A Relevance-Based Language Modeling approach to DUC 2005 *

Jagadeesh J, Prasad Pingali and Vasudeva Varma
International Institute of Information Technology
Hyderabad, India

Abstract

The task in Document Understanding Conferences (DUC[1]) 2005 is to generate fixed length, user oriented, multi document summary. Our approach to address this task is primarily motivated by the observation that metrics based on key concepts overlap give better results when compared to metrics based on n-gram and sentence overlap. In this paper, we present a sentence extraction based summarization system which scores the sentences using Relevance Based Language Modeling, Latent Semantic Indexing and number of special words. From these scored sentences, the system generates a summary of required granularity. Our summarization system was ranked 3^{rd},4^{th},8^{th} and 17^{th} in ROUGE-SU4, ROUGE-2, responsiveness and linguistic quality evaluations respectively. In post DUC analysis we found that LSI has negative effect on the systems performance, and the performance gained by 5.4% when it is implemented using language modeling and number of special words.

1 Introduction

DUC 2005 marks a major change from its previous years. *The system task in 2005 was to synthesize from a set of 25-50 documents a brief, well-organized, fluent answer to a need for information that cannot be met by just stating a name, date, quantity, etc.* That is, given a user's information need, as a DUC topic, and a cluster of documents relevant to the DUC topic, the system needs to create, from the document set, a summary which answers the information need expressed. A DUC topic consists of three components. First part is the title of the topic. Second is the actual information need expressed as single question or a collection of more than one question. Last part is granularity of summary, which indicates the requirement of specificity of information in the summary. The summary should meet the level of granularity specified in the DUC topic. The task aimed at modeling real world complex question answering. This task can be seen as topic-oriented, informative multi-document summarization, where the goal is to produce a single text as a compressed version of a set of documents with all and only the relevant information.

A system that addresses such a complex task may involve the following stages; information need enrichment, content selection and summary generation. In information need enrichment, the system parses and analyzes the information need, enriches the analyzed information either by using world knowledge or the document cluster, and represents the information in a machine readable format so that it can be used in later processing. Content selection would involve identification and selection of the content relevant to the information need from the cluster of documents. The identification of relevant information would need significant amount of natural language processing of documents in the cluster. The summary generation include the generation of summary from the content extracted in second phase.

Building such systems will not only take considerable amount of resources but also significant time to produce the summary, as it involves deep analysis of large number of sentences, once the input and the data cluster is provided. In this paper we have discussed a sentence extraction based summarization to address this problem. Our work is primarily motivated by the observation[3] that in information synthesis tasks like DUC 2005, metrics based on key concepts overlap give better results than metrics based on n-gram overlap and sentence overlap. Hence we represent query words and documents in semantic space and enable a better relevance scoring of sentences towards the information need. These scored sentences are ranked and best sentences are selected to form the summary.

*http://www-nlpir.nist.gov/projects/duc/duc2005/tasks.html
[1]http://duc.nist.gov

The next section (section 2) gives a general overview of our summarization system, which is followed (section 3) by a detailed description of different features that we used to rank the sentences. In section 4, we report the performance of our summarization system when compared to other systems. In this section we also present the post DUC analysis interms of the contribution by individual features towards the performance of our summarization system.

2 Our System

Our system, IIITH-Sum, was built in a very short time and performed better than many other summarization systems. The system was ranked 3rd, 4th and 8th in ROUGESU4, ROUGE2 [7] and responsiveness evaluations respectively. This performance was achieved by using robust and proven statistical features.

We have used the framework of MEAD[8] system along with our own features. MEAD is an extractive summarization environment based on three-step architecture. In the first step the system extracts the feature values for each sentence and constructs a feature vector for each sentence. In the second step, the classifier takes these individual feature values of a sentence and computes score for the sentence. In the third step, sentences are re-ranked and top sentences are selected till the summary length is met. The re-ranking step, in MEAD, involves an adjustment of sentence scores based on the redundancy of information and sentence level dependencies at discourse level like anaphoric relationship etc.. A lot of sophisticated classifiers and re-rankers with the above mentioned functionalities are being distributed along with MEAD, but we did not include all those functionalities in our summarization system. We used *only* the framework of MEAD and the performance reported is a direct result of the features that we have used.

