Survey on Text Summarization

Amit Vhatkar, Pushpak Bhattacharyya, Kavi Arya
Indian Institute of Technology, Bombay
{asvhatkar, pb, kavi} @cse.iitb.ac.in

Abstract

Automatic text summarization is considered to be one of the hard problems because computationally there is no exact way of evaluating summary but the human can distinguish between good summary and bad summary. Also, summaries can be of various types like abstractive where new words and phrases are used, unlike extractive summarization where top scoring sentences from input text gets extracted as a summary sentence. Traditionally the focus of researcher was on building natural language generation which requires proper planning and realization of language. Various machine learning based approaches based on sequence labelling and SVR has been applied to extract summary sentences from the input text. Nowadays deep neural network models like sequence-to-sequence, LSTM, pointer-generator model are getting implemented to generate summaries. This report will give a brief idea about types of summary, summary evaluation measures and various ways to get summary.

Main terms - Automatic text summarization, machine learning, abstractive, extractive, deep neural networks

1 Introduction

Document summarization has been studied vastly in the NLP research community for more than 3 decades. As a number of document and available textual information increases day by day due to the advancement of the Internet, obtaining precise information becomes a challenging task. While acquiring information from a large set of documents user may choose to skip some documents or topics which may lead to loss of some of the important points. In order to avoid such loss, representing document(s) in condensed form without loss of content and without much/negligible repetition is an important task.

Type of Summary	Factors
Single and Multi-document	Number of Document
Extractive and Abstractive	Output(if exact or abstract is required)
Generic and Query-focused	Purpose (whether general or query related data is required)
Supervised and Unsupervised	Availability of training data
Mono, Multi and Cross-lingual	Language
Personalized	Information specific to user's need
Sentiment-based	Opinions are detected
Update	Current Updates regarding topic
E-mail bases	For summarizing e-mails
web-based	For summarizing web pages

Table 1: Various Types of Summarization Techniques(Gambhir and Gupta, 2017)

1.1 Types of Summarization

Broadly summarization approaches are categorized as **abstractive** and extractive. In an **extractive** type of summarization sentences from the input, texts are presented as it is as part of summary whereas in case abstractive summarization new sentences depicting gist of a topic are formed. Summarization approaches based on the number of documents are classified as a **single document** and **multi-document**. When only one document is used to generate a condensed form of text then it is termed as single document summary and when more that one documents are searched for desired information then it is termed as multi-document summarization.

Purpose of summary leads to **generic** and **query-focused** summarization. In a generic type of summarization entire document(s) is searched for various information contents, unlike query-focused where the document(s) are searched for only the topic mentioned in the query. The task of summarization can be applied to and sentiment from the

document and such type of summarization is called as **sentiment-base** summarization. In the **update** type of summary, it is assumed that the user is aware of basic information related to the topic and only need to know recent updates regarding the topic.

If generated summary language is same as input document(s) then it is called as **mono-lingual** summarization and when the language of summary varies with that of input document(s) summary then it is called as **multi-lingual** summarization. Sometimes based on pro file of user nature of summarization gets varied such type of summarization is termed as **personalized** summarization. Apart from these, there are **web-based**, **e-mail based** type of summarization as shown in the table 1.

1.2 Summary Evaluation Techniques

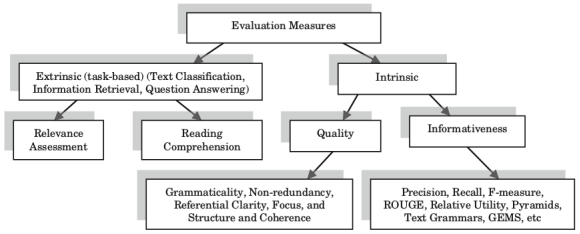

Figure 1: Summary Evaluation Techniques(Gambhir and Gupta, 2017)

Automatic generation of the summary is a hard task since we don't know what part of the information should contribute to the summary. The varying perspective of summary makes it harder to evaluate automatically generated summary even from the trained human. Someone may see a certain point important while others may think that point less important. Purpose of summary can help to evaluate automatically generated summary. As described in the survey paper (Gambhir and Gupta, 2017) evaluation of summary can be broadly categorized as follows,

1.2.1 Extrinsic Evaluation

There are various tasks that help to generate a summary of the text. In the extrinsic type of evaluation approach of summary gets tested for its usefulness to these various supporting tasks. Sometimes this type of evaluation is gets termed as **task-based** evaluation. Extrinsic evaluation is further categorised as follows

1. Relevance assessment: Generated summary is tested against relevance to the topic. This method is mostly used to topic/query-focused types of summarization.

2. Reading comprehension: It tests weather generated summary can be used to answer multiple choice tests.

1.2.2 Intrinsic Evaluation

Generally, reference summaries are used to evaluate generated summary mostly on the basis of informativeness and coverage. The relevance of summary to the input document(s) has an issue of finding a relevant topic from the document(s) as relevance has not a rigid definition. As shown in figure1 intrinsic evaluation is categorised as follows:

1. Quality evaluation: Quality of text in summary is checked on the basis of linguistic parameters like grammatically, structures and coherence, vocabulary, non-redundancy etc.

2. Informativeness evaluation: This is the most used type of summary evaluation techniques. There are two ways in which informativeness of summary is evaluated, they are as follows,

   Automatic: don't need human annotation

   Semi-automatic: needs human annotation

Following session explains some of the informativeness intrinsic evaluation techniques.

- ROUGE

  ROUGE (Recall-Oriented Understudy for Gisting Evaluation) makes use of reference summary for evaluation. It

looks for co-occurrences of various levels grams in the generated summary and reference summary. Five different metrics are available to capture ROUGE.

