Probabilistic Neural Network Based Text Summarization

Mohamed ABDEL FATTAH
Faculty of Engineering, University of Tokushima
2-1 Minamijosanjima
Tokushima, Japan 770-8506
FIE, Helwan University
Cairo, Egypt
mohafi@is.tokushima-u.ac.jp

Fuji REN
Faculty of Engineering, University of Tokushima
2-1 Minamijosanjima
Tokushima, Japan 770-8506
School of Information Engineering, Beijing University of Posts & Telecommunications
Beijing, 100088, China
ren@is.tokushima-u.ac.jp

Abstract:

This work proposes an approach to address the problem of improving content selection in automatic text summarization by using probabilistic neural network (PNN). This approach is a trainable summarizer, which takes into account several features, including sentence position, positive keyword, negative keyword, sentence centrality, sentence resemblance to the title, sentence inclusion of name entity, sentence inclusion of numerical data, sentence relative length, Bushy path of the sentence and aggregated similarity for each sentence to generate summaries. First we investigate the effect of each sentence feature on the summarization task. Then we use all features in combination to train the probabilistic neural network (PNN) in order to construct a text summarizer model.

Keywords:

Automatic Summarization; probabilistic neural network; statistical model.

1. Introduction

Because of the lack of powerful computers and difficulty in nature language processing (NLP), early work on text summarization focused on the study of text genres such as sentence position and cue phrase [1], [2]. During the 1970s, artificial intelligence (AI) started to be applied [3]-[6]. AI exploited knowledge representations, such as frames or templates, to identify conceptual entities from a text and to extract relationships between entities by inference mechanisms. The major drawback is that limitedly-defined frames or templates may lead to incomplete analysis of conceptual entities. During the 1990s, information retrieval (IR) was employed for the text summarization task [7]-[15]. However, most IR techniques that have been exploited in text summarization focus on symbolic-level analysis, and they do not take into account semantics such as synonymy, polysemy, and term dependency [10].

Recently many experiments have been conducted for the text summarization task. Some were about evaluation of summarization using relevance prediction [16], ROUGEeval package [17], SUMMAC, NTCIR, and DUC [18] and voted regression model [19]. Others were about single- and multiple-sentence compression using "parse and trim" approach and a statistical noisy-channel approach [20] and conditional random fields [21]. Other research includes multi-document summarization [22], [23] and summarization for specific domains [24].

In this work, sentences of each document are modeled as vectors of features extracted from the text. The summarization task can be seen as a two-class classification problem, where a sentence is labeled as ''correct'' if it belongs to the extractive reference summary, or as ''incorrect'' otherwise. We may give the ''correct'' class a value '1' and the ''incorrect'' class a value '0'. In testing mode, each sentence is given a value between '0' and '1' (values between 0 and 1 are continuous). Therefore, we can extract the appropriate number of sentences according to the compression rate. The trainable summarizer is expected to "learn" the patterns which lead to the summaries, by identifying relevant feature values which are most correlated

with the classes "correct" or "incorrect". When a new document is given to the system, the "learned" patterns are used to classify each sentence of that document into either a "correct" or "incorrect" sentence by giving it a certain score value between '0' and '1'. A set of highest score sentences are chronologically specified as a document summary based on the compression rate.

2. Text features

1- f1 = Sentence Position.

We assume that the first sentences of a paragraph are the most important. Therefore, we rank a paragraph sentence according to its position in the paragraph and we consider maximum positions of 5. For instance, the first sentence in a paragraph has a score value of 5/5, the second sentence has a score 4/5, and so on. A score of zero is given to sentences paragraph position greater than 5 since position feature of these sentences is not significant.

2- f2 = Positive keyword in the sentence.

