This chapter discusses how the sūtra like representation is identified from the UNL-RST-saṅgati discourse structure, discussed in the previous chapter. The motive behind identifying a sūtra is to represent the UNL-RST-saṅgati discourse structure in a short and crisp form, and also inherit the essential qualities such as, coherence, semantics and language independence. The Nannool defines the characteristics of a sūtra as follows (Pavananthy Munivar 1994).

Transliteration of the text:

อารกอกุล karimā nokku tavalaip
Pāyṭṭu paruntiṇ vīṟ vaṇṇa cūṭṭira nilai

Meaning of the text:

“sūtras have the characteristics of a river’s flow, lion’s vision, frog’s jump and eagle’s flight”.

The river flow characteristic represents a string of sūtras that are linked and interpreted in one direction. The lion has the ability to look forward and turn backward. This characteristic represents sūtras that are linked with the previous and next sūtras. Frogs jump covering a considerable amount of space on the ground. This characteristic represents sūtras that are
connected with sūtrās that are not adjacent to them. Eagles fly high in the sky; when they see a prey, they reach the ground, pick it up and continue to fly in the same direction as they did before. This characteristic represents the sūtrās that deviate from the topic of focus and return to the topic of focus. A pictorial representation of these four characteristics is given in Figure 7.1.

Figure 7.1 sūtra Characteristics
It can be observed that the sūtra characteristics can be compared with the saṅgatis, RST discourse relations, and RST based schema, discussed in the earlier chapters.

Table 7.1 sūtra characteristics- A comparison

<table>
<thead>
<tr>
<th>Sūtra Characteristic</th>
<th>Equivalent saṅgati /RST relation/RST schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aatrozhukku (River Flow)</td>
<td>anantara (sequence)/Sequence</td>
</tr>
<tr>
<td>Arima Nokku (Lion’s sight)</td>
<td>Contrast Schema</td>
</tr>
<tr>
<td>Parunthin veezhu (Eagle’s Flight)</td>
<td>Prathyavasthana(Re-instate)</td>
</tr>
</tbody>
</table>

The characteristic, “Thavalaippaaichal (Frog’s jump)” is mostly observed in the RS-saṅgati tree, in which non-adjacent text units are linked. These observations have led the proposed research to link the sūtra with the saṅgatis and RST based discourse structure.

The sūtra representation can be viewed as a document representation technique, and as an index to the UNL-RST-saṅgati discourse structure. The SemTRex framework uses the sūtra as an index and it identifies the sūtra for each CDUparagraph and builds an inverted index. The applications of sūtra are tested in two NLP applications, namely, summary generation and IR system, and compared with the existing equivalent RST-based NLP applications.

7.1 sūtra BASED TEXT REPRESENTATION

Figure 7.2 shows the architecture of the sūtra based text identifier. The input to the the sūtra-based indexer is a discourse structure which is built using the RST and/or saṅgati. The discourse structure needs to be represented as a tree, wherein each discourse tree represents a CDUparagraph.
The śūtra generation involves analysing the sub graphs of the discourse tree and picking up the most influential sub graph. In this framework, the śūtra is considered as a set of noun concepts and a set of discourse relations (RST/saṅgatis) which are associated with the noun concepts. The details of each of these steps are given below.

![Figure 7.2 Architecture of śūtra Based Indexer](image)

**Figure 7.2 Architecture of śūtra Based Indexer**

### 7.1.1 śūtra generation

śūtra generation is done through the following steps.
(a) Identification of all possible sub graphs that are connected directly to the root of the RS-saṅgati tree at each level, that lies between the root and the leaf of the tree.

(b) Weighting the sub graphs based on the following factors.

- The level at which the sub graphs are present. Leaf nodes are treated as level_0 and the number goes up as we go up the tree.

- The presence of influential discourse relations and saṃgatis in the sub graphs

(c) Generation of a sūtra from the highest weighted sub graph.

