Documentation for SummBank 1.0

Released by the Linguistic Data Consortium

Corpus by:

Dragomir Radev, Simone Teufel, Horacio Saggion, Wai Lam,
John Blitzer, Arda Çelebi, Elliott Drabek, Danyu Liu, and Hong Qi

CD and documentation by:

Tim Allison, tballiso@umich.edu, University of Michigan
Dragomir Radev, radev@umich.edu, University of Michigan

http://www.clsp.jhu.edu/ws2001/groups/asmd/
http://www.summarization.com/summbank/
http://www.summarization.com/mead/

October 14, 2003

Contents

1 JHU 2001 Introduction and the SummBank corpus

The goal of SummBank is to gather together a corpus of original documents and summaries which can be used as gold standards by the document summarization community. Eventually, we envision SummBank including original documents, human written summaries, and machine created summaries from a broad range of genres and applications.

SummBank 1.0 contains the data created for the Summer 2001 Johns Hopkins Workshop which focused on text summarization in a cross-lingual information retrieval framework. While this documentation assumes some familiarity with the final report produced from that workshop, it repeats some of the information which is contained therein for the sake of the users of SummBank 1.0. We have included a pdf of the final report in the documentation directory of SummBank 1.0, but it should also be available at `http://www.clsp.jhu.edu/ws2001/groups/asmd/`.

In the summer of 2001, researchers gathered at Johns Hopkins to study cross-lingual text summarization. The LDC (Linguistic Data Consortium) supplied this group with a corpus of 18,147 bilingual document pairs covering 1997-2000 from the Hong Kong News Parallel Text corpus (corpus number LDC2000T46). These document pairs were used in single-document summarization experiments. However, the LDC also created 40 clusters of news articles from this corpus for use in multi-document summarization experiments. The LDC had annotators create 40 queries ("Y2K readiness", "Flower shows", etc.) which they used in their own information retrieval engine to select candidate sets of documents. Human judges then selected the ten most relevant documents for each cluster.

In addition to providing the raw documents and clusters of documents, the LDC had human annotators judge each sentence's relevance to its cluster's query. The judges used a scale of 0 (not relevant at all) to 10 (very relevant). There were a total of five human judges on this project, and each sentence was judged by 3 judges. These scores were used to judge a summarizer's performance and to create automatic extractive summaries. Finally, the judges were also asked to write summaries at various compression rates for each cluster.

This CD-ROM contains 40 news clusters in English and Chinese, 360 multi-document, human-written summaries, and nearly 2 million single document and multi-document extracts created by automatic and manual methods.

To get full use of this data (especially in evaluation), it will help to have the Hong Kong News Parallel Text corpus (LDC2000T46) available as well. This is available through the LDC `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2000T46`. As mentioned, the corpus contains 18,147 aligned bilingual (Cantonese and English) article pairs from the Information Services Department of the Hong Kong Special Administrative Region of the People's Republic of China. The articles were originally released between 1997 and 2000. The LDC gathered and aligned the articles.

2 Input: Document clusters and docsents

Before turning to the arrangement of the data in SummBank 1.0, it will be useful to give examples of the different data types which were used in the workshop. In this section, we present the objects used for input, and in the next section we present the basic output objects.

Group_2	Meetings with foreign leaders
Group_46	Improving Employment Opportunities
Group_54	Illegal immigrants
Group_60	Customs staff doing good job.
Group_61	Permits for charitable fundraising
Group_62	Y2K readiness
Group_447	Housing (Amendment) Bill Brings Assorted Improvements
Group_827	Health education for youngsters
Group_885	Customs combats contraband/dutiable cigarette operations
Group_1018	Flower Shows

Figure 1: Queries in training clusters

Group_112	Autumn and sports carnivals
Group_125	Narcotics rehabilitation
Group_199	Intellectual Property Rights
Group_241	Fire safety, building management concerns
Group_323	Battle against piracy: Ordinance requires licences for manufacture of optical discs; strict crack-down on piracy operations
Group_398	Flu results in Health Controls
Group_551	Natural disaster victims aided
Group_883	Public health concerns cause food-buisness closings
Group_1014	Traffic Safety Enforcement
Group_1197	Museums: exhibits/hours

Figure 2: Queries in devtesting clusters

2.1 Clusters

The LDC used a clustering algorithm to gather together documents which were on a similar topic. Human annotators then selected for each cluster the top 10 documents which were most relevant to the query. During the workshop, we had 10 “training” clusters and 10 “devtesting” clusters. At the end of the workshop, the LDC also provided us with 20 “testing” clusters. Each cluster contains the 10 documents in English and in Chinese along with other information. We present the queries for each of the subsets of clusters in figures 1, 2, and 3.

Group_64	Chief Executive’s Diplomatic Duties
Group_85	Public transportation operations
Group_133	Endangered Species Controls
Group_202	Air/water polution
Group_203	Housing/care for elderly
Group_265	Cholera is a reminder: cases cause Dept. of Health to stress importance of public and personal hygiene to guard against disease
Group_291	voter/elector registration and info
Group_310	Post office info (souvenirs, philately, etc.)
Group_522	Relationship-building with Mainland
Group_610	Immigration concerns
Group_661	Review of Academic System
Group_812	Mandatory Provident Fund
Group_814	Customs seizes smuggled goods
Group_841	Rewards for info on violent crimes
Group_881	Customs seize counterfeit goods
Group_971	Customs’ involvement in drug conrol.
Group_994	E-commerce and Information/Technology
Group_1190	Spread of Enterovirus 71 monitored
Group_1332	Anti-piracy operations by Customs
Group_1523	Youth Athelete Training

Figure 3: Queries in testing clusters

2.2 Docsents

We converted the text supplied by the LDC into docsent objects, an example of which is shown in figure 4. We used these objects as input into our summarizers.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE DOCSENT SYSTEM "docsent.dtd" >
<DOCSENT DID='D-19991028_005.e' DOCNO='14041' LANG='ENG'
   CORR-DOC='D-19991028_004.c'>
<BODY>
<HEADLINE><S PAR="1" RSNT="1" SNO="1"> Eastern Traffic Day </S></HEADLINE>
<TEXT>