In our system, each sentence from input document cluster is assigned a relevance score based on three features. The classifier takes these individual feature values of a sentence, computes a weighted linear combination of these feature values and this is treated as the sentence score through the rest of processing. Further these sentences are ordered by the score in descending order, and the system iteratively decides whether to add each sentence to the summary or not. At each iteration, the decision is decided by the length of the required summary and the similarity of the new sentence with the already selected summary sentences.

Since a DUC topic expresses a complex information need, most topics contain more than one question. So while calculating the sentence relevance, if we treat the whole topic as a single question, a sentence which is more general and talking about more than one question will get higher score when compared to a sentence which is more precise and addresses one question of the topic. As a result, a wrong sentence will be selected towards the final summary. In order to overcome this problem we have first divided the DUC topic into possible simple questions. And while calculating the sentence relevance for a feature, we compute its score with each of the questions individually and consider the maximum of all scores as its score for that feature. In this way the system would be able to find more precise sentences. The decomposition of DUC topic into multiple simple questions is based on occurrence of some tokens like '?', wh-words and conjunctions like 'and'.

3 Features

We have used relevance-based language modeling [6] and latent semantic indexing technique [2] to compute the relevance of a sentence towards the information need. Generally these features are used to calculate the document relevance towards user's query. In this paper we have extended them to sentence level and were used as sentence level features. These two techniques were chosen based on the intuition that LSI favors improvement in recall of the final summary, while language modeling favors improvement in precision. We had to go by this intuition since we did not have model summaries to verify this intuition. The third feature helps in computing the granularity of the information provided in a sentence towards the query, which is calculated based on the number of special words in the given sentence. The following subsections discuss in detail how these features can be used to quantitatively measure the sentence relevance towards the information need.

3.1 Relevance-Based Language Model

Statistical language models[4] have recently been used heavily in information retrieval applications. In language modeling both query formulation and retrieval of relevant documents are modeled as simple probability

mechanisms. Query formulation is modeled as translation of the information need into search engine request. Retrieving the relevant documents is modeled as the generation of a random query from a relevant document. That is, each document is treated as a language sample and query as a generation process. The retrieved documents are ranked based on the probability of producing the query from the corresponding language model of these documents. One difficulty in applying statistical language modeling to information retrieval is the sparseness of data to compute the document model. The relevance based language modeling[6] is a significant improvement in estimating the relevance model of a document when no training data is available in the form of relevance judgments. The important deviation of relevance based language model from other statistical language models is that, it does not assume the query as a sample from any specific document model, instead it assumes both the query and the document as samples from an unknown relevance model R, hence it is able to overcome the problem of sparseness in the training data.

The relevance based language model approximates $P(w/R)$, the probability of observing a word w in the documents relevant to a particular information need R, using $P(w/Q)$, where $Q = q_1, q_2 ... \ q_k$ is the information need expressed in the form of query words. By definition, the conditional probability can be expressed in terms of the joint probability of observing w with the query words $q_1, q_2 \ ... \ q_k$. [2]

$$P(w/R) \ \approx \ P(w/Q) \ = \ P(w/q_1, q_2 \ ... \ q_k) \ = \ \frac{P(w, q_1 \ ... \ q_k)}{P(q_1 \ ... \ q_k)} \tag{1}$$

Several ways of estimating the joint probability are explored in [1]. Based on their experimental results and the simplicity in terms of implementation, we have decided to use the conditional sampling[6] to compute the required joint probability. Conditional sampling assumes the query words $q1, \ ... \ , q_k$ to be independent of each other while keeping their dependencies on w intact. Using this assumption the joint probability can be calculated as:

$$P(w, q_1, \ ... \ q_k) \ = \ P(w) \prod_{i=1}^{k} P(q_i/w) \tag{2}$$

Now the required term dependencies, $P(q_i/w)$, can be incorporated into the above expression using the probabilistic interpretation of Hyperspace Analogue to Language(HAL) model.