Positive keyword is the keyword frequently included in the summary. It can be calculated as follows:

$$Score_{f_2}(s) = \frac{1}{Length(s)} \sum_{i=1}^{n} tf_i * P(s \in S \mid keyword_i) \qquad (1)$$

Where, s is a sentence, n is the number of keywords in s, tf_i is the occurring frequency of $keyword_i$ in s. We divide the value by the sentence length to avoid the bias of its length (tf_i , n and Length(s) are calculated using sentence "s" from the testing data).

$$P(s \in S \mid keyword_i) = \frac{P(keyword_i \mid s \in S) P(s \in S)}{P(keyword_i)}$$

$$P(keyword_i \mid s \in S) = \frac{\#(sentence\ in\ summary\ ,and\ contains\ keyword_i)}{\#(sentence\ in\ summary)}$$

$$P(s \in S) = \frac{\#(sentence\ in\ training\ corpus\ ,and\ also\ in\ summary)}{\#(sentence\ in\ training\ corpus)}$$

$$P(keyword_i) = \frac{\#(sentence\ in\ training\ corpus\ ,and\ contains\ keyword_i)}{\#(sentence\ in\ training\ corpus)}$$

$P(s \in S \mid keyword_i)$ is calculated from the training corpus (manually summarized articles).

3- f3 = Negative keyword in the sentence.

In contrast to f2, negative keywords are the keywords that are unlikely to occur in the summary. And it can be calculated as follows:

$$Score_{f_3}(s) = \frac{1}{Length(s)} \sum_{i=1}^{n} tf_i * P(s \notin S \mid keyword_i) \qquad (2)$$

4- f4 = Sentence Centrality (similarity with other sentences).

Sentence centrality is the vocabulary overlap between this sentence and other sentences in the document. It is calculated as follows:

$$Score_{f_4}(s) = \left| \frac{Keywords\ in\ s \cap Keywords\ in\ other\ sentences}{Keywords\ in\ s \cup Keywords\ in\ other\ sentences} \right| \qquad (3)$$

5- f5 =Sentence Resemblance to the title.

Sentence resemblance to the title is the vocabulary overlap between this sentence and the document title. It is calculated as follows:

$$Score_{f_5}(s) = \left| \frac{Keywords\ in\ s \cap Keywords\ in\ title}{Keywords\ in\ s \cup Keywords\ title} \right| \qquad (4)$$

6- f6 = sentence inclusion of name entity (proper noun).

Usually the sentence that contains more proper nouns is an important one and it is most probably included in the document summary. The score of f6 is calculated as follows:

$$Score_{f_6}(s) = \frac{\#(proper\ nouns\ in\ s)}{Length(s)} \qquad (5)$$

7- f7 = sentence inclusion of numerical data.

Usually the sentence that contains numerical data is an important one and it is most probably included in the document summary. The score of f7 is calculated as follows:

$$Score_{f_7}(s) = \frac{\#(numerical\ data\ in\ s)}{Length(s)} \qquad (6)$$

8- f8 = sentence relative length.

This feature is employed to penalize sentences that are too short, since these sentences are not expected to belong to the summary. We use the relative length of the sentence, which is calculated as follows:

$$Score_{f_8}(s) = length(s) * average\ sentence\ length \qquad (7)$$

9- f9 = Bushy path of the node (sentence).

The bushiness of a node (sentence) on a map is defined as the number of links connecting it to other nodes (sentences) on the map. Since a highly bushy node is linked to a number of other nodes, it has an overlapping vocabulary with several sentences and is likely to discuss topics covered in many other sentences [13]. The Bushy path is calculated as follows:

$$Score_{f_9}(s) = \#(branches\ connected\ to\ the\ node) \qquad (8)$$

10- f10 = Summation of similarities for each node (aggregate similarity).

Aggregate similarity measures the importance of a sentence. Instead of counting the number of links connecting a node (sentence) to other nodes (Bushy path), aggregate similarity sums the weights (similarities) on the links. Aggregate similarity is calculated as follow:

$$Score_{f_{10}}(s) = \sum Node\ branch\ similarities \qquad (9)$$

3. The proposed automatic summarization model

In the proposed approach, we have two modes of operations:

1- Training mode where features are extracted from 100 manually summarized Arabic documents and used to train the PNN model.
2- Testing mode, in which features are calculated for sentences from 100 Arabic documents. (These documents are different from those that were used for training.) The sentences are ranked according to the sets of feature weights calculated during the training stage. Summaries consist of the highest-ranking sentences.