<S PAR='2' RSNT='1' SNO='2'>Eastern District Police officers
have issued a total of 107 Fixed Penalty Tickets (FPT) and 34
summonses against road users violating traffic regulations during
an operation conducted between 8 am and 4 pm today (Thursday).</S>

<S PAR='3' RSNT='1' SNO='3'>Officers from Eastern District,
North Point Division, Shau Kei Wan Division, Chai Wan Division
and Eastern District Traffic Team were deployed to enforce traffic
regulations in traffic black spots of the district.</S>

<S PAR='4' RSNT='1' SNO='4'>A total of 95 FPTs were issued
against illegal parking and 12 for other traffic offences.</S>
</BODY>
</DOCSENT>
```

Figure 4: An example of a docsent object

Within the metadata section of the docsent, we include the name of the document (DID), the document's original number (DOCNO), and the language (ENG or CHIN). Some docsents include information about the corresponding document in the other language, as this one does. In the description of the docsent, it points to D-19991028_004.c as the corresponding Chinese document. Note that the numbers are not the same.

For each sentence, we also tried to maintain paragraph divisions. RSNT identifies a sentence's number within its paragraph, and SNO identifies a sentence's number in the whole document.

3 Output: Summaries and extracts

In our framework, the output from a summarizer can come in two forms, one is a summary, which contains the text of those sentences chosen by a summarizer, and the other form is an extract, which contains a list of the sentences chosen by a summarizer. The following is an example of an extract. The equivalent summary would include the text of those sentences.

Through the course of this project, we created summaries and extracts in a number of different ways. These differed in method used (human, machine), nature of source document/documents (single-document and multi-document), extraction amount (5%, 10%, 5 sentences, 10 sentences, 50 words, 100 words), and type of extraction amount (word-based or sentence-based). For the sake of avoiding redundancy, we have included on this distribution only the extracts we created during the workshop. See below under the section called "Tools" for how to convert an extract to a summary.

Finally before moving on, we should point out that a few extracts include the text of the sentences that were chosen. We believe that this type of extract is limited to those extracts created by Chin-Yew Lin's summarizer. We present an example of this in 6.

It will be useful to record here what summaries and extracts we include in SummBank:

- Human manual extracts, single document and multi-document
  - Using the relevance scores supplied by the LDC annotators, we created a script to select the appropriate number of sentences with the n highest relevance scores. These were created at a number of different compression ratios.
- Human manual summaries

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EXTRACT SYSTEM "/export/ws01summ/dtd/extract.dtd">

<EXTRACT QID="Group_1014" COMPRESSION="200"
SYSTEM="/home/ws01/blitzer/newMEAD/driver/hkmead.pl Position 1
Centroid 1 Firstsim 0 Length 9 LANG="ENG" SENTS_TOTAL="80"
WORDS_TOTAL="1180">

<S ORDER="1" DID="D-19981230_005.e" SNO="4" />
<S ORDER="2" DID="D-19990312_004.e" SNO="4" />
<S ORDER="3" DID="D-19990428_010.e" SNO="4" />
<S ORDER="4" DID="D-19990526_010.e" SNO="3" />
<S ORDER="5" DID="D-19990526_010.e" SNO="4" />
<S ORDER="6" DID="D-19990929_012.e" SNO="4" />
<S ORDER="7" DID="D-20000127_017.e" SNO="4" />
<S ORDER="8" DID="D-20000127_017.e" SNO="6" />
<S ORDER="9" DID="D-20000218_011.e" SNO="2" />
<S ORDER="10" DID="D-20000218_011.e" SNO="4" />
</EXTRACT>
```

Figure 5: An example of an extract object

```
<?xml version="1.0"?>
<!DOCTYPE EXTRACT SYSTEM "extract.dtd">
<EXTRACT QID="none" COMPRESSION="5S" SYSTEM="SM1998" RUN="08042001"
LANG="ENG" >
<S ORDER="1" DID="../english/199707/19970701_001.e" SNO="1">
A solemn , historic ceremony has marked the resumption of the exercise
of sovereignty over Hong Kong by the People ’s Republic of China.</S>
</EXTRACT>
```

Figure 6: An example a fuller extract object

  - Three LDC annotators each wrote summaries of the clusters at 3 different compression rates.

- Automatic extracts
  - LEAD
  - Greg Silber's LEXCHAIN
  - RANDOM
  - MEAD
  - WEBSUMM
  - CYL (Chin-Yew Lin's summarizer)

4 The layout of the data

In the top directory are the following subdirectories: data and documentation. The documentation directory contains all of the documentation as well as a subdirectory for the dtds. In the data directory live `automatic`, `clusters`, `manual`, and `tools`. Figure 7 offers a rough schematic of the layout of the data. And we will take up each in the following.

4.1 Clusters

The clusters are arranged so that they can easily be used by MEAD, the single- and multi-document extractive summarization system, which was developed in the workshop. For a description of how MEAD works, see its documentation at `http://www.summarization.com/mead`, section 4.2 of the workshop's final report `http://www.clsp.jhu.edu/ws2001/groups/asmd/`, or DUC 2001, `http://www-nlpir.nist.gov/projects/duc/pubs/2001papers/umich.pdf`.

```
data\
   automatic\
      docjudges\              all docjudges
      features\               sentence features computed by the original version of MEAD
      summaries\              automatic extracts, created by MEAD, CYL, LEXCHAIN, etc.

   clusters\
      alignments\             sentence by sentence English + Chinese alignment information
      chinese\                Chinese clusters
      english\                English clusters
      single-doc-cluster-files\ cluster files for all single documents

   manual\
      manual_extracts\        extracts based on human judgements of sentence relevance
      manual_summaries\       multi-document summaries written by humans

   tools\
      MEAD 3.07
      dtder.pl                script for setting dtds locally
      mf.pl                   script for creating single document cluster files
      formatting\             contains scripts for converting documents to
                              MEAD-readable clusters

documentation\
   jhufinalreport          Final report from the workshop
   SummBank 1.0 doc.       documentation for this data
   dtd\                    all dtds
```

Figure 7: Data layout

MEAD is only needed for certain types of access to SummBank, such as evaluating summaries and producing full text summaries from extracts, but we thought it would be useful to arrange the clusters in a MEAD-friendly way in case people were interested in using MEAD on the clusters in future experiments. Nevertheless, SummBank does not require MEAD.

