ModDBS-X^M: A Diversity Based Summarizer for DUC2001

Tadashi Nomoto
National Institute of Japanese Literature
1-16-10 Yutaka Shinagawa
Tokyo 142-8585 Japan
nomoto@nijl.ac.jp

1 Description of the system

ModDBS-X^M is a clustering based single document summarizer. It is an open-domain extractive summarizer, demanding of the input nothing more than the availability of basic IR statistics such as term and document frequency. Therefore it could be adapted for any language and domain without much effort.

The system goes through three major states to generate a summary: data preparation, summarization, and post-summarization. An input text is first examined for its conformity to the XML syntax; some portions of it are extracted for use in summarization, which are passed on to the sentence selection step, which in turn builds diverse topical clusters over the input and chooses representative sentences thereof. The selected sentences are then put through a post-summarization process, where parenthetical expressions are identified and removed.

1.1 Data preparation

The system takes as input an XML document and runs a sequence of operations to prepare data for summarization. These include:

1. validating the document with an XML parser (Derksen, 2000)
2. isolating particular XML elements for use in summarization, i.e. subtrees rooted at `<TEXT>`, `<HEADLINE>` (or `<HL>`) and `<LEADPARA>` (or `<LP>`).
3. identifying sentences in selected elements using MXTERMINATOR (Ratnaparkhi, 1997)
4. assigning POS tags to word tokens found in the sentences, which is done with LimaTK (Yamashita, 1999).

Texts not compliant with the XML 1.0 standards are rejected. (There were a few of them in the DUC test data.) Also rejected are texts with more than one `<TEXT>` element since we took `<TEXT>` to mean the main body of the text, as was the case with the majority of texts in the test data. (In the FBIS articles, we took `[Text]` as `<TEXT>`.)

The purpose of the initial sequence of operations is to identify sentences in particular portions of a well-formed XML document and to produce POS information on each word in the sentences. Let us call the set of sentences generated by the preparation process, a *source text*.

The initial sequence is followed by the creation of a table holding the tfidf weight for each word type in the source text, which is defined as: for a given term x,

$$T(x) = (1 + \log(tf(x))) \cdot idf(x)$$

$tf(x)$ is the frequency of term x in a document, $idf(x)$ is the inverse document frequency of x. The document frequency of a term was determined with a reference to the test data set alone.

For index terms, we used everything except for punctuation marks, non-linguistic symbols. We did not use a stoplist except for those elements already excluded.

1.2 Summarization Core Steps

The summarization step here uses a slightly modified version of DBS/X^M (Nomoto and Matsumoto, 2001). (Hence the name ModDBS-X^M.) It takes as input a source text with supplementary information such as POS assignments and the weight table, and performs the following operations:

1. **Find-Diversity**: Find diverse topic clusters in the source text.
2. **Reduce-Redundancy**: For each topic cluster, locate most important sentences and take them as representative of that cluster.

The goal of the summarization here is to extract sentences in such a way that they collectively create a general picture of what the source text is about.The following look at some details of each operation.

1.2.1 Find-Diversity

Find-Diversity is a clustering algorithm built upon the K-means extended with Minimum Description Length Principle (MDL) (Rissanen, 1997; Li, 1998). In fact the algorithm here is an MDL-version of X-means (Pelleg and Moore, 2000), an extension of the K-means clustering algorithm with the functionality

Table 2: Results on grammaticality, cohesion, and organization for single-doc summaries. Figures in each criterion refer to the total number of points scored. The "average" section lists total points averaged over documents. Figures in parentheses indicate the average score for the relevant criterion.

system	grammar	cohesion	organization	#docs	total	average
V	549 (3.709)	451 (3.047)	475 (3.209)	148	1475	9.966
R	561 (3.816)	434 (2.952)	470 (3.197)	147	1465	9.966
O	554 (3.743)	422 (2.851)	452 (3.054)	148	1428	9.649
T	534 (3.584)	410 (2.752)	460 (3.087)	149	1404	9.423
P	558 (3.745)	390 (2.617)	424 (2.846)	149	1372	9.208
W	521 (3.593)	393 (2.710)	409 (2.821)	145	1323	9.124
Q	556 (3.732)	381 (2.557)	397 (2.664)	149	1334	8.953
ModDBS	**531 (3.588)**	**373 (2.520)**	**409 (2.764)**	**148**	**1313**	**8.872**
S	543 (3.669)	365 (2.466)	380 (2.568)	148	1288	8.703
Y	481 (3.228)	369 (2.477)	403 (2.705)	149	1253	8.409
Z	493 (3.309)	334 (2.242)	363 (2.436)	149	1190	7.987

Table 1: The average per doc units for the models and the peers. "All systems" gives the average unit length for the peer systems.

model	ModDBS	all systems
6.39	3.41	4.08

of estimating K, the number of clusters which otherwise needs to be supplied by the user. We call our adaptation of X-means 'X^M means.'