The sub graphs denote the NRS sequences of the RS-saṅgati tree. The sūtra generation aims at focussing on the influential sub graphs. The supposition is that the influential sub graphs are the ones which are coherently linked with the most influential node of the RS-saṅgati tree, which is the root of the tree. Hence, the sub graphs that are directly connected to the root node are extracted from each level for sūtra generation. The leaf level is omitted for this analysis.

The notion behind considering the level factor to assign weights to the sub graphs to form the sūtra is that, each level of the RS-saṅgati tree is constructed by popping it up from the level below. Consequently, the importance of the NRS sequences or the sub graphs increases as the level increases in the RS-saṅgati tree. Hence, a sub graph is ranked based on the level at which it is present. The weight factor is normalised by considering the number of levels present in the RS-saṅgati tree. The number of levels present in the RS-saṅgati tree depends on the size of the CDU for which the tree is constructed, and hence, the normalization is done by dividing the level of the tree by the number of levels present in the RS-saṅgati tree.
The influential discourse relations or saṅgatis refer to the ones that are specific to the domain, and are presumed to link important parts of the text. Furthermore, the queries posed by the user may be biased towards the domain, and hence, the sūtra which represents a document as an index needs to consider the domain specific discourse relations and saṅgatis. The influence score is assigned to each discourse relation and saṅgati, by calculating the probabilities of its occurrence in the corpora of 1000 Tamil language tourism domain specific text documents. For each sub graph, the normalized influence weight is calculated by adding the influence scores of the relations present in that sub graph, and dividing it by the number of relations in the sub graph.

Example 7.1 shows the Tamil text taken from the corpus, its transliteration along with the gloss, and the discourse structure for the text. Figure 7.3 shows the RS-saṅgati tree for the example text.

Example 7.1

English Transliteration:

Tirunelvēli māvaṭṭam tirunelvēli nakarai talaimaiyakanāka koṇū iyāṅkukiratu(Sen₁). Intiyāvīṇi paḷamaiyāṇa nakaraṅkalīl tirunelvēliyum
Tirunelveli city is the headquarters of Tirunelveli district (Sen₁).
Tirunelveli city is one of the oldest cities in India (Sen₂). 3000 years’ old Tirunelveli city is located on the river banks of Thamirabarani river (Sen₃). The discovery of an urn by researchers near Arichanallur area is evidence to show, that Tirunelveli is an ancient city (Sen₄). After investigating, archaeologists confirmed that Tirunelveli is a 2800 year old city (Sen₅).

**Figure 7.3 Discourse Structure for Example 7.1**
It can be observed that the example text constitutes a single CDU\textsubscript{paragraph}, and a sūtra needs to be generated for it.

The influence score of the discourse relations and the saṅgatis present in the discourse structure shown in Figure 7.3, is shown in Table 7.2. The complete list of influence scores for all the RST based discourse relations and saṅgatis are given in Appendix 3.

**Table 7.2 Influence scores for the Example**

<table>
<thead>
<tr>
<th>Discourse Relations/ saṅgatis</th>
<th>Influence Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>višeṣa</td>
<td>0.015</td>
</tr>
<tr>
<td>Upodghāta</td>
<td>0.005</td>
</tr>
<tr>
<td>Elaboration</td>
<td>0.2</td>
</tr>
<tr>
<td>prāsangika</td>
<td>0.075</td>
</tr>
<tr>
<td>Anantara</td>
<td>0.125</td>
</tr>
</tbody>
</table>

It can be observed that the sub graphs, namely, Sen\textsubscript{1}-Sen\textsubscript{2} (SubG1), Sen\textsubscript{2}-Sen\textsubscript{4} (SubG2) and Sen\textsubscript{1}-Sen\textsubscript{2}, Sen\textsubscript{4} (SubG3), are identified from the level l\textsubscript{1}, and the sub graph, Sen\textsubscript{1}-Sen\textsubscript{2} (SubG4) is identified from level l\textsubscript{2} of the RS-saṅgati tree. It should be noted that the sub graphs, Sen\textsubscript{1}-Sen\textsubscript{2} extracted twice at different levels, as the saṅgatis linking the nodes, Sen\textsubscript{1} and Sen\textsubscript{2} are different at these levels. Table 7.3 shows the weights assigned to these sub graphs.

