuning, in the pre-training stage a separate neural network is used to generate training weights for the AE network. The paper uses a Restricted Boltzmann Machine (RBM) to generate weights, a two-layer neural network the single hidden layer of the RBM helps reduce features. Several RBMs were run (Gaussian–Bernoulli, Bernoulli–Bernoulli, Noisy Rectified Linear Unit (NReLU)) as a stack where the outputs of one RMM is fed into the next, then back propagation was used to refine the weights.

Their implementation uses a custom representation for the words in the corpus to reduce the sparsity of representations like bag of words a local vocabulary is proposed only terms in a single document are considered with the vocabularies being the same size across the corpus this reduces sparsity in the vocabulary.

### Experiments

The concept of the model is that a single encoder is run for a single document and the ensemble results from each AE run is fused together to create a model for the corpus this fusion changes the model from a feed forward network to a stochastic model.

This ensemble network was run on two e-mail corpora that are specifically focused on summarization training, Summarization and Keyword Extraction from Emails (SKE) and the BC3 from British Columbia University. The queries used on the emails were either the subject of the email and for those that did not have a subject keyword that are seen through the dataset are used.

To get a base line tf-idf for the equivalent documents was carried out and different vocabulary sizes were used (1000, 60, 5%, 6%). The experimental AE had the structure of 140,40,30,10 in the hidden layer with 10 being the bottleneck and the reverse going to the decoder. The aim was that the first layer would being around double the size of the smallest vocabulary, the next layers were added as a deeper AE was seen to be more effective than shallow networks in previous research.

### Evaluation

Using the ROUGE and ROUGE2 metrics for evaluations they provide baselines for n-gram overlap for the auto summarization results of the different models based on the number of sentences included in a summary.

The Local vocabulary term frequency (Ltf) showed lower results than tf-idf of large vocabulary sets (above 5%) but the AE of Ltf had better results than both.

From the results it is seen that ENAE models have lower results than tf-idf when single sentences are used in the summary with an increase in sentences the results are much better with runs having uniform noise added scoring higher than gaussian noise.

When key value summarization is performed the results are similar ENAE uniform has the highest scores across all models, and the ENAE models being on average equal to or better than tf-idf the ENAE models show better results with larger summaries.

When compared to ROUGE results of abstractive text summarization models the Ltf and Ltf-AE models do not score as high as their tf-idf equivalents, this is a sharp contrast to the other experiments results.

The paper also evaluates the error values of the various models that were tested.

### Conclusions

The paper suggests a new method of auto summarizing documents using focused keywords and subjects to generate the summary. The evaluation of the model and number of experiments was remarkably in depth. While there is a depth of mathematical information on the RBM implementation there is less equations on the EA portion of the research.

Paper 3

Models

Models

Experiments

Experiments

Evaluation

Evaluation

Conclusions

conclusion

Paper 4

Models

Models

Experiments

Experiments

Evaluation

Evaluation

Conclusions

conclusion