ROUGE-N: checks for overlap of N gram

ROUGE-L: checks for longest common sub-sequences(LCS)

ROUGE-W: weighted LCS, favours longest LCS

ROUGE-S: skip-bigram based co-occurrence check

ROUGE-SU: checks of co-occurrence except bi-gram and uni-gram.

- BLEU (Bilingual Evaluation Understudy)

  It is a modified form of precision. The modification comes from overlap between candidate summary and reference summary. Here overlap of words in summary is calculated with respect to the maximum count of that word from all reference summaries. It can be written in the equation as follows,

$$P = m_{max}/w_t \tag{1}$$

  where m_{max} is maximum time occurrence of word from all reference summaries and w_t is total number of words present in generated summary.

- Basic Element(BE)

  Sentences are expressed in the form of using three word namely head, modifier/argument and relation(between head and modifier). Then these are mapped against various equivalence expressions.

- DEPEVAL

  This evaluation method is similar to BE method wherein parsers are used in this method unlike minipar in BE. Dependency triplets (head —modifier— relation) are from the automatically generated text are checked against the ones from reference summaries.

- Pyramid Method

  It is semi-automatic intrinsic informativeness evaluation method which makes use of nation of Summary Content Unit(SCU) which is nothing but the set of sentences with the similar quotient of informativeness. SCUs generated as part of summary and one which are similar to various human level SCUs gets higher weight.

1.3 Outline

Rest of the document is organised as per chronological approaches applied and suggested by the community to provide a solution for Text Summarizing. section 2 suggests machine learning based approaches which are further categorised into sequence labelling task and statistical approaches. section 3 briefs about recent summarization approaches from sequence-to-sequence(Nallapati et al., 2016) RNN to attention models(Rush et al., 2015). This section is followed by conclusion and future work.

2 Machine Learning Based Summarization Approaches

Machine learning is broadly used to perform two types of tasks namely classification and regression. In the case of classification task, class of given input is decided based on the similarity of its features against features represented by classes. Nature of regressing task is to predict certain value which is a function of features of the given input. Molding machine learning approaches to generate text summary is an interesting area of research, where summarization is posed as either labelling task which can be sequence labelling(Shen et al., 2007) where labels of other sentences are also considered or general labelling task where approaches like SVM-based ensemble(Chali et al., 2009) or SVR based ranking algorithm(Li et al.,

2007) are used to decide rank of the sentence depending on features of sentence. Also various statistical approaches like graph-based ranking approach(Mihalcea, 2004), manifold-ranking based approach(Wan et al., 2007) and discourse structure based approach (Marcu, 1997) etc. can be applied to get summary sentences from the document(s).

2.1 Summarization as Labelling Task

2.1.1 Labelling Using SVM and SVR

- SVM Based Ensemble Approach to Multi-Document Summarization(Chali et al., 2009) This is topic-focused extractive multi-document text summarization approach. Proposed approach is targeted for Document Understanding Conference(DUC) 2007.

  Problem Definition(Chali et al., 2009): *Given a complex question and collection of relevant documents, the task is to synthesize a fluent, well-organized 250-word summary of the document that answers the question(s) in the topic.*

  Features: Query related and other important features like N-gram overlap, LCS, WLCS, skip-bigram, gloss overlap, BE overlap, length of sentence, position of the sentence, NE, cue word match, title match etc are extracted for each sentence.

  SVM Ensemble: By using cross-validation with 25% data out and 75% data for training, 4 different SVM model are trained using above mentioned features. An ensemble of these 4 classifiers are used for deciding the rank of the sentence and top N sentences are labelled as a summary sentence and others are labelled as non-summary sentence.

  Result: Author has compared their approach with a baseline which selects lead sentences and Single SVM approach on various level of ROUGE measure.

Systems	R-1	R-L	R-W	R-SU
Baseline	0.3347	0.3107	0.1138	0.1127
Single	0.3708	0.3035	0.1113	0.1329
Ensemble	0.3883	0.3197	0.1177	0.1463

Table 2: Result of SVM-Based Ensemble Approach to Multi-Document Summarization (Chali et al., 2009)

- Multi-document Summarization Using Support Vector Regression(Li et al., 2007)

  It is multi-document extractive test summarization approach which makes use of documents made available by DUC-2006 for training purpose. DUC-2006 Data set contains 50 topics each having 25 news documents and 4 reference summaries for each topic. Proposed system (Li et al., 2007) has three steps: Text preprocessing, Sentence scoring and Post-processing. Preprocessing of text carries segmentation of sentences and removing of news heads from the document (DUC- 2006's data comprise of news articles). Sentence scoring makes use of various features of sentences like word-based features, phrase-based NE features, semantic-based WordNet feature, centroid feature, NE number feature, sentence position etc. SVR uses combination these features to generate sentence ranking.

  Hypothesis(Li et al., 2007): More similar a sentence to four summaries, larger its score must be. Authors had come up with two strategies to score sentence based on similarity of sentence with reference summaries. Let, s be the sentence under consideration, sim be the similarity function and S_i be the i^{th} summary document.

  Average: Here final score of the sentence is average of its score with reference summaries.

  $$Score(s) = \sum_i sim(s, S_i)$$

  Maximum: Here final score of the sentence is the maximum score among all reference summaries.

$$Score(s) = \max_i sim(s, S_i)$$

After obtaining scores for each sentence all sentences are represented as a feature vector along with their similarity score i.e. D=$\{V_s$, score(s)$\}$. Finally regressing function is learned by SVR model. Accuracy of this method is mentioned in table 3 where baseline systems train SVR with the same set of features but with a manual assignment of weights and Best submitted systems is the one which performed best in DUC-2006.