The Total weight is calculated as the weighted sum of the two weight factors, level-based weight and influence weight, as shown in Equation 7.1.
Total weight = \( \alpha \times \text{(Level-based weight)} + (1-\alpha) \times \text{(Influence weight)} \) \hspace{1cm} (7.1)

**Table 7.3 Weight Calculation**

<table>
<thead>
<tr>
<th>Weight Factors</th>
<th>SubG1</th>
<th>SubG2</th>
<th>SubG3</th>
<th>SubG4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level Based weights</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.75</td>
</tr>
<tr>
<td>Influence weight</td>
<td>0.005</td>
<td>0.125</td>
<td>0.065</td>
<td>0.015</td>
</tr>
<tr>
<td>Total Weight</td>
<td>0.2525</td>
<td>0.3125</td>
<td>0.2812</td>
<td>0.3835</td>
</tr>
</tbody>
</table>

It can be observed that the sub graphs at the highest level of the tree will tend to get the highest score. To normalise this, the Level based weight is multiplied by a factor, “\( \alpha \)”, and the Influence weight is multiplied by a factor, “\((1-\alpha)\)”. With the limited experimentations, the value of \( \alpha \) is set as 0.5 and it has provided equal weight to the weight factors, level-based weight and influence weight. The value of \( \alpha \) can be varied according to the requirement. In this example, the sub graph SubG4 gets the higher score and is chosen for sūtra generation.

The sūtra for a CDU comprises a set of noun concepts, set of discourse relations and saṅgatis chosen from the top weighted sub graph. Since nouns convey the essence of a text, the noun concepts are chosen to represent the CDU. The set of discourse relations and saṅgatis which are present in the top weighted sub graph are chosen to form a sūtra along with the noun concepts. Also, the discourse relations/saṅgatis that link the top weighted sub graph with the rest of the sub graphs present in the CDU, are also added to the sūtra. Figure 7.4 shows the sūtra generation for a single text document, and Figure 7.5 shows the sūtra for the RS tree shown in Figure 7.3.
It can be observed that the text given in Example 7.1 expresses the specialities of Tirunelveli, and it is captured by the sütra through the noun concepts and the associated saṅgati, namely, viśeṣa. This sütra can be used as the semantic index to the CDU in NLP applications, such as the IR system, summary generation or QA systems. On the NLP application front, given a query, “Specialities of Tirunelveli” or “Specialities of India”, the document containing the text will be retrieved by mapping the cue word, “speciality” with the sangati, “viśeṣa,” along with the query words “Tirunelveli or India”.

7.1.2  sūtra as Index

For a given document, the sūtras are constructed for each CDU_paragraph. Figure 7.6 shows the inverted index representation of the sūtra. Each sūtra identified for a CDU_paragraph is stored along with the CDUs_paragraph that are linked to it at the document level RS tree. This unique quality of the sūtra aids in retaining the coherence of the discourse structure in the index too. Each sūtra is tagged with its respective CDU identifiers and the document identifier.

![Figure 7.6 Representation of the sūtra as an Inverted Index](image)

The next section discusses the usage of the sūtra in an IR system.

7.2  sūtra USAGE IN AN IR SYSTEM

The sūtra discussed in the previous section has been tested as an indexer on an IR system, using a corpora of 1000 Tamil language tourism domain specific text documents. The input to the sūtra based indexer is the CDU_paragraph level RS-saṅgati trees. The output is the inverted index representation, as shown in Figure 7.6. The IR system that uses the sūtra
based indexer has been tested using fifteen queries. The precision for the first ten documents (P@10) has been used as the parameter for evaluation.

A comparison has been done with an existing RST based indexing technique proposed by (Sahib & Ali sha 2006). The P@10 values of both the IR systems are shown in Figure 7.7.