3.1.1 Hyperspace Analogue to Language (HAL)

Hyperspace Analogue to Language[5] model constructs the dependencies of a word w on other words based on their occurrence in the context of w in a sufficiently large corpus. The intuition underlying HAL is that when a human encounters a new concept, they derive its meaning from accumulated experience of the context in which the concept appears. Thus the meaning of the new concept can be learnt from its usage with other concepts within the same context. Lund and Burgess [5] discusses the use of lexical co-occurrence to construct high dimensional semantic spaces in which a word can be represented as a point. The representational model of this space can be constructed automatically from a corpus of text.

The construction of HAL space can be seen as a vector representation of each word w, occurring in the vocabulary T, in a high dimensional space spanned by different words in the vocabulary. This process results in a $|T| \times |T|$ HAL matrix, where $|T|$ is the number of different words in the vocabulary. The HAL matrix is constructed by taking a window of length k words and moving it across the corpus at one term increments. All words in the window are said to co-occur with the first word with strengths inversely proportional to the distance between them. In our system we have considered the co-occurrence to be bi-directional, because in general it is agreed that preserving the word order is not useful for IR. The weights assigned to each co-occurrence of terms are accumulated over the entire corpus. That is, if $n(w, k, w')$ denote the number of times word w' occurs k distance away from w when considered a window of length K, and $W(k) \ = \ K - k + 1$ denotes the strength of this co-occurrence between the two words, then

$$\text{HAL}(w'/w) \ = \ \sum_{k=0}^{K} W(k) \, n(w, k, w')$$

The HAL space naturally leads itself to probabilistic interpretation as term co-occurrence counts can be used to define the conditional probabilities. The pHAL, probabilistic HAL, can be interpreted as, given a word w

[2] k is used as a temporary variable in this paper whereever necessary

what is the probability of associating a word w' with w in a window of size K. This can be expressed in terms of probability of observing w' at a distance of $k < K$ from w, as

$$\text{pHAL}(w'/w) \;=\; \sum_{k=0}^{K} P(k)\; P(w'/w,k)$$

where $P(w'/w,k) = \frac{n(w,k,w')}{\sum_{w''} n(w,k,w'')}$

$$\begin{aligned}\text{pHAL}(w'/w) &= \sum_{k=0}^{K} P(k)\; P(w'/w,k) \\ &= \frac{\sum_k P(k)\; n(w,k,w')}{\sum_k \sum_{w''} n(w,k,w'')}\end{aligned}$$

$\sum_{w''} n(w,k,w'')$ is the number of times some word w'' followed the word w at a distance of k. This is the number of times w has occurred at the start of a window of length k, which is equal to the unigram frequency of the word w, if the corpus is sufficiently large. $\sum_k n(w,k,w'')$ is the total number of times w'' followed w in the given window. From both these observations,

$$\begin{aligned}\sum_k \sum_{w''} n(w,k,w'') &= n(w) \times K \qquad n(w) \text{ is the unigram frequency of } w \\ \text{pHAL}(w'/w) &= \frac{\sum_k P(k)\; n(w,k,w')}{n(w) \times K} \\ &= c\,\frac{\text{HAL}(w'/w)}{n(w) \times K} \qquad (3)\end{aligned}$$

where $P(k)$ is the prior probability, and it is assumed the priors are proportional to the co-occurrence strength. To ensure that we obtain valid probability distribution, the constraint $\sum_{w'} pHAL(w'/w) \;=\; 1$ is imposed.

3.1.2 Sentence Score

From equations 1,2 and 3, the relevance of a word towards the information need or the probability of observing a word w in sentences relevant to an information need can be calculated as,

$$P(w/R) \approx P(w/Q) = \frac{P(w,q_1,\;\ldots\;,q_k)}{P(Q)} \approx \frac{P(w)}{P(Q)} \prod_{q_j} P(q_j/w) = \frac{P(w)}{P(Q)} \prod_{q_j} \text{pHAL}(q_j/w)$$

Assuming that the different words in a sentence are independent and removing the constant terms, the relevance of a sentence S, can be expressed as,

$$P(S/R) = \prod_{w_i \in S} P(w_i/R) = \prod_{w_i \in S} \frac{P(w_i)}{P(Q)} \prod_{q_j} \text{pHAL}(q_j/w_i) \;\approx\; \prod_{w_i \in S} P(w_i) \prod_{q_j} \text{pHAL}(q_j/w_i)$$

Based on the discussion in [5], we set the window size to be equal to 8, while constructing the HAL matrix. If the window size is set to 1, then this model reduces to vector space model with the number of key words matched as the relevance score of the sentence towards the information need. Even when the window size is set to a value other than 1, this will still prefer the sentences which contain the user's query words or other words which occur frequently with query words. In other words, this HAL will help in extracting the sentences which are more related to the query in their surface forms.