4.1.1 The arrangement in the `clusters` directory

In the `clusters` folder, there are 4 subdirectories: `english`, `chinese`, `alignments` and `single-doc-cluster-files`.

In the `english` and `chinese` subdirectories, there are the following subdirectories: `training`, `devtesting` and `testing`. Beneath each of these, we have the news clusters.

4.1.2 The arrangement of data within a cluster

In our framework, each cluster of documents has (at the minimum) a `cluster` file, which contains the names of the files within a cluster, and a `docsent` folder, which contains the files in docsent format which are named in the `cluster` file.

In addition to these minimum requirements, each cluster can have several other types of information, and I will turn to these in the following.

4.1.3 The minimum

Within each cluster, there is at the minimum:

- cluster file: contains the list of documents in the cluster
- docsent folder: contains the articles in docsent format for the cluster

4.1.4 Data for some English documents

For the English documents we also have the following:

1. sentjudge file: contains the relevance judgements on each sentence in the cluster to the overall cluster
2. ldc_orig directory: contains the original, raw data from the LDC,
   - stg files contain the text as we originally segmented it
   - doclabel files contain the binary relevance judgements of the documents to the cluster's query
   - sentlabel files contain the judges' relevance judgements for each sentence. An example line is the following:

```
Group_1014      19990428_010.e  4.2     judge4  4
```

This means for cluster 1014, document 19990428_010.e, paragraph 4, sentence #2 in that paragraph, judge4 gave a relevance score of 4.

Beyond this information, there are some differences in the type and number of files we have for training, devtesting and testing clusters.

4.1.5 Data for English training clusters

For training clusters, we have:

1. docjudge files, which are xml formatted equivalents to the raw doclabels files. These files derive from the LDC's automatic binary (relevant/not-relevant) clustering algorithm. As we will discuss below, we also use docjudge files to record the rank of each document and summary to a given query using SMART. The LDC-derived docjudge files also contain document alignment information:

```
<D DID="D-19970916_002.e" SCORE="N" CORR-DOC="19970916_002.c"/>
<D DID="D-19971009_014.e" SCORE="N" CORR-DOC="19971009_014.c"/>
<D DID="D-19971023_023.e" SCORE="N" CORR-DOC="19971023_023.c"/>
<D DID="D-19971026_004.e" SCORE="N" CORR-DOC="19971026_004.c"/>
<D DID="D-19971116_001.e" SCORE="N" CORR-DOC="19971116_001.c"/>
<D DID="D-19971214_006.e" SCORE="B" CORR-DOC="19971214_009.c"/>
```

Figure 8: An example of the LDC's docjudge object

2. query files, these xml files contain the query from which LDC created the cluster.
3. subsume files, these are the raw files which record judge's evaluations of which sentences subsume which sentences. We actually have subsume files for the training clusters and for clusters 447 and 827.

4.1.6 Data for English devtesting and testing clusters

For the devtesting clusters, in addition to the basic files, we have only the query files. For the testing clusters, we have only the basic files outlined above.

4.1.7 Data specific to Chinese clusters

The information we have for the Chinese clusters differs somewhat from what we have for the English clusters. First, there is a cluster file for each cluster because sometimes corresponding Chinese and English documents do not have the same name.

For the Chinese training clusters, we also include the following:

1. manual query file, which contains the manual translation of the English query.
2. manual.seg.query, which contains the segmented version of the manual translation of the English query

For the Chinese devtesting clusters, there is only the manual query file.

4.1.8 Alignment information

At the level of the `english` and `chinese` directories, there's also an `alignments` directory; under this we have put all the clusters at the same level. Under each cluster's directory, we have put the `align` files, which contain information on which English documents and sentences correspond to which Chinese documents and sentences.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SENTALIGN SYSTEM "sentalign.dtd">
<SENTALIGN ENG="19981230\_005.e" CHI="19981230\_005.c"
LANG="english-chinese">
<SENT ORDER="1" EDID="D-19981230\_005.e" ESNO="1" CDID="D-19981230\_005.c" CSNO="1" />
<SENT ORDER="2" EDID="D-19981230\_005.e" ESNO="2" CDID="D-19981230\_005.c" CSNO="2" />
<SENT ORDER="3" EDID="D-19981230\_005.e" ESNO="3" CDID="D-19981230\_005.c" CSNO="3" />
<SENT ORDER="4" EDID="D-19981230\_005.e" ESNO="4" CDID="D-19981230\_005.c" CSNO="4" />
<SENT ORDER="5" EDID="D-19981230\_005.e" ESNO="5" CDID="D-19981230\_005.c" CSNO="5" />
<SENT ORDER="6" EDID="D-19981230\_005.e" ESNO="6" CDID="D-19981230\_005.c" CSNO="6" />
<SENT ORDER="7" EDID="D-19981230\_005.e" ESNO="7" CDID="D-19981230\_005.c" CSNO="7" />
<SENT ORDER="8" EDID="D-19981230\_005.e" ESNO="8" CDID="D-19981230\_005.c" CSNO="7" />
</SENTALIGN>
```

Figure 9: An example of a sentalign object

As one can see, there is normally good correspondence between English and Chinese sentences. However, as in the last sentence, occasionally, there is a mismatch. In this case, two English sentences correspond to one Chinese sentence.

4.1.9 Single document cluster files

At the same level as the `english`, `chinese` and `alignments` directory, we have also put the `single-document-cluster-files` directory, which contains the cluster files for each individual file. Although the name of this directory is a bit of an oxymoron, these files are meant to be used for single document summaries. We also included in the `single-document-cluster-files` directory a script called `mf.pl`, which was written to create these single-document cluster files.

4.2 Manual summaries

If we go back to the top `data` directory, below that is a `manual` directory. Underneath this, we have put all of the summaries which were created by the LDC annotators. Below manual are manual_extracts and manual_summaries. Under manual_extracts, there are multi_document and single_document. The directory naming scheme under multi_ and single_documents is the same, so the following explanation is intended to cover both. Under multi_document are folders for each of the training clusters (only), and under these are the manual multi-document extract folders. Under these is the extract folder. We used to include a "document folder" at the same level which contained the text of the summary. However, for the sake of space, as mentioned, we removed the summaries from this distribution. In the extract folder, at long last, however is the appropriate extract (so the final path looks like, e.g.
manual/manual_extracts/multi_document/1014/M-E-C_1014-20S-LDC_J001/extract/Group1014.e.extract).

Both the single_ and multi_document manual extracts were created by a script which sorted the sentences by their relevance judgements, and then selected the top n sentences. The way to read the folder names is the following:

`M-E-C_1014-20S-LDC_J001`
Multidocument-English-Group_1014-20% of the sentences summary-LDC_judge001

There are 2 important pieces of information in this folder name:

1. the -20S-, which means a summary which has 20% of the number of sentences of the total cluster. Another option for this position is -20W- which means that the extract creates a summary which contains 20% of the words of the original cluster or document. The last option is -050-, which means that the summary was

created to have roughly 50 words. (see the final report, pp. 24-25, for the details of how they calculated this).

2. the _J001 identifies which judge's scores the extract is based on. _ALLJ means that the judges' scores were combined. These judge numbers do not correspond to the judge numbers in the sentjudge files. We have also included the lemmatized versions.

Underneath manual, if we go to the automatic_summaries directory, we find lemmatized and regular. Under regular, we have all of the manually written summaries. For each cluster three judges were asked to write a summary at 3 lengths, 50 words, 100 words and 200 words. The judge #'s on these filenames do correspond with the judges in the sentjudge files.

4.3 Automatic data

4.3.1 Features

In this directory, we have put all of the features created by MEAD for each single document summary. In creating summaries, MEAD creates a features directory, which contains three files, the cosine similarity of a sentence with its document or cluster, the sentence length and the sentence position in the document. We archived the feature files which were created by MEAD for single document summarization in the FEATURES directory. During the workshop we used an older version of MEAD than the current release version (3.07), and we wanted to archive the features that were created with the older version of MEAD.

4.3.2 Automatic summaries

There were five summarizers that we used to create automatic summaries and extracts (ALG (alignment), CYL (Chin-Yew Lin's summarizer), MEAD, LEXCHAIN and WEBSUMM), and we also created two baseline summaries, random and lead. See section 4.3 of the JHU 2001 Final Report for a description of the different summarizers. All of the output from these methods is placed in the `automatic` directory. The file structure under each is nearly the same, so I will focus on MEAD as an example. The key thing is to understand the directory names:

```
M-GENERC-E-050-MEADORIG
S-Q00125-C-20S-MEADHONG
```

1. M means multidocument. Under this heading, we have included extracts made for the training and devtesting clusters. The alternate is S, which stands for single-document. Under these directories, we have summarized every document.

2. The GENERC means that the summarizer was run generically, without a query. MEAD also has a query capability, so that if in the second position there is a Q00112, for example, that means that we had MEAD create a query-based summary for each of the documents for cluster 112.

3. The third item identifies the language of the summaries or extracts.

4. The fourth item identifies the summary amount. 050, 100, 200, refers to an extract which will create a summary of roughly 50, 100 or 200 words. 20S means that the summaries contain 20% of the number of the original document's or cluster's sentences. 20W means that the summaries contain 20% of the number of the original document's or cluster's words.

5. The fifth item identifies the summarizer or the configuration of the summarizer.

Originally there were 10 or so configurations for MEAD. On the CD-ROM, we include MEADORIG, MEAD0002, MEADS002, MEAD0003, MEADS007, and MEADHONG. The basic configurations are explained in the following table.

As this shows, summaries identified with MEAD0002, for example, were created by MEAD with a Centroid weight of 1, a Position weight of 1, and an Overlap weight of 1.

The weights for MEADS002 summaries were calculated by a machine learning algorithm for generic summaries. The weights for MEADS007 were calculated in the same way for query-based summaries.

The weights for MEADHONG are as follows:

```
1 Centroid
2 Position
3 Overlap
4 Q-cosine
5 Q-overlap

MEAD0000 1-1-0-0-0          - same as MEADORIG
MEAD0001 1-1-0-0-0          - normalized
MEAD0002 1-1-1-0-0
MEAD0003 1-1-1-1-0
MEAD0004 1-1-1-1-1
MEAD0005 1-1-0-1-0
MEAD0006 from SVM - query-based
MEAD0007 from SVM - query-independent
```

Figure 10: The different configurations of MEAD

```
Position 1 Centroid 1 Firstsim 0 Query-Title-Sqrtcosine 1 Length 9
```

Finally, we have included some query-based extracts created by WEBSUMM and MEAD. However, we have only included multi-document extracts created by MEAD.

4.3.3 Docjudges

As discussed in the report from the workshop (section 3), one of the methods of evaluation we used is based on principles from information retrieval. We wanted to determine what effect a given summarizer would have on ranking a document's relevance to a given query. The intuition is that the ideal summarizer's summaries would be ranked in the same order as the original documents. And so, we have included all of the docjudges we created automatically with SMART.

The names for docjudges are similar to those for summaries.