3.1. Probabilistic neural network (PNN)

Probabilistic Neural Networks are a versatile and efficient tool to classify high-dimensional data [25]-[27]. The network weights and functions are backed by straightforward Bayesian probability, giving them an edge over other network models that have to be gradually optimized using techniques like gradient descent. Bayes' theorem can be used to perform probabilistically optimal classification as follows:

The probability distribution function (PDF) for a feature vector (X) to be of a certain category (class A for example as a summary sentence) is given by:

$$f_a(X) = 1/(2\pi)^{p/2}\,\sigma^p\,(1/n_a)\sum_{i=1}^{n_a}\exp(-(X-Y_{ai})^\tau(X-Y_{ai})/2\sigma^2) \quad (10)$$

Where

$f_a(X)$ = the value of the PDF for class A at point X.

X = test vector to be classified.

i = training vector number.

p = The training vector size.

n_a = number of training vectors in class A.

Y_{ai} = i^{th} training vector for class A.

τ = Transpose

σ = The standard deviation of the Gaussian curves used to construct the PDF.

Introducing a term to represent the relative number of trials in each category (n_a / n_{total}) simplifies the expression. Hence $(1/n_a)$ term is canceled out as follows:

$$f_a(X) = 1/(2\pi)^{p/2}\,\sigma^p\,(1/n_{total})\sum_{i=1}^{n_a}\exp(-(X-Y_{ai})^\tau(X-Y_{ai})/2\sigma^2) \quad (11)$$

Terms common to all classes such as $1/(2\pi)^{p/2}$, σ^p and n_{total} can also be eliminated, leaving the following formula:

$$f_a(X)\,\alpha\sum_{i=1}^{n_a}\exp(-(X-Y_{ai})^\tau(X-Y_{ai})/2\sigma^2) \quad (12)$$

Hence the classifier can be expressed as follows:

For a feature parameter X to belong to a category (r); the following formula must be verified:

$$\sum_i \exp(-(X-Y_{ri})^\tau(X-Y_{ri})/2\sigma^2) \geq \sum_i \exp(-(X-Y_{si})^\tau(X-Y_{si})/2\sigma^2) \quad (13)$$

where (s) represents the other category.

The expression $(X-Y_{ri})^\tau(X-Y_{ri}) = (X^\tau X)-(2X^\tau Y_{ri})+(Y_{ri}{}^\tau Y_{ri}) = 1-(2X^\tau Y_{ri})+1 = 2-(2X^\tau Y_{ri})$ allowing formula (13) to be simplified as follows:

$$\sum_i \exp((X^\tau Y_{ri}-1)/\sigma^2) \geq \sum_i \exp((X^\tau Y_{si}-1)/\sigma^2) \quad (14)$$

Figure 1 shows the structure of the P-NNT implementation. Each neuron in layer one receives an element of vector X (x1, x2.... xn) of a certain sentence to be classified as summary or not summary sentence, (n=10 in our case). The weight matrix scaling these inputs is formed by the elements of the training vectors divided by the constant (σ^2). The first layer has a bias of $-1/\sigma^2$. The inputs of layer one are summed, producing $(X^\tau Y_{ri}-1)$. Then, this value is divided by σ^2 and the exponential transfer function is applied, resulting in outputs of $\exp((X^\tau Y_{ri}-1)/\sigma^2)$ and $\exp((X^\tau Y_{si}-1)/\sigma^2)$, where (s) represents the remaining category. The second layer has two neurons: each one is associated with a specific category of output mentioned before (summary category and not summary category). The inputs from the first layer of each category are summed to produce the expressions $\sum_{i=1}^{m}\exp((X^\tau Y_{ri}-1)/\sigma^2)$ and $\sum_{i=1}^{m}\exp((X^\tau Y_{si}-1)/\sigma^2)$. The output of each neuron represents the probability that the vector X belongs to its class. The neuron in layer 2 with the greatest output determines how the vector is classified.