3.2 Latent Semantic Indexing

In a multi-document scenario, since the input documents are authored by different persons and a concept can be represented in many ways, the number of matching keywords between the user's information need (expressed in the form of questions) and sentences which satisfy the user's information need may not be

significant. Taking this fact into consideration, apart from keyword matching techniques, we need to relate the user's information need and the sentences semantically. Latent Semantic Indexing (LSI)[2] is a technique to overcome the problems of lexical matching, by using statistically derived concepts instead of individual words, for relevance ranking. LSI assumes and tries to extract the underlying or latent structure in word usage that is partially obscured by variability in the word choice. In general, Information Retrieval applications apply LSI at document level, but in this paper we have applied LSI technique to sentence and used it to rank the sentences based on their relevance towards the information need.

Using LSI, we project information need and sentences into "latent" semantic dimension space. In this latent space, a query and a sentence can have high similarity even if they do not share any terms, as long as their terms are semantically similar. First by observing the corpus, the system builds a word-by-sentence matrix. Then LSI applies, a particular mathematical technique called Single Value Decomposition (SVD) to decompose word-by-sentence matrix into a product of three different matrices. Let $A_{w\times s}$ be the word-by-sentence matrix, whose rows represent words and columns represent sentences. And let the singular values of A be $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r$ where r is rank of the matrix A. Then SVD decomposes A into product of three matrices $A = UDV^T$, where $D =$ diag $(\sigma_1, ..., \sigma_r)$ is an $r \times r$ matrix, $U = (u_1, ..., u_r)$ is an $w \times r$ matrix whose columns are orthogonal and $V = (v_1, ..., v_r)$ is an $s \times r$ matrix which is also column-orthogonal. SVD is a method for rotating the axes of the n-dimensional space such that first axis runs along the direction of largest variation in the data, second dimension runs along direction with the second largest variation and so forth. So by keeping the first $k < r$ dimensions and omitting the rest, we can still maintain the representation of the data points with a little deviation from their representation in the original space.

By restricting the matrices U, D, V to their first $k < r$ rows, $D' =$ diag $(\sigma_1, ..., \sigma_k)$, $U' = (u_1, ..., u_k)$ and $V' = (v_1, ..., v_k)$; $A' = U'D'V'^T$ is the best possible approximation to A in k dimensions. The rows of $V'_{s\times k} \cdot D'_{k\times k}$ are used to represent the sentences in lower dimension space. And the user's information need can also be projected into latent dimensions using $Q' = Q^T_{t\times 1} U_{t\times k} D^{-1}_{k\times k}$. Now the reduced sentences(S_i) are compared to the new query vector, and are ranked by their similarity measure with the information need. We have used cosine of the angle between the query and sentence as the similarity measure. Hence the relevance score of a sentence is given as:

$$\cos(Q', S_i) = \frac{\vec{Q'} \cdot \vec{S_i}}{|\vec{Q'}||\vec{S_i}|}$$

In our system, we have fixed the number latent dimensions to be equal to 20. That is, the words and sentences are reduced into a space with 20 dimensions. These 20 reduced dimensions are orthogonal to each other and relevance scoring of a sentence with respect to information need is done in this space. The sentence ranking is done in a semantic space, hence LSI might prefer a sentence which doesn't contain any of the DUC topic keywords but still is related to user's query semantically. At the same time, it can also introduce noise into the summary if the information need is not focussed enough.

3.3 Number of Special Words (NSWs)

Another important aspect of DUC 2005 task is the granularity of the summary towards the information need. Granularity is a binary valued feature, either specific or general, which indicates the requirement of specificity of information in the summary. A "specific" summary should describe and name specific events, people, places etc. while a "general" summary refers to categories/types of things but it can also refer to specific information if space allows. We have used a simple heuristic to approximate the granularity of the information, i.e. to identify the specificity of the information expressed in each sentence.