![Figure 7.7 Performance Evaluation of sūtra using IR system](image)

**Figure 7.7 Performance Evaluation of sūtra using IR system**

Precision $\text{SemTRex Approach} = 0.76$

Precision $\text{Sahib & Ali Shah(2006) Approach} = 0.62$

Precision $\text{SemTRex(RST-sūtra) Approach} = 0.72$

It can be observed that the precision values achieved by the SemTRex approach are higher, than those of the Sahib & Ali Shah (2006) approach. Also, it can be observed that the SemTRex approach which uses only RST for sūtra construction, achieves a higher precision than Sahib & Ali Shah (2006) approach, but achieves a lower precision when compared to that.
of the SemTRex approach. The four main reasons behind such high performance of SemTRex are as follows.

- Presence of more than one discourse relation/saṅgatis present in each index.
- saṅgati relations
- Context coverage of the indices
- Search mechanism

The influence of these factors is explained using one of the queries used which is shown in Example 7.2. The P@10 values obtained for the SemTRex approach, Sahib & Ali Shah (2006) approach and SemTRex approach that uses RST based sūtra are 70%, 40% and 65% respectively.

**Example 7.2**

**Query:** பொள்ளைசு பள்ளியில் உள்ள ஒரு வேறுப்பு ஆய்க்குடி அங்கிலிக்கு திரும்பி

**Transliteration:** Pol|ăcci' arukē u|a' ājiyāru 'anaiyin cirappu

**Translation:** The speciality of the Aazhiyaaru dam which is near Pollachi.

The noun concepts present in the query are, “Pollachi (iof>city)”, “Aazhiyaaru Anai (iof>dam)”. The cue word, "sirappu (Speciality)" present in the query can be mapped with the discourse relation, "Elaboration", and the saṅgati, "viśeṣa.

The factor, “*Presence of more than one discourse relation/saṅgatis present in each index*” denotes the sūtra representation, which is a set of noun concepts and a set of discourse relations and or saṅgatis. With respect to Example 7.2, the discourse relation, "Elaboration", and the saṅgati, "viśeṣa” are used to find the relevant documents in the proposed approach, whereas, in
the Sahib & Ali Shah (2006) sūtra approach only one discourse relation, namely, “Elaboration” is used. Using more discourse relations will result in the retrieval of more relevant documents.

The usage of saṅgatis in addition to the RST based discourse relations used for discourse structure construction is also a major reason for the higher precision. Since, saṅgatis are more capable of capturing the coherence between the complex sentences, it stands as one of the reasons for the higher-precision values. This can be observed from the results of RST based sūtra approach when compared to that of the SemTRex approach.

The factor, “context coverage” indicates the text representative quality of the index. In the SemTRex approach, an index is a representative of a CDU which is essentially a paragraph, whereas, in the Sahib & Ali Shah (2006) approach, the index represents a clause or a sentence. With respect to the query given in Example 7.2, the indices, “Pollachi (iof>city)”, “Aazhiyaaru Anai(iof>dam)” while representing a CDU paragraph lead to the retrieval of semantically closer documents, than when the indices represent a clause or a sentence.

The factor, “search mechanism” denotes how the query is matched with the index. In the SemTRex approach, the search is done by using various combinations of the noun concepts-discourse relations. In the Sahib & Ali Shah (2006) approach, the NRS sequences that contain the discourse relation, “Elaboration” are first retrieved, and the NRS sequences containing the various combinations of the query terms are used for document retrieval. Since the methodology of matching the query terms and the index terms is not stated in the Sahib & Ali Shah (2006) approach, various combinations of all the query terms have been used for retrieving the results. The query combinations used by both the approaches are shown in Table 7.4.
Table 7.4 Comparison in terms of Query Formation