4. Experimental results

4.1. The Arabic data

200 Arabic articles in the domain of politics were collected from the Internet archive. 100 Arabic articles were manually summarized using compression rate of 30%. These manually summarized articles were used to train the previously mentioned model. The other 100 Arabic articles were used for testing. The average number of sentences per Arabic article is 26.8.

We use an intrinsic evaluation to judge the quality of a summary based on the coverage between it and the manual summary. We measure the system performance in terms of recall from the following formula:

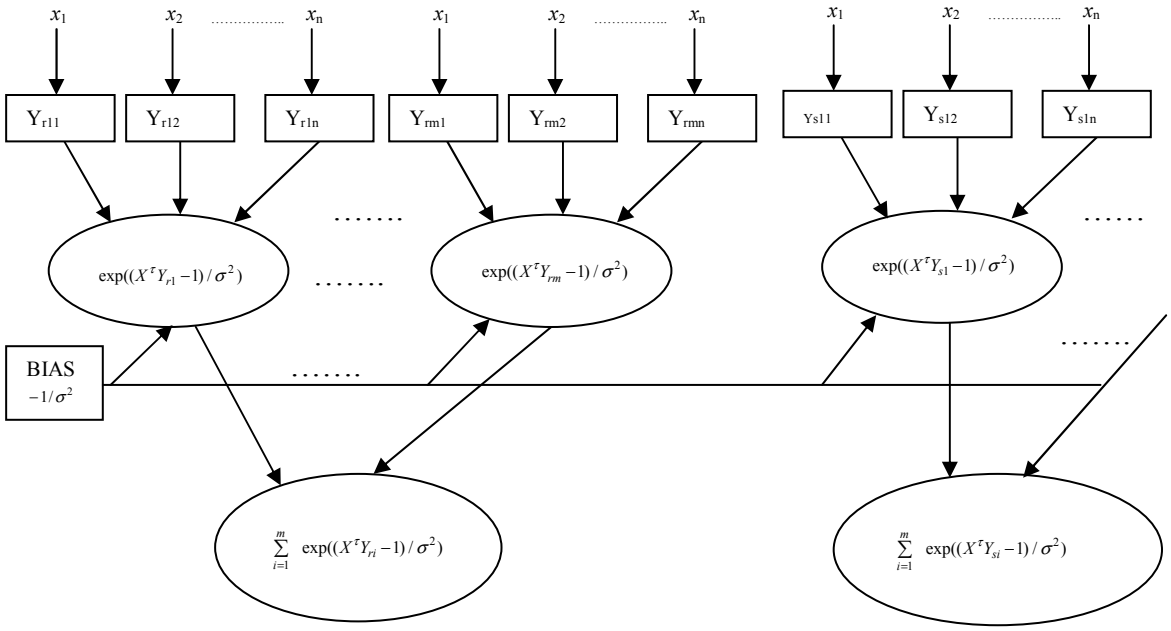

Figure 1. The structure of PNN implementation

$$R = \frac{|S \cap T|}{|T|} \tag{15}$$

Where, R is the recall, T is the manual summary and S is the machine-generated summary. Moreover, a 95% confidence interval was considered for all tests to estimate the statistical uncertainty.

4.2. The effect of each feature on summarization performance

In this section, we investigate the effect of each feature parameter on summarization by using the individual score of each feature. For instance, to investigate the first feature (sentence position) on summarization performance, we use the following equation:

$$Score(s) = Score_{f_1}(s) \tag{16}$$

Table1 shows the summarization recall associated with each feature for different compression rates.

4.3. The results of probabilistic neural network (PNN)

The system is extracting features from the 100 Arabic manually summarized documents and uses them to train probabilistic neural network. Use the other 100 Arabic documents as a testing set. Then apply the sentences of these documents as inputs to the Probabilistic Neural Network after feature extraction step as follows:

1- Extract features from the sentences of the document.
2- Construct the feature vector X as shown in figure 1.
3- Use this feature vector as an input of the probabilistic neural network.
4- Save the output of the neural network for each sentence.
5- Rank all document sentences based on their scores then arrange them in a descending order.
6- Chronologically select the set of sentences of highest scores based on the required compression rate.