K-means is a hard clustering algorithm that produces a clustering of input data points into K disjoint subsets. It dynamically redefines a clustering by relocating each centroid to the center of mass of points associated with it and re-associating the centroid with points closest to it.

In an effort to improve the time-efficiency and scalability of clustering, we have modified **Find-Diversity** as formulated as part of DBS/X^M to incorporate a strategy for rapidly refining initial points for clustering (Bradley and Fayyad, 1998). The modified **Find-Diversity** determines initial points by clustering, using K-means, *centroids* of a set of subsamples drawn randomly from the source text (each subsample containing 10% of the source), and selecting, among them, those which give a least distorted clustering for the centroids.

1.2.2 Reduce-Redundancy

Reduce-Redundancy is a simple sentence ranking algorithm based on Zechner (1996), where one takes the weight of a given sentence as the sum of tfidf scores of index terms occurring in that sentence. The weight W of sentence s is given by:

$$W(s) = \sum_{x \in s} T(x)$$

where x denotes an index term. **Reduce-Redundancy** applies itself to each topic cluster from **Find-Diversity**, ranks sentences there according to the weighting model above, and selects those with highest scores. As is apparent from the above formula, the sentence weight is *not* normalized for length. The idea of choosing as representative best scoring sentences is intended to minimize the loss of the resulting summary's relevance to a potential query. (The exact number of sentences to select from each cluster depends on the desired length of a summary.)

1.3 Post-Summarization Step

The post-summarization process operates on each of the sentences from the summarization step. What it does is a simple removal of parenthetical expressions from a sentence. Other than that, no paraphrasing or transformation of a sentence is performed; conjunctives such as *but* or *and* occurring sentence initially are also left intact. (Incidentally, we have found that about a half of the DUC per-doc summaries which have a low rating ("2") for grammaticality had some parentheses incorrectly removed.) Finally, the sentences are put in the order in which they appear in the source text.

2 Discussion

The average number of peer units (PUs) for (per-doc) summaries generated by ModDBS was 3.41 while that of model units (MUs) was 6.39. The average count of peer units for the participating systems was 4.08 (Table 1). Though the precise definition of model or peer unit was not available at the time of writing, the DUC results appear to indicate that model summaries are about twice as long in unit as those generated by the system; this would mean that in order to make system summaries comparable in unit length to model summaries, one has to make them twice as long, i.e. 200-word long. The number of PUs for human created (per-doc) summaries was found to be 5.53 on average, which is far closer to the average number of MUs.

The reason why the system summaries tend to contain less units (and therefore achieve less coverage) apparently has to do with the system's inability to properly perform within sentence (or unit) reductions; since a per-doc summary is limited to 100 words in length, to increase the number of units included in a summary demands reduction of words contained in a unit, which, however, does not happen with the present system, as it does no more than the simple removal of parentheses.

Table 2 lists the results on the quality metrics, overall peer grammaticality, cohesion, and organization of the single doc summaries. As is apparent from the table, the systems vary more widely in cohesion and organization than in grammaticality. The quality metric is given on the nominal scale of 0 to 4: all = 4 ,most = 3 ,some = 2, hardly any = 1, none = 0. On the average, ModDBS scored 3.588 for grammaticality (over "most" level), 2.520 for cohesion (somewhere between "most" and "some"), and 2.764 for organization (over "some", close to "most"). It is not surprising to have a high mark on grammaticality, since the system usually generates extracts verbatim from the text. What could be surprising, perhaps, is that the system performs moderately well on organization, despite the fact that it takes no account of how the summary should be textually organized.

References

P. S. Bradley and Usama M. Fayyad. 1998. Refining initial points for k-means clustering. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML98)*, pages 91–99. Morgan Kaufmann.

Enno Derksen. 2000. Perl module: libxml-enno. http://www.cpan.org.

Hang Li. 1998. *A Probabilistic Approach to Lexical Semantic Knowledge Acquistion and Structural Disambiguation*. Ph.D. thesis, University of Tokyo, Tokyo.

Tadashi Nomoto and Yuji Matsumoto. 2001. A new approach to unsupervised text summarization. In *Proceedings of the 24th International ACM/SIGIR Conference on Research and Development in Informational Retrieval*, New Orleans, September. ACM.

Dan Pelleg and Andrew Moore. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000)*, pages 727–734, Stanford University, CA, June-July. Morgan Kaufmann.

Adwait Ratnaparkhi. 1997. MXTERMINATOR. http://www.cis.upenn.edu/˜adwait/statnlp.html.

Jorma Rissanen. 1997. Stochastic complexity in learning. *Journal of Computer and System Sciences*, 55:89–95.

Tatuo Yamashita. 1999. LimaTK. http://cl.aist-nara.ac.jp/˜tatuo-y/ma.

Klaus Zechner. 1996. Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 986–989. Copenhagen, Denmark.