The heuristic is based on the observation that, in a sentence most of the specific information is described in the form of named entities like names, dates or numbers etc.. And generally all of them are expressed using a combination of different characters, i.e. they require the use of characters other than in range [a-z]. Therefore we have considered a word which contains characters other than the characters in range [a-z] as a special word, with the word at beginning of a sentence is considered as an exception. Now the number of special words (NSWs) in a sentence indicate the specificity of information mentioned in the sentence.

4 Results

In the system that we have submitted, sticking to the assumption that we have made i.e. LSI may introduce noise into the summary when the user's query is not focussed, we have used only the title of DUC topic as the query for LSI feature. For HAL feature, whole DUC topic was the input, and the information need is treated as a collection of multiple independent questions. Since number of special words is based on a sentence it doesn't make any difference whether we treat the DUC topic as a single question or as a collection of multiple independent questions. In DUC, the evaluation of peer summaries was done both manually, for the responsiveness, and by automatic evaluation techniques like ROUGE. Manual evaluation includes scoring each summary, based on how relatively well it responds to the DUC topic, on a 1 (worst) to 5 (best) scale. Responsiveness was measured primarily in terms of the amount of information in the summary that actually helps to satisfy the information need expressed in the DUC topic, at the level of granularity requested.

System ID	Responsiveness	ROUGE-2	ROUGE-SU4	Linguistic Quality
Human Mean	35.25	0.1025	0.1624	4.86
10	20.88 (1)	0.0698 (3)	0.1252 (5)	2.92 (28)
5	20.54 (2)	0.0674 (6)	0.1232 (7)	3.53 (6)
4	20.09 (3)	0.0685 (5)	0.1277 (4)	3.40 (10)
15	19.91 (4)	0.0725 (1)	0.1316 (1)	3.04 (25)
29	19.78 (5)	0.0609 (13)	0.1138 (18)	3.50 (8)
11	19.64 (6)	0.0642 (7)	0.1225 (7)	3.16 (22)
17	19.61 (7)	0.0717 (2)	0.1297 (2)	3.53 (6)
8	19.44 (8)	0.0696 (4)	0.1279 (3)	3.21 (17)
7	18.81 (9)	0.0627 (11)	0.1189 (10)	3.25 (16)
14	18.77 (10)	0.0634 (8)	0.1176 (13)	3.65 (3)
BaseLine	12.61	0.0402	0.0871	4.41

Table 1: Official scores of summarization systems at DUC-2005, sorted based on responsiveness scores

Table 1, shows the performance of our system when compared to the best 10 systems interms of responsiveness evaluations. In each case the rank obtained by a system, under the evaluation criteria mentioned as the column name, is shown in braces. Our summarization system, with system ID 8, was ranked 8^{th},4^{th}, 3^{rd} and 17^{th} in responsiveness, ROUGE-2, ROUGE-SU4 and linguistic quality evaluations respectively. It can be observed from the evaluations that our system did well in ROUGE scores, while it scored relatively low in responsiveness evaluation. We tried to analyze the reason behind this phenomenon. Our understanding is that the system looks for concepts in sentences which are very close to concepts expressed in information need. We then rank the sentences based on the number of such concepts found in each sentence. In this process we are not performing any deeper analysis such as, an understanding how these concepts are related to each other or what is the semantic content of the sentence chosen. Therefore the sentences chosen by the system hold a good chance of satisfying the information need, but this is not always the case. At the same time, since ROUGE also ranks the peer summary based on the number of concepts that matched with model summary, our peer summaries got high scores in ROUGE evaluation.

We have observed that, for DUC topic 442 which asks for specific summary, our summary got a very low ROUGE score (ROUGE-SU4 score of 0.06) while a high responsiveness score (4 out of 5). While trying to analyze the reason, we found that the peer summary has a lot of specific details which are not matching with model summaries but are still responsive to the information need. This could also be a possible reason for the difference in systems rank in ROUGE and responsiveness evaluations.