Table2 shows the results of PNN for the 100 Arabic articles.

4.4. Discussion

It is clear from table 1 that the most important text feature for summarization is f9 (Bushy path) since it gives the best results. It is reasonable, since the sentence that has a maximum number of branches should convey the most important part in the article. f4 (centrality) also gives good results since it conveys the vocabulary overlap between this sentence and other sentences in the document. Usually, the document title conveys the main topic of this document. Therefore, f5 (sentence resemblance to the title) which is the vocabulary overlap between this sentence and the document title gives good results. The lowest results are associated with f7 (sentence inclusion of numerical data) since most of political and religious articles do not contain many numerical data. Therefore, the system ranks a sentence that does not contain numerical data according to its position.

Table 1
The summarization recall associated with each feature for different compression rates

Compression rate (CR)	10%	95% Confidence Interval	20%	95% Confidence Interval	30%	95% Confidence Interval
R(f1)	0.3523	0.3315, 0.3732	0.3625	0.3417, 0.3834	0.3536	0.3328, 0.3745
R(f2)	0.3428	0.3206, 0.3650	0.3414	0.3192, 0.3636	0.3342	0.3120, 0.3564
R(f3)	0.3023	0.2834, 0.3213	0.2923	0.2734, 0.3113	0.2987	0.2798, 0.3177
R(f4)	0.4023	0.3775, 0.4271	0.4045	0.3882, 0.4209	0.4086	0.3923, 0.4250
R(f5)	0.3623	0.3413, 0.3834	0.3723	0.3513, 0.3934	0.3873	0.3663, 0.4084
R(f6)	0.3243	0.3114, 0.3371	0.3134	0.3005, 0.3262	0.3256	0.3127, 0.3384
R(f7)	0.2645	0.2423, 0.2867	0.2635	0.2413, 0.2857	0.2535	0.2313, 0.2757
R(f8)	0.2847	0.2658, 0.3037	0.2798	0.2609, 0.2988	0.2865	0.2676, 0.3055
R(f9)	0.4265	0.4017, 0.4513	0.4236	0.3988, 0.4484	0.4137	0.3889, 0.4385
R(f10)	0.3825	0.3615, 0.4036	0.3634	0.3424, 0.3845	0.3736	0.3526, 0.3947

Table 2
The PNN approach performance evaluation based on recall

Compression rate (CR)	10%	20%	30%
Recall (R)	0.4592	0.4665	0.4712
95% Confidence Interval	0.4423, 0.4761	0.4502, 0.4829	0.4508, 0.4917

5. Conclusions and future work

In this work, we have investigated the use of probabilistic neural network (PNN) for automatic text summarization task. We have applied our new approach on a sample of 100 Arabic articles. Our approach results outperform each feature approach results. Our approach has been used the feature extraction criteria which gives researchers opportunity to use many varieties of these features based on the used language and the text type. Some text features are language independent.

In the future work, we will investigate the effect of the output summary from this system on information retrieval and cross language information retrieval systems.

Acknowledgment

This research has been partially supported by the Japan Society for the Promotion of Science (JSPS), Grant No. 07077 and the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (B), 19300029.

References

[1] Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM), 16(2), 264–285.*

[2] Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research andDevelopment, 2(2), 159–165.*

[3] Azzam, S., Humphreys, K., & Gaizauskas, R. (1999). Using coreference chains for text summarization. *In Proceedings of the ACL'99, College Park, MD, USA, 77–84.*

[4] McKeown, K., & Radev, D. R. (1995). Generating summaries of multiple news articles. *In Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'95), Seattle, WA, USA, 74–82.*

[5] Schank, R., & Abelson, R. (1977). Scripts, Plans, Goals, and Understanding. *Hillsdale, NJ: Lawrence Erlbaum Associates.*