System	Rouge-2
Best submitted system	0.09558
SVR-based system	0.09057
Baseline system system	0.08012

Table 3: Performance of SVR-based Summarization Technique (Li et al., 2007)

2.2 Other Statistical Based Approaches

There are methods which makes use of statistics for ranking sentences. Some of them use graph-based(Mihalcea, 2004) approach while others use manifold-ranking(Wan et al., 2007). Also historically some of the approaches have considered use of discourse structure(Marcu, 1997) for summary generation. This subsection explains some of these statistical based summary generation algorithms in details.

2.2.1 Graph-based Ranking Algorithm for Sentence Extraction

Traditionally graph-based approaches are used for analysing link structures of Word Wide Web also for analysing citation and social networks etc. But the approach suggested in paper (Mihalcea, 2004) goes one step ahead and makes use of graph structure for extracting important sentences from documents. This approach can be categorised as unsupervised extractive multi-document summarization. Authors have named it as **TextRank**.

To have common notation throughout discussion about this approach, let's consider G = (V, E) where V is set of vertices and E is set of edges connecting those vertices. Let $In(V_i)$ be the set of nodes pointing to the node V_i and $Out(V_i)$ be the set of nodes to whom node v_i points. In the case of un-directed graph $In(V_i)$ will be same as $Out(V_i)$.

- Introduction
  Summarization approach described in the paper (Mihalcea, 2004) make use of modified versions of following tradition graph-based approaches,

  HITS (Hyperlinked Induced Topic Search)
  This algorithm is proposed for ranking web pages. It generates two values for each node in the graph namely *Authority Score*($HITS_A(V_i)$) and *Hub Score*$HITS_H(V_i)$. Authority value represents number of incoming links whereas hub value denotes number of outgoing links. Formulation of these values are as given below,

$$HITS_A(V_i) = \sum_{V_j \epsilon In(V_i)} HITS_H(V_j) \tag{2}$$

$$HITS_H(V_i) = \sum_{V_j \epsilon Out(V_i)} HITS_A(V_j) \tag{3}$$

  Positional Power Function
  Positional power function (POS_P) takes number of successors of node and their scores into consideration for calculating score for a specific node. On the other hand, positional weakness function (POS_W) determines score of node based on its ancestors and their scores.

$$POS_P(V_i) = \frac{1}{|V|} \sum_{V_j \epsilon Out(V_i)} (1 + POS_P(V_j)) \tag{4}$$

$$POS_W(V_i) = \frac{1}{|V|} \sum_{V_j \epsilon In(V_i)} (1 + POW_W(V_j)) \tag{5}$$

PageRank

This graph based algorithm was designed to analyse web links. It gives single score value to the node after considering out-going and in-coming links in single equation as shown below,

$$PR(V_i) = (1-d)+d* \sum_{V_j \epsilon In(V_i)} \frac{PR(V_j)}{|Out(V_j)|} \quad (6)$$

where d ϵ [0,1] and usually set to be 0.85.

These algorithms start executing in iterative manned from random node with random initialization of weights till convergence. Some threshold values have to be considered for convergence. It is assumed that if difference between values of current iteration and previous iteration is less than the threshold then algorithm has converged. Also, it has been observed that starting node will not affect values obtained after convergence but it affects number of iterations.

- Weighted Graphs

  In TextRank approach nodes are considered as sentences there can be multiple partial links between nodes. To capture the importance of these partial links, each link gets weight assigned to it. This causes above mentioned formulas to change so as to adapt the notion of weights of the link. Final values of score after consideration of weight changes as compared to unweighted formulation but the shape of the convergence curve and number of required iteration remains almost same.

- Sentence Extraction

  The first step of TextRank is building a graph with a node representing sentence and link between nodes depicting similarity i.e. content overlap between nodes. Author of this approach has hypothesized as follows *sentences that address certain concept in text gives the reader 'recommendation' to refer to other sentences that address the same concept*. Given two sentences S_i and S_j and sentences are formed from set of N_i words. The term in the denominator is used for normalization purpose which avoids giving more weights to long sentences.

$$Similarity(S_i, S_j) = \frac{|W_k|W_k \epsilon S_i \& W_k \epsilon S_j|}{log(|S_i|+log|S_j|)} \quad (7)$$

  In the second step, graph can be represented as follows,

  Un-directed

  Directed forward: orientation of edges follows pattern from text.

  Directed backward: orientation of edges is exactly opposite as that of flow of sentence text.

  In the last, after running ranking algorithm, sentences get sorted on the score and top N scoring sentences are considered as summary sentences.

Algorithm	Graph		
	Un-directed Graph	Dir. forward	Dir. backward
$HITS_A^W$	0.4912	0.4584	0.5023
$HITS_H^W$	0.4912	0.5023	0.4584
PSP_P^W	0.4878	0.4538	0.3910
PSP_W^W	0.4878	0.3910	0.4538
PageRank	0.4904	0.4202	0.5008

Table 4: Result of Graph-Based Ranking Algorithm for Text Summarization Approach(Mihalcea, 2004)

- Evaluation and Discussion

  Authors have measured the performance of TextRank with on DUC-2002 Dataset with ROUGE as evaluation measure. The results are as shown in table 4. If directed graphs with HITS for sentence ranking is used TextRank gives best results.

  Author mentions that as score for each sentence is calculated this method can be extended to generate long summaries and this method makes use of information drawn from text only making it fall into the category of unsupervised algorithms.