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pollachi (iof&gt;city)+ Aazhiyaaru Anai(iof&gt;dam)+ Êlaboration+ višeša</td>
<td>Pollachyil + Aazhiyaaru Anai+ulla</td>
</tr>
<tr>
<td>Pollachi (iof&gt;city+ Aazhiyaaru Anai(iof&gt;dam)+ Êlaboration</td>
<td>Pollachyil+ Aazhiyaaru Anai</td>
</tr>
<tr>
<td>Pollachi (iof&gt;city+ Aazhiyaaru Anai(iof&gt;dam)+ višeša</td>
<td>Pollachyil+ ulla</td>
</tr>
<tr>
<td>Pollachi (iof&gt;city)+ Êlaboration</td>
<td>Aazhiyaaru Anai+ ulla</td>
</tr>
<tr>
<td>Pollachi (iof&gt;city)+ višeša</td>
<td>Pollachyil</td>
</tr>
<tr>
<td>Aazhiyaaru Anai(iof&gt;dam)+ Êlaboration</td>
<td>Aazhiyaaru Anai</td>
</tr>
<tr>
<td>Aazhiyaaru Anai(iof&gt;dam)+ višeša</td>
<td>Ulla</td>
</tr>
</tbody>
</table>

It can be observed that the query contains “noun concepts (UWs)” in the SemTRex approach, whereas the query contains “terms” in the Sahib & Ali Shah (2006) approach. The chance of getting a relevant document when using concepts is always higher than when using terms. Moreover, it is observed that using only noun concepts avoids the retrieval of irrelevant documents. In the Sahib & Ali Shah (2006) approach, the search is done in two stages, which involve retrieving the NRS sequences from the documents using only the discourse relation, and then, picking up the NRS sequences that match with the query terms. This results in reducing the system performance in terms of speed.

It was observed from the corpus that the discourse relation, "Elaboration" is predominantly present, and hence, despite the presence or absence of the cue term in the query, the "Elaboration" discourse relation is
used as a default discourse relation, and is used along with the query terms for retrieval. By using the saṅgati, it can also be observed that the proposed indexer is scalable to handle other types of discourse relations as well. The indices built by the sūtra based indexer emulate the characteristics of the sūtra, and aid in retaining the semantics and coherence of the discourse structure in the indices. The sūtra based indexing technique differs from the existing RST based techniques in the way it identifies the indices, stores them and uses them for retrieval.

7.3 sūtra BASED SUMMARY GENERATION

This section discusses the summary generation mechanism that uses the sūtra based indexer. The summary generator has been designed in order to use the advantages of the RS-Saṅgati discourse structure to the fullest. Given an NL query as input, the sūtra based summary generation mechanism gives a query focused summary as the output. The presence of an inverted index in the sūtra based indexer makes it possible to generate a single document summary or a multi-document summary. The existing RST based summary generation approaches generate the summary, based on the cue phrases present in the query. The sūtra based summary generation technique generates summaries even for queries that do not contain sufficient cue phrases.

7.3.1 Summary Generation Mechanism

The architecture of the summary generation system is shown in Figure 7.8. In the IR system discussed in the previous section, the query is matched with the sūtra index, and the documents pertinent to the query are retrieved. In the case of summary generation, instead of retrieving the documents, the CDUs are retrieved, thereby forming a summary.
Since the noun concepts that constitute the sūtra are UWs, the nouns present in the NL query are converted to UWs. If the cue phrases are explicit in the query, then they are mapped with the discourse relations and saṅgatis; else, a default domain dependent discourse relation or saṅgati is used for the CDU retrieval. Then, these sets of discourse relations and saṅgatis and the noun concepts present in the query are matched with the sūtra index and the associated CDUs are retrieved to form a summary. As discussed previously, the sūtra based indexer stores each CDU, along with the CDUs that are semantically related through the discourse relations in the same document. Let the main CDU be denoted as CDU_{main} and the related CDUs as CDU_{linked}. Then the summary is comprised of CDUs_{main} and CDUs_{linked}. Since the sūtra indexer stores each index as an inverted index, a multi document summary can also be generated. Since the summary is built on top of the UNL-RST-saṅgati framework, the summary is a graph that comprises the components of this framework such as, UNL relations, discourse relations and saṅgatis. The NL summary can be obtained by removing the discourse relations and saṅgatis leaving behind only the nuclei and satellites. The nucleus and satellites are, in turn, UNL graphs which can be deconverted by a deconverter framework.