[6] Young, S. R., & Hayes, P. J. (1985). Automatic classification and summarization of banking telexes. *In Proceedings of the 2nd Conference on Artificial Intelligence Application, 402–408.*

[7] Aone, C., Okurowski, M. E., Gorlinsky, J., & Larsen, B. (1997). A scalable summarization system using robust NLP. *In Proceedings of the ACL'97/EACL'97 workshop on intelligent scalable text summarization, Madrid, Spain, 10–17.*

[8] Goldstein, J., Kantrowitz, M., Mittal, V., & Carbonell, J. (1999). Summarizing text documents: sentence selection and evaluation metrics. *In Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'99), Berkeley, CA, USA,121–128.*

[9] Gong, Y., & Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. *In Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'01), New Orleans, LA, USA, 19–25.*

[10] Hovy, E., & Lin, C. Y. (1997). Automatic text summarization in SUMMARIST. *In Proceedings of*

the ACL'97/EACL'97 workshop on intelligent scalable text summarization, Madrid, Spain, 18–24.

[11] Kupiec, J., Pedersen, J., Chen, & F. (1995). A trainable document summarizer. *In Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'95), Seattle, WA, USA, 68–73.*

[12] Mani, I., & Bloedorn, E. (1999). Summarizing similarities and differences among related documents. *Information Retrieval, 1(1–2), 35–67.*

[13] Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing & Management, 33(2), 193–207.*

[14] Teufel, S. H., & Moens, M. (1997). Sentence extraction as a classification task. *In Proceedings of the ACL'97/EACL'97 workshop on intelligent scalable text summarization, Madrid, Spain, 58–65.*

[15] Yeh, J. Y., Ke, H. R., & Yang, W. P. (2002). Chinese text summarization using a trainable summarizer and latent semantic analysis. *In Lecture Notes in Computer Science (Vol. 2555). In Proceedings of the 5th international conference on Asian digital libraries (ICADL'02), Singapore. Berlin: Springer-Verlag, 76–87.*

[16] Hobson, S., Dorr, B., Monz, C., & Schwartz, R. (2007). Task-based evaluation of text summarization using Relevance Prediction *Information Processing & Management*, *43(6)*,*1482-1499.*

[17] Sjöbergh, J. (2007). Older versions of the ROUGEeval summarization evaluation system were easier to fool. *Information Processing & Management*, *43(6)*, *1500-1505.*

[18] Over, P., Dang, H., & Harman, D. (2007). DUC in context. *Information Processing & Management*, *43(6)*, *1506-1520.*

[19] Hirao, T., Okumura, M., Yasuda, N., & Isozaki, H. (2007). Supervised automatic evaluation for summarization with voted regression model. *Information Processing & Management*, *43(6)*, *1521-1535*

[20] Zajic, D., Dorr, B., Lin, J., & Schwartz, R. (2007). Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, *43(6), 1549-1570.*

[21] Nomoto, T. (2007). Discriminative sentence compression with conditional random fields. *Information Processing & Management*, *43(6)*, *1571-1587.*

[22] Vanderwende, L., Suzuki, H., Brockett, C., & Nenkova, A. (2007). Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, *43(6), 1606-1618.*

[23] Harabagiu, S., Hickl, A., & Lacatusu, F. (2007). Satisfying information needs with multi-document summaries. *Information Processing & Management*, *43(6)*, *1619-1642.*

[24] Moens, M. (2007). Summarizing court decisions. *Information Processing & Management*, *43(6) 1748-1764.*

[25] Specht, D.F. (1990). Probabilistic neural networks. *Neural Networks, 3(1), 109-118.*

[26] Ganchev, T., Tasoulis, D. K., Vrahatis, M. N., Fakotakis, N. (2003). Locally recurrent probabilistic neural networks for text independent speaker verification. *Proc. of the EuroSpeech, 3, 1673–1676.*

[27] Cain, B. J. (1990). Improved probabilistic neural networks and its performance relative to the other models. *Proc. SPIE, Applications of Artificial Neural Networks, 1294, 354–365.*