2.2.2 Manifold-Ranking Based Topic-Focused Multi-Document Summarization

This is topic-based multi-document extractive text summarization which makes use of dependence between sentences of documents and between sentences of topic. Summary generated by this approach tends to have high bias towards topic and less redundancy. Paper (Wan et al., 2007) states that topic-focused summaries should keep the information mentioned in documents, tries to make the information as novel as possible and importantly it should be biased to the topic. Rest of this session describes how the proposed approach tries to achieve these properties.

- Overview

  The manifold-ranking based approach mainly comprised of two steps, in first step, manifold-ranking score for each sentence get computed and in second step diversity penalty gets imposed on the score calculated in the previous step. In order to score the sentences paper (Wan et al., 2007) suggest to assign weights to inter-document and intra-document connections between sentences.

  As described above, to maintain properties of good summary proposed approach makes use of **biased information richness** and **information novelty**. These terminologies are defined in details in paper (Wan et al., 2007). In final step after imposing penalty for obtaining diversity sentences with top scores are considered as summary sentences.

- Manifold-Ranking Process

  Manifold-ranking process assumes that nearby points are likely to have the same ranking and points on the same structure/cluster/manifold are likely to have same score. Ranking process can be described as

  Form a network of data point in current case sentences and topic description. Initially, positive rank gets assigned to known points and all other points get ranked as 0.

  Points the get spread based on their ranking score to their nearby nodes

  Previous step is repeated till convergence.

  Let χ be the set of all sentences $(x1, ..., x_n)$ in documents including topic description(x_0) i.e. $\chi = \{x_0, x_1, ..., x_n\} \subset R^m$. Let, $f : \chi \rightarrow R$ be to ranking function which assign rank f_i to each x_i. Let $y = [y_0, ..., y_n]^T$ and $y_0 = 1$ since x_0 is topic sentence.

  Normalization in third step of algorithm is required to guaranty convergence of algorithm. Fourth step is required to spared the ranking to neighbouring nodes. Parameter α in step four describes relative contribution to ranking score from neighbours and initial score. The affinity matrix W is used to capture the notion of importance of link between sentences the document.

$$W = \lambda_1 W_{intra} + \lambda_2 W_{inter} \quad (8)$$

  Where, $\lambda_1, \lambda_2 \epsilon [0, 1]$, if $\lambda_1 = \lambda_2 = 1$ then both inter and intra document gets equal importance. Paper (Wan et al., 2007) also suggest greedy approach to impose diversity penalty. As shown in equation 9 rank of sentence is decreased by the factor of ω(penalty degree factor) after considering similarity with other sentences. If $\omega = 0$ then no diversity penalty get imposed on the rank of the sentence.

$$RankScore(x_j) = RankScore(x_j) - \omega \cdot S_{ji} \cdot f_i^* \quad (9)$$

- Evaluation

  Author mentions use of DUC-2003, 2005 data set with ROUGE as an evaluation metric. Parameter setting for the algorithm was, $\omega = 8$, $\lambda_1 = 0.3$, $\lambda_2 = 1$ and $\alpha = 0.6$. Lead baseline takes first sentence one-by-one in the last document which are

Algorithm 1 Manifold-Ranking Algorithm (Wan et al., 2007)

1: Compute the pair-wise similarity values between sentences (points) using the standard Cosine measure. The weight associated with term t is calculated with the $tf_t * isf_t$ formula, where tf_t is the frequency of term t in the sentence and is_f t is the inverse sentence frequency of term t, i.e. $1 + log(N/n_t)$, where N is the total number of sentences and n t is the number of the sentences containing term t. Given two sentences (data points) x_i and x_j , the Cosine similarity is denoted as $sim(x_i, x_j)$, computed as the normalized inner product of the corresponding term vectors.

2: Connect any two points with an edge if their similarity value exceeds 0. We define the affinity matrix W by $W_{ij} = sim(x_i, x_j)$ if there is an edge linking x_i and x_j. Note that we let $W_i i = 0$ to avoid loops in the graph built in next step.

3: Symmetrically normalize W by $S = D^{-1/2} W D^{-1/2}$ in which D is the diagonal matrix with (i,i)-element equal to the sum of the i-th row of W.

4: Iterate $f(t + 1) = \alpha S f(t) + (1 - \alpha) y$. until convergence, where α is a parameter in (0,1).

5: Let f_i^* denote the limit of the sequence $\{f_i(t)\}$. Each sentences $x_i (1 \leq i \leq n)$ gets its ranking score f_i^*.

Systems	R-1	R-2	R-W
Manifold-Ranking	0.37332	0.07677	0.11869
Similarity-Ranking1	0.36088	0.07229	0.11540
S16	0.35001	0.07305	0.10969
Similarity-Ranking2	0.34542	0.07283	0.1115
S13	0.31986	0.05831	0.10016
S17	0.31809	0.04981	0.09887
Coverage Baseline	0.30290	0.05968	0.09678
Lead Baseline	0.28200	0.0468	0.09077

Table 5: Result of SVM-Based Ensemble Approach to Multi-Document Summarization (Wan et al., 2007)

chronologically ordered whereas coverage baseline takes first sentence one-by-one from first document to last documents. Similarity-Ranking1 don't apply weight to inter and intra-document sentences whereas Similarity-Ranking2 doesn't impose diversity penalty. S13, S16, S17 are top performing systems in DUC-2003. Table 5 shows the comparison of Manifold-Ranking based Summarization with above-mentioned summarization approaches.

- Parameter Tuning

  Graph in figure 2 shows the result of various systems captured on the values of the penalty degree factor. Using this information author states that no diversity penalty and too much of diversity penalty degrades the performance of the summarization systems. Graph in figure 3 is plotted for ratio of weights assigned to the links between inter-document sentences and intra-document sentence against ROUGE-1 measure. As a score of inter-document link is always greater than intra-document link author in paper (Wan et al., 2007) suggest to give more importance to the links between sentences of the same documents.