**Figure 7.8 Architecture of summary generation system**

The summary generation algorithm is as follows.
7.3.1.1 Summary Generation Algorithm

Given an NL query, the sutra based summary generator matches the query concepts with the sutra based indices and the summary is generated using any of the following cases.

Case 1: If the query contains cue phrases, then it is mapped with the respective RST based discourse relations/saṅgatis and used along with the UWs present in the query, to retrieve CDUs\textsubscript{main} and CDUs\textsubscript{linked}. The sutras of the CDUs\textsubscript{linked} and the UWs present in the query are matched. The CDUs\textsubscript{linked} that match with the query and the CDUs\textsubscript{main} form the summary graph.

Case 2: If the query does not contain cue phrases, then the semantic constraint of the last UW is checked. If it contains “iof”, then the UWs present in the query along with the default RST based discourse relation, “Elaboration” and the “prasangika” saṅgati are used to retrieve the CDUs\textsubscript{main} and CDUs\textsubscript{linked}. Again the CDUs\textsubscript{linked} are included in the summary, based on the query-sutra relevance.

Case 3: If step 1 is false and if the last UW of the query does not contain, “iof”, then the CDUs\textsubscript{main} and CDUs\textsubscript{linked} are retrieved using the UWs and the default RST based discourse relation and saṅgati. The CDUs\textsubscript{linked} are included in the summary, if their sutra contains the UWs whose semantic constraint contains, “iof” and the last NL word of the query.

This summary generation algorithm leans towards generating summaries for the tourism domain specific documents. Each case of the algorithm is explained below with an example.
7.3.1.2 Explanation with Examples

Case 1: For the queries containing the cue phrases, the cue phrases are mapped to a discourse relation or a saigati, and the relevant CDUs are retrieved. Consider the query given in Example 7.3.

Example 7.3

Query: ஆழ்தியாரு வாய்ப்பு விளக்கம்

Transliteration: Ālyāṟu aṇaiyin cīrappu

Translation: Speciality of Aazhiyaaru Dam.

The cue phrase, “சிறாறு (speciality)” can be mapped with the saigati, “visēxa” and can be used along with the UWs present in the query for retrieving the CDUs, thereby forming a summary.

Case 2: The presence of “iof” in the last UW of the query denotes that the query is specific about an instance which may denote a city, river, etc. Consider the query given in Example 7.4. The last UW of the query is, “Manakulavinayagar temple (iof> temple)”, which is an instance of the super class, “Temple”. The CDUs whose sūrās contain the UWs along with the default RST based discourse relation and sangati will yield a relevant summary.

Example 7.4

Query: புதுவையில் மாணகுலவியகர் தெய்வேயியம்

Transliteration: Putuvaiyil ulla maṇakkulavināyakar ālayam

Translation: Manakulavinayagar temple at Pondichery.
Case 3 : The last case deals with the queries whose last UW does not contain, “iof”. This denotes the scenario in which the query demands a summary of a specific instance or instances, not mentioned but implied in the query. Consider Example 7.5

Example 7.5

Query: “தமிழகத்தின் கோவில்சாலை”.

English Transliteration: Tamil[ṉāṭি] kōvil nakaram

English Translation: Temple city of Tamil Nadu

The summary is expected to contain information about the city, “கஞ்சிபுரம்” (Kanchipuram), which is the Temple city of Tamil Nadu, and is not present in the query.