```
S-GENERC-E-20S-MEAD0002-Q01197-M-SM002.docjudge
```

One difference is in the M or T which comes before the SM001 or SM002 towards the end of the name. T signifies translingual, and M signifies multilingual. Within these files, the query type differs as well. For example, in S-GENERC-C-40W-ALGCLEAD-Q01197-M-SM001.docjudge, the query type is simply -C (Chinese): `<DOC-JUDGE QID="Q-1197-C" SYSTEM="SMART" LANG="CHIN">`, whereas in S-GENERC-C-40W-ALGCLEAD-Q01197-T-SM001.docjudge, the query type is -CA (Chinese aligned):
`<DOC-JUDGE QID="Q-1197-CA" SYSTEM="SMART" LANG="CHIN">`.

The other difference is in the SM001 or SM002 towards the end of the filenames. These signify two different parameter settings of SMART.

The summarizers were run at ten different target summary lengths to produce a very large number of summaries, and we here present a figure of the various parameter settings that were used in the original data. The current distribution has nearly 2 million extracts. For a list of the runs compiled during the original data packing and storing, see the file named RESOURCES under the documentation subdirectory.

5 Tools

For information, updates and patches on all of these tools, please visit `http://www.summarization.com/mead` and `http://www.summarization.com/summbank`.

5.1 MEAD and MEAD add_ons

We have included the most recent version of MEAD (3.07) with this distribution. Please see the most recent documentation for how to run it. We are constantly adding new functionality to MEAD, so please visit the website: `http://www.summarization.com/mead`. As of this writing, we have recently released a new suite of addons to add greater functionality to MEAD 3.07.

	Lengths																					#dj
	05W	05S	10W	10S	20W	20S	30W	30S	40W	40S	50W	50S	60W	60S	70W	70S	80W	80S	90W	90S	FD	
E-FD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	40
E-LD	X	X	X	X	x	x	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-	440
E-RA	X	X	X	X	x	x	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-	440
E-MO	x	x	X	x	x	x	X	x	X	x	-	x	-	x	-	x	-	x	-	x	-	540
E-M2	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	20
E-M3	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8
E-S2	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8
E-WS	-	X	-	X	x	x	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-	160
E-WQ	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	10
E-LC	-	-	-	-	-	-	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	40
E-CY	-	X	-	X	-	x	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	120
E-AL	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	200
E-AR	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	200
E-AM	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	200
C-FD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	40
C-LD	X	X	X	X	x	x	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	240
C-RA	X	X	X	X	x	x	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	240
C-MO	X	x	X	x	x	x	X	x	X	x	-	-	-	-	-	-	-	-	-	-	-	320
C-M2	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	20
C-CY	-	X	-	X	-	x	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	120
C-AL	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	180
C-AR	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	200
C-AM	-	X	X	X	X	X	X	X		X	-	-	-	-	-	-	-	-	-	-	-	120
X-FD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	40
X-LD	X	X	X	X	x	x	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	240
X-RA	X	X	X	X	x	x	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	240
X-MO	X	x	X	x	x	x	X	x	X	x	-	-	-	-	-	-	-	-	-	-	-	320
X-M2	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	20
X-CY	-	X	-	X	-	x	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	120
X-AL	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	140
X-AR	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	160
X-AM	-	X	X	X	X	X	X	X	-	X	-	-	-	-	-	-	-	-	-	-	-	120

Figure 11: All runs performed (X = 20 clusters, x = 10 clusters). Language: E = English, C = Chinese, X = X-lingual; Summarizer: LD=LEAD, RA=RAND, WS=WEBS, WS=WEBS-query based, etc.; S = sentence-based, W = word-based; #dj = number of "docjudges" (ranked lists of documents and summaries).

5.2 Evaluation

Please see section 13 of the MEAD documentation for how to evaluate summaries based on precision/recall/kappa, relative utility, relevance correlation and lexical similarity.

5.3 Formatting issues: HK News segmentation and converting HK News corpus files to clusters

One challenge we faced originally and in the publication of this data was segmenting the sentences in the original HK News corpus. We segmented the HK News corpus using our own scripts, and this poses a bit of a challenge for users who have the HK corpus but do not have our docsents (with our segmentation). He have included the original scripts for segmentation of English documents in `data/tools/segment_scripts_orig`, and the scripts are documented by the author in the README file in that subdirectory.

Before you use these, the author of the scripts gives the following two warnings. First, you can only use this pipeline if you have a licence for and have already installed LTD TTT, available at `http://www.ltg.ed.ac.uk/software/ttt/`. Second, these scripts were written for the workshop and may be a bit raw.

Because of the English sentence segmentation challenge, we decided to offer a more robust way of allowing a user to regenerate our docsent files with our segmentation from the original HK News corpus. Be warned that our segmentation has its problems. For example, we are aware that our segmenter split "(and so it goes.)" into "and so it goes." and ")". Nevertheless, it will be useful to be able to reproduce our segmentation. The method for recreating English sentence segmentation relies on tables of sentence beginnings and endings. Because Chinese sentence segmentation is more straightforward, we simply incorporated the original scripts into hksar2cluster.pl.

In `data/tools/formatting`, you will find a script called hksar2cluster.pl. It expects that it will be called from that location and that the `tables` directory will be at the same level as it.

Before running it, you should change the variable $dtd_dir on line 9 of the script. This specifies the location of the dtd file for the docsents that will be created. The variable is currently set to "/clair6/projects/ldc/documentation/dtd." This will likely do you no good. You will also need to change the path variable to your HK News corpus. This variable is called $hkcorpus_dir.

This script takes various arrangements of original HK News files and outputs a cluster with the sentences segmented in the same way that our sentences were.

The commandline for this script is:

```
./hksar2cluster.pl NAMEOFCLUSTER HKSARFILE
```

or

```
./hksar2cluster.pl testingdir hksar_corpus/1997/english/19970701_001.e
```

The script will read the HKSARFILE and create a single document cluster with the NAMEOFCLUSTER. Theoretically it will allow the NAMEOFCLUSTER to include a path, as in /a/b/c/NAMEOFCLUSTER, as long as /a/b/c already exists. This was originally designed, though, to create the cluster in the same directory.

This script can also take as input the name of a directory of HKSAR files, as in:

```
./hksar2cluster.pl NAMEOFCLUSTER HKSARDIR
```

It will convert all of the HKSAR files in HKSARDIR to docsents and then put them into one multi-document cluster.

Another way of calling this script is:

```
./hksar2cluster.pl NAMEOFCLUSTER HKSARFILE1 HKSARFILE2...
```

This will take as many HK News files as specificed and create one multi-document cluster from them.

This method should recreate the original docsents we used with the following two inconsequential exceptions. First, we do not include the original paragraph numbers and rsnt numbers in the recreated docsents; we treat every sentence as a new paragraph. The other small difference is that we deleted any whitespace before and after the sentences.

This method will work for re-creating our sentence segmentation for Chinese documents. We have included our original Chinese sentence segmentation scripts within this larger script. Wai Lam kindly offered a more recent version of his word segmenter, and we use this now in hksar2cluster.pl.

5.4 Converting stg files to docsents and clusters

As mentioned, we have included within the 40 clusters the stg files, which represent our initial segmentation of the HK News corpus. If you would like to convert those files to clusters, please read on.

Under the tools directory, there is a subdirectory named "formatting," in which we have included a script called stg2cluster.pl. This is designed to take a `stg` file from one of the clusters and create single-document clusters from each news article in the file. This was designed to make it easier for people to work with the SummBank 1.0 data. This script maintains paragraph and paragraph_sent (rsnt in MEAD lingo) numbers. The input for this script looks like this:

```
<DOC id=19990526_010.e>
 <TEXT>
 <s id=19990526_010.e.1.1>
 Eastern Traffic Day
 <s id=19990526_010.e.1.2>
 Eastern District Police officers have issued a total of 126 fixed penalty...
</TEXT>
</DOC>
```

Figure 12: An example HK News corpus file

It is assumed that there could be more than one news article in a given file, and this script creates a new cluster for each file. This was the case in the data that was used during the workshop.

Before beginning, UTIL.pm must be in the same directory as stg2cluster.pl, or the location of UTIL.pm must be your perl library path.

The commandline is:

```
./stg2cluster.pl FILE TARGETDIRECTORY
```

so with the test material:

```
./stg2cluster.pl stg.test.txt HERE
```

This will then create 10 single-document clusters in the subdirectory HERE. Each cluster will be named for the one document in its cluster.

If you do not specify a target directory, the script will default to the current directory.

This script and other scripts for converting text files and html files to our cluster format are fully documented in the README file in the "formatting" subdirectory and in the latest MEAD documentation, available at `http://www.summarization.com/mead`.

For updates on these scripts, please visit `http://www.summarization.com/mead/addons`, and `http://www.summarization.com/summbank`.

5.5 Converting extracts to summaries

Although this is covered in the MEAD documentation, we think this information is important enough to repeat here. You will need to have the Hong Kong SAR corpus (LDC2000T46) available to create summaries from documents not included in the 40 clusters created for the multi-document summarization experiments.

For English and Chinese there are two options for accomplishing this task. The easier method is to find the script called `tools/formatting/hkextract2summary.pl`. Before using this, you must make sure that the directory `tables` is at the same level as the script. Then you must change the value of the variable $hkcorpus_dir. This should point to the location of your HK News corpus, and it is assumed that the structure of the corpus is the same as it is on the cds. The commandline for this script is:

```
./hkextract2summary.pl EXTRACT
```

This script will output the summary as MEAD does, which means that each sentence is preceded by the number of its order in the extract surrounded by straight brackets.

The second option is to create MEAD-readable clusters; see the section above on converting Hong Kong News corpus data to clusters.

Once you have a cluster (be it a single document or multi-document cluster) you will need `extract-to-summary.pl` which is included in the MEAD (3.07) release (under mead307/stable/mead/bin/).

The syntax to use is:

```
./extract-to-summary.pl cluster_file docsent_directory extract_file
```

If one wanted a summary from an extract called GA3.10.extract, then one would run:

```
% ./extract-to-summary.pl ../data/GA3/GA3.cluster ../data/GA3/docsent \
     ../data/GA3/GA3.10.extract
```

Be warned that this method might not work on the small number of CYL extracts which already include the text.

As in the script to convert HKSAR News corpus files to clusters, we use the original scripts for segmenting Chinese sentences, and we use Wai Lam's most recent word-segmenter.

5.6 Setting the dtd paths

As the data currently stands, in each xml file, there is a pointer to a dtd file in the same directory as the file. To change the paths, go to the data/tools folder and run dtder.pl. Before running this, set the directory where you want to begin this procedure (variable $first_dir) and the absolute path to your dtd files (variable $dtd_location). Examples are given in the file. When you run this the first time, it will create a file called "inventory", which is a list of all files in that directory and below. In the second phase, the script runs through this file and changes the dtd in every file in "inventory." If you have to run the script again, it will read "inventory" and won't have to recreate it each time. However, if you want to run this on a different directory than you originally did, you'll want to rename, move or remove "inventory." There are a number of more economical ways of coding this procedure, and I encourage you to use your own method, but this had advantages over some of the more obvious methods.

6 Known problems

In a few cases, there is some disagreement about the right number of sentences in the manual extracts (those extracts which were created automatically from each judge's sentjudge scores).

Before giving the genuine problem, we should make it clear that there are a number of cases where extracts from different judges have different numbers of sentences. Most of the time, this happens in extracts whose length was calculated based on the number of words in the summary. Therefore, it can and frequently does happen that judge 1 may pick, for example, 2 long sentences, while judge 3 may pick 4 shorter sentences. Both summaries may be 50 words long, but the number of sentences picked differs.

However, there is one genuine problem among the single document extracts for one file in cluster 1018. We are aware that the extracts that were created for 19990211_020.e.extract have different numbers of sentences across the three judges for the following compression rates: 20S, 30S, 40S, and 50S.

In a very small number of cases, some of the extract files are empty. We are aware that the files listed in figure 13 are empty. These files are all in:
`data/automatic/summaries/mead/S-Q00125-E-20S-MEADS007/extract`.

19980626_030.e.extract	19980626_031.e.extract
19980626_032.e.extract	19980626_033.e.extract
19980727_012.e.extract	19980727_013.e.extract
19980727_016.e.extract	19980727_017.e.extract
19980829_008.e.extract	19980829_009.e.extract
19980829_010.e.extract	19980829_011.e.extract
19980921_021.e.extract	19980921_022.e.extract
19980921_023.e.extract	19980921_024.e.extract
19981020_022.e.extract	19981020_023.e.extract
19981020_024.e.extract	19981020_025.e.extract
19981120_007.e.extract	19981120_008.e.extract
19981120_009.e.extract	19981120_010.e.extract
19981216_015.e.extract	19981216_016.e.extract
19981216_017.e.extract	19981216_018.e.extract
19990116_001.e.extract	19990116_002.e.extract
19990116_003.e.extract	19990116_005.e.extract
19990212_011.e.extract	19990212_012.e.extract
19990212_013.e.extract	19990212_014.e.extract
19990311_020.e.extract	19990311_021.e.extract
19990311_022.e.extract	19990311_023.e.extract
19990412_015.e.extract	19990412_017.e.extract
19990412_019.e.extract	19990412_021.e.extract
19990507_034.e.extract	19990507_035.e.extract
19990507_036.e.extract	19990507_037.e.extract
19990604_031.e.extract	19990604_032.e.extract
19990604_033.e.extract	

Figure 13: Empty files

We are also aware that `19970920_001.e.docsent.extract` and `19970921_003.e.docsent.extract` are missing under
`data/automatic/summaries/lexchain/S-GENERC-E-30S-LEXCHAIN`.

Further, we are aware that we are missing extracts for:
`data/automatic/summaries/mead/S-GENERC-C-05S-MEADORIG`. The summaries were preserved, and we may try to create the extracts from these at a future date. Please check `http://www.summarization.com/summbank` for updates on this issue.

Finally, we are aware that our extract to summary routine and our segmentation routine do not work properly for 19981014_004.e. We therefore include our original docsent for this file in data/tools/formatting/extras.

7 Credits

This corpus was built in Summer 2001 at the JHU workshop by a team consisting of the following people: Dragomir Radev, Simone Teufel, Wai Lam, Horacio Saggion, Danyu Liu, Hong Qi, Elliott Drabek, John Blitzer, and Arda Çelebi.

The following people have also been involved with the project: Inderjeet Mani, Chin-Yew Lin, Sanjeev Khudanpur, Greg Silber.

Tim Allison has been responsible for arranging and documenting the data.

We would like to thank Stephanie Strassel and Cristina Tofan at the LDC for their generous help on this project.

8 Updates, contacts, tools, patches, mailing list and further reading

For all of this information, please refer to SummBank's website: `http://www.summarization.com/summbank`.

In the following, we present the dtds for the various data structures used during the workshop. These are all available in SummBank 1.0 in `documentation/dtd/`.

A XML DTDs

A.1 cluster.dtd