3 Recent Summarization Approaches

So far, we have discussed traditional and machine learning based ways of summary generation. Machine learning based approaches are mostly of an extractive type of summa-

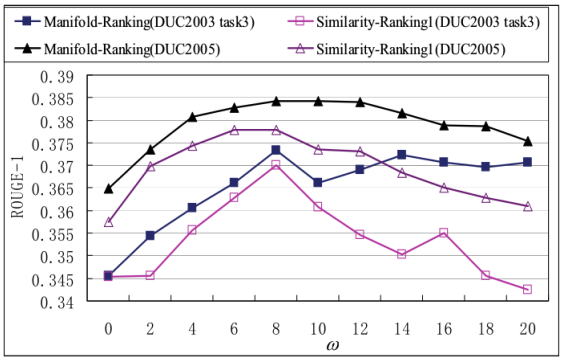

Figure 2: Performance of Manifold-Ranking Based Summarization Approach Based on Penalty Degree Factor (Wan et al., 2007)

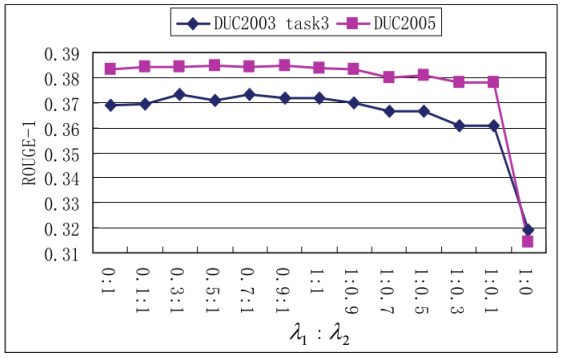

Figure 3: Performance of Manifold-Ranking Based Summarization Approach Based on Importance Given to Inter Document Link and Intra-Document Link (Wan et al., 2007)

rization. Recently due to the success of neural networks, various newer approaches using deep neural networks are getting proposed. Along with summarization, Machine Translation(MT) is also getting discussed vastly. In machine translation, input text presented in one language is converted to another language. Like summarization, machine translation also makes use of language interpretation and generation modules.

On top of the basic idea presented in the approach of Neural Machine Translation(NMT), various other approaches were proposed, summarization using sequence-to-sequence RNN(section 3.1) is one of them. Recurring Neural Network(RNN) is an advanced form of feedforward neural network, where the neuron is used to train recursively in a single pass of training and such multiple passes can be used to train the model. Enhancements are made to the Sequence-to-Sequence(S2S) models by adding the notion of attention. Attention allows the model to search for a specific location to learn from. Neural Attention Model for Sentence Summarization as explained in section 4 is one of such models. Finally, we describe pointer-generator network model in section 5 for text summarization, which enhances attention model by probabilistically choosing between generation and extraction.

3.1 Sequence-to-Sequence RNNs for Text Summarization(Nallapati et al., 2016)

In summarization length of output i.e. summary in not much related to the length of the input text unlike machine translation where length of out put i.e. translation is a function of the length of the input text. In terms of the loss of information, MT has to be loss-less while summary can skip over unimportant topics. The paper (Nallapati et al., 2016) propose an approach to perform abstractive summarization based on basic encoder-decoder RNN. The authors also have introduced new dataset for abstractive summarization but we are not considering that part in our discussion. Rest of this session describes the model.

3.1.1 System Overview

Capturing keywords and handling out of vocabulary (OOV) words is a challenging task for deep neural network based approaches. To identify the topic discussed in the text *feature-rich* encoder is proposed in the paper (Nallapati et al., 2016).

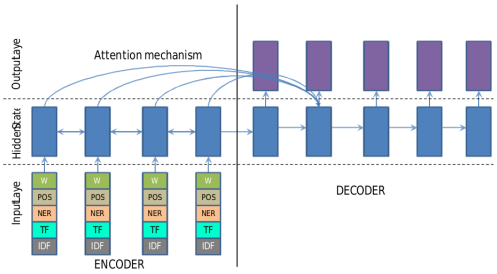

Figure 4: Feature-Rich Encoder Proposed in the Paper (Nallapati et al., 2016)

As shown in figure 4 apart from the em-

bedding of the word, named-entity tag(NER), part-of-speech(POS) tag, term frequency(TF), inverse document frequency(IDF) are considered as additional features of the input word. For the case of continuous features like TF and IDF fixed number of bins are used to convert them to categorical feature and then one-hot encoding is applied. These newly added features are concatenated to the original word embedding.

3.1.2 Switching Generator/Pointer Model

The vocabulary of decoder gets fixed at the time of training so it cant emit unseen or OOV words. Previously these OOVs are handled by replacing it with UNK(unknown). This approach makes use of switching generator/-pointer model for handling OOVs.