The sūtra based summary generator first retrieves the CDUs whose sūtras contain the noun concepts, Tamil Nadu (iof>state) and (Temple City). Since there is no cue phrase present in the query, the default discourse relation, “Elaboration” and the sārga, “prasangika” are used along with the noun concepts for the CDU retrieval. All the CDUs main are included in the summary. Since the term, “Kanchipuram” is not present in the query, it may or may not be present in the CDUs main. Since the CDU main is the nucleus and CDUs linked that are linked to it are satellites, the intuition is that the CDUs linked may contain the elaborated content of CDUs main. Hence, there is a chance of the presence of the term “Kanchipuram” in any of the sūtras of the CDUs linked. This is checked by searching for a noun concept with a semantic constraint “iof>city”, which denotes an instance of a city in the sūtras of CDUs linked. The semantic constraint, “iof>city ” is chosen by finding the instance of (iof ) the last NL word of the query. Here, the last UW is, “கோவில்சாலை”(Temple City) and the last NL word is “சாலை”(city) and so the
semantic constraint, “iof>city” is used for searching the UW in the sūtra of the CDUs\textsubscript{linked} along with the other UWs, and the default RST based discourse relation and sangati identified from the query. The CDUs\textsubscript{linked} that satisfy this condition and the CDUs\textsubscript{main} together form the summary. It can be observed that the sūtras of the CDUs\textsubscript{main} and CDUs\textsubscript{linked} aid in picking up the closely relevant summary, even if the query contains insufficient details. Consider the queries given in Examples 7.6, 7.7 and 7.8 which can be handled in a similar manner.

**Example: 7.6**

**Query:** தமிழ்நாட்டின் பாலையாய்க்கான் கோவில்கள்

**Transliteration:** Tamilnāṭṭin paḷaimaiyāna kōvilkaḷ

**Translation:** Ancient Temples of Tamil Nadu

**Example: 7.7**

**Query:** தமிழ்நாட்டின் குருவுல் தலங்கள்

**Transliteration:** Tamilnāṭṭin cuṟulāt talaṅkai

**Translation:** Tourist Spots of Tamil Nadu

**Example: 7.8**

**Query:** இந்தியாவின் புகைமிக்க நதிகள்

**Transliteration:** Intiyāviṇa pukaṃikka natikaḷ

**Translation:** Famous Rivers of India

It can be observed that the above queries demand a summary, which should contain information about various places like, temples, tourist spots and rivers, whereas, the summary for the query, Temple city of Tamil
Nadu” should deal only with a single place. The proposed sūtra based summary generation mechanism generates summaries for both the cases, as the sūtra based indexer is capable of retrieving different CDUs that are semantically woven together based on some common criteria. In other words, the coherence captured in the RST-saṅgati discourse structure is retained in its indices too.

### 7.3.2 Evaluation

The sūtra based summary generation system has been evaluated, using the Forum for Information Retrieval Evaluation (FIRE) guidelines (FIRE, 2010). The FIRE evaluation task comprises 2,000,000 documents, a set of 15 queries and a narration that states the necessary information to be present in each document retrieved by the IR system. The proposed summary generation approach uses these set of documents for constructing the summary generation system. The queries are used to generate the summaries, and the quality of the summaries is analysed manually by comparing it with the “narration” criteria given for each query in the FIRE task.

A comparison has also been made with the existing RST based summary generation proposed by Bosma (2004). Bosma methodology is implemented using the RST tagged discourse corpus built by the UNL-RST discourse parser. So, the documents set are the same and the comparison is done in terms of the methodology for summary generation. The summaries are evaluated through human judgement by three people who are experts in this domain. The summaries are assigned scores 1, 2 and 3 which indicate low, medium and high query-summary relevance respectively. The sūtra based summary generator has achieved an average score of 2.3 and the summary generation system proposed by Bosma (2004) has achieved an average score of 1.75.
The summary generation mechanism proposed by Bosma (2004) generates a query-focused summary using the RST based discourse structure which is represented as a graph. Given a query, a QA system locates an answer sentence which is represented as a node in the discourse graph. Using the answer sentence or node as the entry point, a summary is generated by appending the nodes that are semantically closer to the answer node in the discourse structure. This is measured by using certain weight measures. The query processing and the query-index match done by the QA system, have not been discussed by Bosma (2004). The weights of each node or vertex (represented by an EDU) and an edge (represented by a discourse relation) are determined as shown in Equations 7.2 and 7.3.

\[
\text{weight(edge)} = a + b * \left( \frac{1}{\text{sentences(satellite(r))}} \right)
\]  

(7.2)

where, \( e \) is the edge that was created for the discourse relation \( r \), where \( \text{satellite(r)} \) is the satellite of \( r \), and \( \text{sentences(s)} \) is defined as the number of sentences of a span \( s \), \( a \) is the basic weight, and \( b \) is a constant factor of the ‘satellite size’ component of the edge weight;

\[
\text{weight(vertex)} = c * \left( \frac{1}{\text{words(s)}} \right)
\]  

(7.3)

where, \( v \) is the vertex that was created for the sentence \( s \), where, \( \text{words(s)} \) is the number of words in \( s \), and \( c \) is a constant.