We have also tried to evaluate the contribution of individual features towards the final summary. Table 2 show the average recall of system generated summaries for DUC topics when compared to model summaries using ROUGE scores. When more than one feature are used, a weighted linear combination of these individual feature values is used to compute the sentence score. We have assigned an equal weight of 1 to both HAL and LSI while a weight of 0.01 is given to number of special words (measure of granularity). From the results (in table 2), one can see that the influence of LSI feature was negative on overall performance of the system.

Eval. Criteria	HAL	HAL + NSWs	LSI	LSI + NSWs	HAL + LSI + NSWs
ROUGE-1	0.37316	0.37416	0.31249	0.31621	0.36495
ROUGE-2	0.07697	0.07792	0.04677	0.04797	0.07132
ROUGE-SU4	0.13658	0.13774	0.10042	0.10219	0.13065

Table 2: Performance of features

Language modeling on the other hand worked much better in terms of the ROUGE scores. After implementing the system with only two features, namely HAL(language model) and number of special words we found the average ROUGE-SU4 score of the system gained by 5.4% over the previous one. This might be either because of choosing wrong weights while calculating the weighted linear combination or HAL feature is able to capture more information than what LSI does. The ROUGE scores differ by a very little amount when the measure for granularity is included as a feature.

It is also observed that the system failed to capture inter dependencies in information need. If the information need is such that some part of it modifies the rest of questions in the topic, since we are not using any parsing for the query, the system failed to identify these dependencies and hence it failed to generate an appropriate summary. This explains the low responsiveness score of 1 for the topics d332h and d383j, but it is also observed that the rest of the systems also got low scores in these two topics. If the information need is less complex and has sufficient explanation within itself (in terms of its length), then the system generated relatively good summary. This observation is also supported by the responsiveness evaluation. As evident from the evaluation results for linguistic-quality of our system, we made no efforts on making the summary cohesive and neither the sentence reduction techniques nor paraphrasing of sentences was used.

5 Conclusion

In this paper we have presented a sentence extraction based technique to generate a 250 word summary from a document cluster, which answers the user's information need provided as a DUC topic, at the level of granularity required. We have applied IR techniques like Relevance-based language modeling and Latent Semantic Indexing to rank the sentences based on their relevance towards the user's information need. The top sentences are selected to form the summary till the summary length reaches a maximum of 250 words. The performance of our system was better than the average performance of all the systems.

In this paper, we have considered sentence as a basic unit, i.e. sentences were scored, ranked and summary is generated out of them. The features that have been used were able to identify the relevant sentences correctly. But typically in most cases, documents do not contain sentences which exactly answer the information need. So treating a sentence as basic unit and generating summary based on the sentences is not sufficient. The fact that the best peer summaries generated are way behind the average model summaries in terms of responsiveness evaluation, support this observation. A sentence extraction technique such as this could be a good preprocessing stage for the later summary generation component, which takes these extracted sentences and generates the summary.

References

[1] Peter Bruza and Dawei Song. A comparison of various approaches for using probabilistic dependencies in language modeling. In *Proceedings of International ACM SIGIR conference on Research and development in informaion retrieval*, pages 419–420, 2003.

[2] C.H. Papadimitriou, P. Raghavan, H. Tamaki, S.Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of ACM Conference on Principles of Database Systems (PODS)*, 1998.

[3] Enrique Amigo, Julio Gonzalo, Victor Peinado, Anselmo Penas, Felisa Vedejo Enrique Amigo. An emperical study of information synthesis tasks. In *Proceedings of Association for Computational Linguistics (ACL), july 2004, Barcelona.*

[4] Bruce Croft B Fei Song. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321, 1999.

[5] Lund K and Burgess C. Producing high-dimensional semantic spaces from lexical co-occurrence. In *Behavior Research Methods, Instrumentation, and Computers*, pages 203–208, 1996.

[6] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *International ACM SIGIR conference on Research and development in Information Retrieval*, pages 120–127, 2001.

[7] Chin-Yew Lin and E.H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), Edmonton, Canada*, 2003.

[8] Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Elliott Drabek, Wai Lam, Danyu Liu, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, and Adam Winkel. The MEAD Multidocument Summarizer. http://www.summarization.com/mead/, 2003.