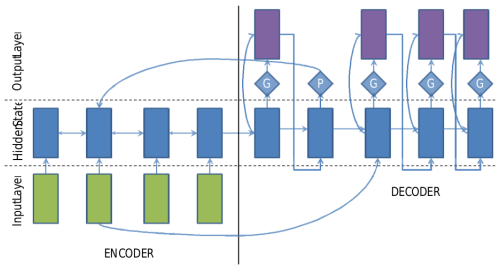

Figure 5: Switching Generator-Pointer Model Proposed in the Paper (Nallapati et al., 2016)

The authors have termed their approach for handling OOV as the *switch*. This switch is located at the decoder and based on the value produced by switch decoder decided to generate a new word or simply copying the word from input text. Whenever switch is on decoder generates word from embedding and whenever switch is off decoder generates a pointer to the word in input text. Switch produces probabilistic value based as shown in figure 10.

$$P(s_i = 1) = \sigma(v^s \cdot (W_h^s)h_i + W_e^s E[o_{i-1}] + W_c^s c_i + b^s) \quad (10)$$

Above equation, give the probability of switching begin on in i-th time-step of decoder. h_i is hidden state, $E[o_{i-1}]$ embedding vector of previous emission, c_i is a context vector which handles attention over input sequence. Here W's are learn-able parameters. As stated, in case of pointer generation, the proposed model probabilistically generates appropriate pointer to the location in the input document. The equation 11 and 12 represents the way to find appropriate position in input text to copy word from. In equation p_i denotes pointing in input text which is used to produce i-th word in summary and P_i^a is attention distribution for every word in the input document. If two locations get same probability value then hard assignment is done by giving preference to the first occurring word.

$$P_i^a \propto exp(v^a \cdot (W_h^a h_{i-1} + W_e^a E[o_{i-1}] + W_c^a h_j^d + b^a)) \quad (11)$$

$$p_i = \underset{j}{argmax}(P_i^a(j)) \ for j \, \epsilon \{1, \ldots, N_d\} \quad (12)$$

As shown in figure 5, if pointer is used for producing word as the summary word, decoder uses its embedding as input for next time-step, otherwise embedding of previous time-step is used.

3.1.3 Capturing Hierarchical Document Structure

Till now in proposed approach, each word is checked for producing the summary moreover, model suggests using sentence level information along with word level information for producing summary. Sentence level attention is mixed with word level attention and then updated word level attention is used by decoder to produce the summary words.

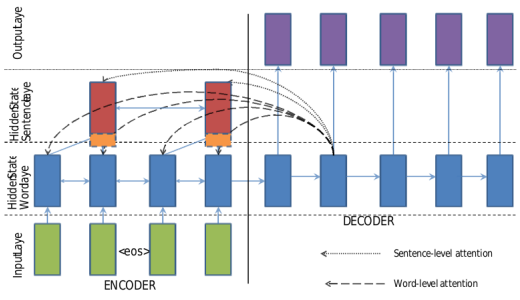

Figure 6: Hierarchical Encoder for Capturing Sentence Level Information (Nallapati et al., 2016)

Figure 6 shows incorporation of sentence level attention for summarization. Updated word level attentions are used as input to the decoder. Sentence level attentions are concatenated by the positional embedding of the sentence.

$$P^a(j) = \frac{P^a_w(j)P^a_s(s(j))}{\sum_{k=1}^{N_d} P^a_w(k)P^a_s(s(j))} \quad (13)$$

The equation 13 show update performed on the word level attention, where $P^a_w(j)$ original word attention at j-th position in input text, $s(j)$ denotes ID of sentence at j-th position, $P^a_s(l)$ is weight given to the l-th sentences attention, N_d is total number of words. Positional embedding is concatenated to the embedding of sentence so as to give importance to sentences present at a specific location in the input text.

3.1.4 Result and Analysis

Authors have compared their approach with the model presented in Rush et al(2015)(Rush et al., 2015). Original model proposed by Rush et al(2015) (Rush et al., 2015) is termed as ABS and its variant which combines additional log-linear model with manually picked features is termed as ABS+.

Model	Rouge-1	Rouge-2	Rouge-L
TOPIARY	25.12	6.46	20.12
ABS	26.55	7.06	22.05
ABS+	28.18	8.49	23.81
Sequence-to-Sequence RNN	28.35	9.46	24.59

Table 6: Performance of Sequence-to-Sequence Abstractive Summarization Model(Nallapati et al., 2016)

All these models are trained on *Gigaward* corpus and the proposed model is trained only on the first sentence of *Gigaword* corpus. For testing DUC-2003 corpus was selected and ROUGE as a measure of performance. DUC-2003 top performing model TOPIARY is also considered for comparison. Table 6 shows results of the evaluation.

4 Neural Attention Model for Sentence Summarization(Rush et al., 2015)

It is also called as Attention Based Summarization(ABS). It tries to make use of the linguistic structure for generation of summaries, for that it captures the attention in input sequence to produce correct output. This is successor of this model is presented in section 3.1. The approach presented in the paper (Rush et al., 2015) is of abstractive sentence summarization which takes sentence as input and converts it into a condensed form. This approach can be further extended to produce summary of documents.

4.1 Proposed Model

Let's consider x as input sequence of words presented in the sentence, y_i be to i-th generated word and y_c denotes context of output word y. Then conditional log probability of the summary sentence can be written as in equation 14.

$$log\, p(y|x;\theta) \propto \sum_{i=0}^{N-1} log\, p(y_{(i-1)}|x, y_c;\theta) \quad (14)$$

where N is length of output sentence which is fixed as proposed in the model and θ represents other learn-able parameters. This is same as *Markov model* with c begin variable defining degree of Markov assumption.
The objective of the model can be formalized as shown in below equation,

$$\underset{y}{argmax}\, log(p(y|x)) \quad (15)$$

4.2 Natural Language Model

The approach proposed in the paper (Rush et al., 2015) makes use of basic feedforward neural networks and generates probability distribution over output sequence. Probability of generating next word is given in equation 16 where, $\theta = (E, U, V, W)$ and C is window size of context, E is embedding martix and V and W are weights associated with hidden

state and encoder.