It can be observed that the vertices and the edges that are appended to the node pointed by the query to form a summary do not have any direct link with the query, and there may be possibilities where the summary could deviate from the query. Consider the query and the summary output of both the sūtra based summary generation mechanism, and Bosma’s approach shown in Example 7.9.
Note: Though the output summary is not deconverted, the NL sentences corresponding to the retrieved summary are shown here, for better understanding.

Example 7.9

Query: “சோப்பாம் மறுசெய்தல் மறுசெய்தல்”

Transliteration: Siv irvin marangam

Translation: Death of Steve Irvin

sūtra based summary

Translation: Death of Steve Irvin (03.09.2006) and the incident of Black Sequoia and the incident of the India Fire (CDU id: 365; Document id: 271)

English Transliteration

Tikilai ēpaṭuttum mutalai końci kulāvum avarūaiya kācikalai ciṟu kulantaikal maṭṭumìṇi tolaikkāṭiyil aṇāvairiy kavanattaiyum īrtta āstirēliyāvai cērnta sīv irvin īṟū (03.09.2006) Tannērīl āpattāna vilankiṇaṅkaḷ paṟṟiya oru tākumentari ēṭukkumpoḷṭu sēṟṅk rēs ēṭappāṭum oruvakai mēniṇattin tākkutalukku ullāki irantar
English Translation:

Steve Irwin, an Australian who attracted not only kids but everybody on TV through his brave gestures by playing with crocodiles dies today on 3.9.2006. While shooting a documentary on a dangerous fish variety named, string rays, he died due to the attack of the fish.

The crocodile hunter Steve Irwin pulled out the poisonous tail of the fish which tried to attack him; became unconscious and then he died.

Bosma’s Approach

Transliteration
On September 4, 2006, while taking a documentary about dangerous species, he died due to the attack of a fish variety called, string rays.

His wife also takes care of crocodiles. Irwin’s father and mother taught him about taking care of crocodiles.

It can be observed that both the CDUs retrieved by the sutra based summary generation approach are related to the query, whereas in Bosma’s approach, the second EDU is related to the person, “Steve Irwin” specified in the query, but not completely related to the entire query. While looking from the query focussed summary generation perspective, the proposed approach generates a summary by retrieving the texts through the index from different documents that are completely related to the query. In Bosma’s approach, the summary is generated by retrieving a text through the index and adding the texts that are related to the text which may or may not be related to the query. This leads to deviations in the sentences constituting the summary generated by Bosma’s approach. The other reasons for better performance by the sutra based summary generation mechanism, are the use of saṅgatis and retrieval of CDUs for summary generation rather than EDUs, which were discussed in section 7.2.
7.4 SUMMARY

In this chapter, śūtra based concise representation of a discourse structure has been presented. This has been used as an indexer and tested on two NLP applications, namely, IR and summary generation systems. Both the NLP applications have been compared with the existing equivalent RST based applications. Generating cross lingual and multi lingual summary generation is possible with the proposed approach. The śūtra representation is obtained using only two factors, which need to be enhanced. The factor, “α” needs to examined by setting different values to it.

The overhead involved in the proposed approach is the requirement of a UNL framework, as the UNL-RST-saṅgati discourse parser requires the input text documents to be enconverted to UNL. Since the output of the śūtra based summary generator is a language independent UNL-RST-saṅgati graph, the NL summary can be decoded to any target language by using an appropriate UNL deconverter and RS-saṅgati decoders. Though the proposed research does not involve the generation of cross lingual and multi lingual summaries, the current śūtra based summary generator can be extended to generate such summaries in the future. Apart from the NLP applications designed, the śūtra based concise representation is also suitable for the QA system.