```
<!ELEMENT CLUSTER (D)*>
<!ATTLIST CLUSTER
   LANG (CHIN|ENG) "ENG">

<!ELEMENT D EMPTY>
<!ATTLIST D
   DID   ID    #REQUIRED
   ORDER CDATA #IMPLIED>
```

A.2 docjudge.dtd

```
<!ELEMENT DOC-JUDGE (D)*>
<!ATTLIST DOC-JUDGE
     QID    CDATA #REQUIRED
     SYSTEM CDATA #REQUIRED
     LANG   (CHIN|ENG) "ENG">

<!-- LANG refers to the language of the retrieval process.
     Thus, it is the language of the documents.
     However, the original language of the query might be
     different.
     Look this up in QID. -->

<!ELEMENT D EMPTY>
<!ATTLIST D
     DID   ID     #REQUIRED
     RANK  CDATA  #IMPLIED
     CORR-DOC CDATA #IMPLIED
     SCORE CDATA  #REQUIRED>
```

A.3 docpos.dtd

```
<!--  DTD for POS tagged text  -->
<!ELEMENT DOCPOS (EXTRACTION-INFO?, BODY)>
<!ATTLIST DOCPOS
   DID   CDATA      #REQUIRED
   DOCNO CDATA      #IMPLIED
   LANG  (CHIN|ENG) "ENG"
   CORR-DOC CDATA   #IMPLIED>
   <!-- DID : documentid
       LANG: language    -->

<!ELEMENT EXTRACTION-INFO EMPTY>
<!ATTLIST EXTRACTION-INFO
    SYSTEM  CDATA  #REQUIRED
    RUN     CDATA  #IMPLIED
    COMPRESSION CDATA #REQUIRED
    QID     CDATA  #REQUIRED>

<!ELEMENT BODY (HEADLINE?,TEXT)>

<!ELEMENT HEADLINE (S)*>
<!ELEMENT TEXT (S)*>

<!ELEMENT S (W)*>
<!ATTLIST S
    PAR  CDATA #REQUIRED
    RSNT CDATA #REQUIRED
    SNO  CDATA #REQUIRED>
    <!-- PAR: paragraph no
         RSNT: relative sentence no (within paragraph)
         SNO: absolute sentence no -->

<!ELEMENT W (#PCDATA)>
<!ATTLIST W
    C   CDATA #REQUIRED
    L   CDATA #IMPLIED>

<!-- C is the POS category. L is the lemma -->
```

A.4 docsent.dtd

```
<!--  DTD for sentence-segmented text  -->
<!ELEMENT DOCSENT (EXTRACTION-INFO?, BODY)>
<!ATTLIST DOCSENT
   DID   CDATA      #REQUIRED
   DOCNO CDATA      #IMPLIED
   LANG  (CHIN|ENG) "ENG"
   CORR-DOC CDATA   #IMPLIED>
   <!-- DID : documentid
       LANG: language    -->

<!ELEMENT EXTRACTION-INFO EMPTY>
<!ATTLIST EXTRACTION-INFO
    SYSTEM  CDATA  #REQUIRED
    RUN     CDATA  #IMPLIED
    COMPRESSION CDATA #REQUIRED
    QID     CDATA  #REQUIRED>

<!ELEMENT BODY (HEADLINE?,TEXT)>

<!ELEMENT HEADLINE (S)*>
<!ELEMENT TEXT (S)*>

<!ELEMENT S (#PCDATA)>
<!ATTLIST S
    PAR  CDATA #REQUIRED
    RSNT CDATA #REQUIRED
    SNO  CDATA #REQUIRED>
    <!-- PAR: paragraph no
         RSNT: relative sentence no (within paragraph)
         SNO: absolute sentence no -->
```

A.5 document.dtd

```
<!--  DTD for original, non-segmented text  -->
<!ELEMENT DOCUMENT (EXTRACTION-INFO?, BODY)>
<!ATTLIST DOCUMENT
   DID   CDATA      #REQUIRED
   DOCNO   CDATA      #IMPLIED
   LANG  (CHIN|ENG) "ENG"
   CORR-DOC CDATA   #IMPLIED>
   <!-- DID : documentid
       LANG: language    -->

<!ELEMENT EXTRACTION-INFO EMPTY>
<!ATTLIST EXTRACTION-INFO
    SYSTEM  CDATA  #REQUIRED
    RUN     CDATA  #IMPLIED
    COMPRESSION CDATA #REQUIRED
    QID     CDATA  #REQUIRED>

<!ELEMENT BODY (HEADLINE?,TEXT)>

<!ELEMENT HEADLINE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
```

A.6 extract.dtd

```
<!ELEMENT EXTRACT (S)*>
<!ATTLIST EXTRACT
   QID         CDATA #REQUIRED
   COMPRESSION CDATA #REQUIRED
   SYSTEM      CDATA #REQUIRED
   JUDGE       CDATA #IMPLIED
   JUDGENO     CDATA #IMPLIED
   RUN         CDATA #IMPLIED
   SENTS_TOTAL CDATA #IMPLIED
   WORDS_TOTAL CDATA #IMPLIED
   LANG        CDATA #REQUIRED>