$$p(y_{i-1}|y_c, x; \theta) \propto exp(Vh + Wenc(x, y_c)) \tag{16}$$

$$y_c = [Ey_{i-C+1} \dots Ey_i] \tag{17}$$

$$h = tanh(Uy) \tag{18}$$

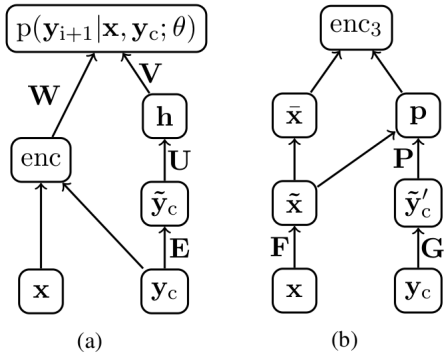

Figure 7: System Architecture of Attention Based Sentence Summarization, (a) Decoder with additional encoder element (b) Attention based encoder (Rush et al., 2015)

Figure7 represents the same formulation in pictorial view. In figure, part (a) explains the calculation required for getting probability of specific word as summary word and part (b) show formulation of attention based encoder (enc_3) which is discussed in 4.3.

4.3 Encoder and Decoder

Encoder takes input words x and already generated words y_c as input and transforms this using feature matrix F. Three encoders are suggested in the paper. they are as follows,

- Bag-of-Words Encoder (enc_1):
  It assigns equal probability to each word presented in the input sentence and the multiplies it with the transformed x. This approach is same as uniform distribution over input words. This model looses the connection between words and it also sufferers from stop words.

$$enc_1(x, y_c) = p^T x^{'}$$
$$p = [1/M, \dots, 1/M]$$
$$x^{'} = [Fx_1, \dots, Fx_M]$$

- Convolutional Encoder (enc_2):
  This type of encoders makes use of standard time-delay neural networks between convolution layers and max pulling. In order to avoid issues faced by bag-of-words it considers interaction between words in the input sentence.

- Attention-Based Encoder (enc_3):
  Convolutional encoders are hard to learn therefor authors has suggested simpler attention-based encoder with following formulation,

$$enc_3(x, y_c) = p^T x^{'} \tag{19}$$
$$p \propto exp(x^{'} P y_c^{'}) \tag{20}$$
$$x^{'} = [Fx_1, \dots, Fx_M] \tag{21}$$
$$y_c^{'} = [Gy_{i-C+1}, \dots Gy_i] \tag{22}$$
$$\forall\, x_i^{'} = \sum_{q=i-Q}^{i+Q} x_i^{'}/Q \tag{23}$$

Where G is embedding for context and Q is smoothing window, which allows to focus on certain part of the sentences if current context aligns with current input word x. This approach is very similar to the bag-of-words approach with replacement of uniform distribution to probablistic distribution over alignment of input words.

$$NLL(\theta) = -\sum_{j=1}^{J}\sum_{i=1}^{N-1} log\, p(y_{i+1}^{(j)}|x^{(j)}, y_c; \theta) \tag{24}$$

Let's consider training set with J pairs of sentence $((x^1, y^1), \ldots, (x^J, y^J))$ and associated summary sentence. To train the model negative log likelihood objective function is used and to generate the summary sentence. In decoder beam search algorithm is proposed, as complicity of greedy algorithm is exponential in terms of the window. The equation24 describes objective function for training and learning various parameters,

4.4 Result and Analysis

The authors compare their results based on ROUGE measure and DUC-2004 as evaluation data set. They compare their approach with TOPIARY which is top performing system in DUC-2004 and MOSES+ which is phrase based statistical MT system. Table 7 shows actual results of the evaluation.

Model	Rouge-1	Rouge-2	Rouge-L
TOPIARY	25.12	6.46	20.12
MOSES	26.5	8.13	22.85
ABS	26.55	7.06	22.05
ABS+	28.18	8.49	23.81

Table 7: Comparison of Results of ABS with Various Approaches Summarization Systems Summarization on DUC-2004 Dataset (Rush et al., 2015)

5 Summarization with Pointer-Generator Network(See et al., 2017)

Recent summarization approaches discussed so far tries to generate the summaries irrespective of correctness of factual data and without considering novelty of information in produced summary. Abstractive summarization proposed in the paper (See et al., 2017) tries to overcome these shortcomings along with handling of OOVs. The author discusses three approaches (1)Baseline model (section 5.1) (2)Pointer generator model (section 5.2) and (3)Coverage mechanism (section 5.3). Rest of this session discusses these approaches in detail.

5.1 Sequence-to-Sequence Attention Model

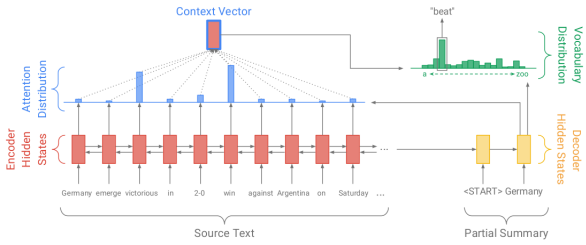

Figure 8: Baseline Sequence-to-Sequence Attention Model for Abstractive Text Summarization (See et al., 2017)

The paper (See et al., 2017) discussed baseline model which is similar to the one we have discussed in section 3.1. Proposed baseline uses single bidirectional LSTM as encoder and single layer unidirectional LSTM as decoder. This baseline model is depicted in figure 8 from which it gets clear that the word *beat* gets generate based on present context of sentence. Let encoder hidden states be h_i and decoder hidden states be s_i then attention distribution at time-step t can be formulated as shown in equation 25 and 26 where v, W_h, W_s and b_{atten} are learnable parameters.