<!ELEMENT S EMPTY>
<!ATTLIST S
   ORDER       CDATA #REQUIRED
   DID         CDATA #REQUIRED
   SNO         CDATA #IMPLIED
   PAR         CDATA #IMPLIED
   RSNT        CDATA #IMPLIED
   UTIL        CDATA #IMPLIED>
```

A.7 mead-config.dtd

```
<!ELEMENT MEAD-CONFIG (FEATURE-SET, CLASSIFIER, RERANKER,
COMPRESSION) >
<!ATTLIST MEAD-CONFIG
   LANG CDATA #REQUIRED
   CLUSTER-PATH CDATA #IMPLIED
   DATA-DIRECTORY CDATA #IMPLIED
   TARGET CDATA #IMPLIED >

<!ELEMENT FEATURE-SET (FEATURE*) >
   BASE-PATH CDATA #IMPLIED >

<!ELEMENT FEATURE EMPTY >
<!ATTLIST FEATURE
   FEATURE CDATA #REQUIRED >

<!ELEMENT CLASSIFIER EMPTY >
<!ATTLIST CLASSIFIER
   COMMAND-LINE CDATA #REQUIRED
   SYSTEM CDATA #IMPLIED
   RUN CDATA #IMPLIED >

<!ELEMENT RERANKER EMPTY >
<!ATTLIST RERANKER
   COMMAND-LINE CDATA #REQUIRED>

<!ELEMENT COMPRESSION EMPTY >
<!ATTLIST COMPRESSION
   BASIS (sentences|words) #REQUIRED
   PERCENT CDATA #IMPLIED
   ABSOLUTE CDATA #IMPLIED >
```

A.8 query.dtd

```
<!ELEMENT QUERY (TITLE,DESCRIPTION?,NARRATIVE?)>
<!ATTLIST QUERY
   QID  CDATA #REQUIRED
   QNO  CDATA #REQUIRED
   LANG (CHIN|ENG) "ENG"
   TRANSLATED (YES|NO) "NO"
   ORIGLANG (CHIN|ENG) "CHIN"
   TRANS-METHOD (AUTO|MAN) "AUTO">

<!-- QID: unique query no, eg. 125-CA or 125-E
     QNO: LDC query no for content, eg. 125
     LANG: of query
     TRANSLATED: is it an original query or not?
     ORIGLANG: If translated, from which language (from the other
     one, of course!)
     TRANS-METHOD: Automatically translated or manually? -->

<!ELEMENT TITLE       (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT NARRATIVE   (#PCDATA)>
```

A.9 reranker-info.dtd

```
<!-- DTD for input to rerankers -->
<!ELEMENT RERANKER-INFO (COMPRESSION, CLUSTER, SENT-JUDGE)

<!ELEMENT COMPRESSION EMPTY>
<!ATTLIST COMPRESSION
    PERCENT CDATA #REQUIRED
    BASIS   CDATA #REQUIRED>

<!ELEMENT CLUSTER (D)*>
<!ATTLIST CLUSTER
    LANG (CHIN|ENG) "ENG">

<!ELEMENT D EMPTY>
<!ATTLIST D
    DID   ID    #REQUIRED
    ORDER CDATA #IMPLIED>

<!ELEMENT SENT-JUDGE (S)*>
<!ATTLIST SENT-JUDGE
    QID  CDATA  #REQUIRED>

<!ELEMENT S (JUDGE)*>
<!ATTLIST S
    DID  CDATA #REQUIRED
    PAR  CDATA #REQUIRED
    RSNT CDATA #REQUIRED
    SNO  CDATA #REQUIRED>

<!ELEMENT JUDGE EMPTY>
<!ATTLIST JUDGE
    N    CDATA #REQUIRED
    UTIL CDATA #REQUIRED>
```

A.10 sentalign.dtd

```
<!ELEMENT SENTALIGN (SENT+)>
<!ATTLIST SENTALIGN
   ENG         CDATA #REQUIRED
   CHI         CDATA #REQUIRED
   LANG        CDATA #REQUIRED>

<!ELEMENT SENT EMPTY>
<!ATTLIST SENT
   ORDER       CDATA #REQUIRED
   EDID        CDATA #REQUIRED
   ESNO        CDATA #REQUIRED
   CDID        CDATA #REQUIRED
   CSNO        CDATA #REQUIRED>

<!-- ORDER: the pairwise number
     EDID:  english document name
     ESNO:  english sentence number
     CDID:  chinese document name
     CSNO:  chinese sentence number -->
```

A.11 sentfeature.dtd

```
<!ELEMENT SENT-FEATURE (S)*>

<!ELEMENT S (FEATURE)*>
<!ATTLIST S
    DID  CDATA #REQUIRED
    SNO  CDATA #REQUIRED>

<!ELEMENT FEATURE EMPTY>
<!ATTLIST FEATURE
    N CDATA #REQUIRED
    V CDATA #REQUIRED>
```

A.12 sentjudge.dtd

```
<!ELEMENT SENT-JUDGE (S)*>
<!ATTLIST SENT-JUDGE
    QID  CDATA  #REQUIRED>

<!ELEMENT S (JUDGE)*>
<!ATTLIST S
    DID  CDATA #REQUIRED
    PAR  CDATA #REQUIRED
    RSNT CDATA #REQUIRED
    SNO  CDATA #REQUIRED>

<!ELEMENT JUDGE EMPTY>
<!ATTLIST JUDGE
    N    CDATA #REQUIRED
    UTIL CDATA #REQUIRED>
```

A.13 sentrel.dtd

```
<!ELEMENT SENT-REL (R)*>

<!ELEMENT R (RELATION)*>
<!ATTLIST R
   SDID         CDATA   #REQUIRED
   SSENT        CDATA   #REQUIRED
   TDID         CDATA   #REQUIRED
   TSENT        CDATA   #REQUIRED>

<!ELEMENT RELATION EMPTY>
<!ATTLIST RELATION
   TYPE         CDATA   #REQUIRED
   JUDGE        CDATA   #REQUIRED>
```