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + b_{attn}) \quad (25)$$

$$a^t = softmax(e_t) \quad (26)$$

Attention can be considered as the location to produce next word from. Attention is used to get weighted sum of hidden state which represents overall hidden state h^*. This hidden state along with hidden state of decoder then used to probability distribution P_{vocab} over all words in vocabulary. The equation 27 captures the calculation required to generate probability distribution over all vocabulary words.

$$P_{vocab} = softmax(V^{'}(V[s_t, h_t^*] + b) + b^{'}) \quad (27)$$

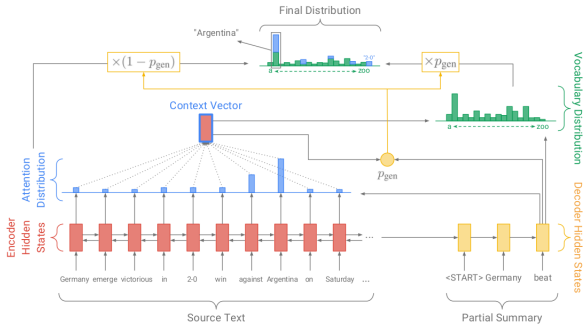

Figure 9: Pointer-Generator Model for Abstractive Text Summarization (See et al., 2017)

where, V, V', b, b' are learnable parameters. Negative log-likelihood is used to train and learn the parameters.

5.2 Pointer-Generator Network

This is hybrid model which combines the baseline model and the model of pointer network proposed in Vinyals et al., 2015. Pointer generation model tries to handle OOVs either by copying from input text or by generating from decoder vocabulary. Figure 9 describes use of working of pointer-generator model. Generation probability is calculated as shown in equation 28 where w's and b_{ptr} are learnable parameters.

$$p_{gen} = \sigma(w_h^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \quad (28)$$

Unlike the approach proposed in section 3.1, where pointer-generation a switch is binary variable, in proposed model switch is modelled as continuous variable between range [0, 1]. The authors call this is a *soft switch* and σ function is used to decide between generation and pointer mechanism. The notion of extended vocabulary which is a combination of vocabulary and all words appearing in the input text. The equation 29 give probability distribution of vocabulary words.

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \quad (29)$$

When w happens to be OOV, P_{vocab} becomes zero and in case of non-appearance in source document, attention term becomes zero. negative log-likelihood is used as loss function to train the model and learn the parametes.

5.3 Coverage Mechanism

The main purpose of coverage mechanism is to avoid repetition in the generated summary. To achieve this, paper (See et al., 2017) suggest maintaining coverage vector c^t which is attention distribution over all previous decoder time-steps. The equation 30

$$c^t = \sum_{t'=0}^{t-1} a_t \quad (30)$$

Updated equation form of the equation 31 after considering coverage vector is as shown bellow,

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + w_c c_i^t + b_{attn}) \quad (31)$$

The author also suggests to add coverage loss to negative log likelihood, then equation 32 describe overall loss for learning parameters, where λ also gets learnt.

$$loss_t = -log\, P(w_t^*) + \lambda \sum_i min(a_i^t, c_i^t) \quad (32)$$

min of attention and coverage is useful for penalizing only overlapping part.

5.4 Result and Analysis

The authors compare their model with abstractive model presented in section 3.1 and then the combination of sequence-to-sequence(s2s) with baseline by training on 150k vocabulary words and 50k vocabulary words. Table 8 shows results of the evaluation on the basis of ROUGE measure on CNN/Daily Mail training dataset where the proposed model makes use of 256-dimensional hidden states and 128-dimensional word embedding.

The Author has concluded that abstractive summarization is hard to achieve using pointer generator model since the probability

Model	Rouge-1	Rouge-2	Rouge-L
Abstractive Model (Nallapati et al., 2016)	35.46	13.3	32.65
s2s 150k vocab	30.49	11.17	28.08
s2s 50k vocab	31.33	11.81	28.83
Pointer Generator	36.44	15.66	33.42
Pointer Generator + Coverage	39.53	17.28	36.38

Table 8: Comparison of Results of Models Suggested in Paper(See et al., 2017) with Basic Sequence-to-Sequence Model Proposed in Paper (Nallapati et al., 2016)

of generation from 0.3 to 0.53 during training but while testing it gets stuck at 0.17.

6 Conclusion

In this survey we have categorized ways of summarization as traditional approaches, machine learning based approaches and recent approaches which uses notion of deep neural network for generating summary. We have also described various of types of summarization like abstractive-extractive, multi-lingual-monolingual, supervised-unsupervised etc. Some of summary evaluation measures like ROUGE, BLEU, DEPVAL etc. are also described.

Recently, due to advances in computational power, sophisticated models based on neural networks, joint learning, reinforcement learning etc. are getting proposed and year by year more accurate and acceptable summaries are getting produced. Also various evaluation measures like ROUGE, BLEU, METEOR etc. are used to decide quality of generated text.

After exploring this much, we can conclude that Text Summarization is vastly studied topic in the field of AI-NLP and research is still going on to achieve human-level excellence for producing summaries. As there is not exact measure to declare a summary as good or bad and as the readers perception changes as per domain knowledge, topic of text summarization remains open for researchers.

References

Yllias Chali, Sadid A Hasan, and Shafiq R Joty. 2009. A svm-based ensemble approach to multi-document summarization. In *Canadian Conference on Artificial Intelligence*, pages 199–202. Springer.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.

Sujian Li, You Ouyang, Wei Wang, and Bin Sun. 2007. Multi-document summarization using support vector regression. In *Proceedings of DUC*. Citeseer.

Daniel Marcu. 1997. From discourse structures to text summaries. *Intelligent Scalable Text Summarization*.

Rada Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. Document summarization using conditional random fields. In *IJCAI*, volume 7, pages 2862–2867.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI*, volume 7, pages 2903–2908.