Cross-Document Summarization by Concept Classification

Hilda Hardy, Nobuyuki Shimizu, Tomek Strzalkowski, Liu Ting, Xinyang Zhang
NLIP Laboratory, University at Albany
1400 Washington Avenue, Albany, NY 12222
and

G. Bowden Wise
GE Corporate R&D Center
1 Research Circle, Niskayuna, NY 12309

Abstract

In this paper we describe a Cross Document Summarizer XDoX designed specifically to summarize large document sets (50-500 documents and more). Such sets of documents are typically obtained from routing or filtering systems run against a continuous stream of data, such as a newswire. XDoX works by identifying the most salient themes within the set (at the granularity level that is regulated by the user) and composing an extraction summary, which reflects these main themes. In the current version, XDoX is not optimized to produce a summary based on a few unrelated documents; indeed such summaries are best obtained simply by concatenating summaries of individual documents. We show examples of summaries obtained in our tests as well as from our participation in the first Document Understanding Conference (DUC).

1. XDoX Overview

The XDoX system (Cross Document Summarizer) is designed to summarize sets of 50-500 documents that have been retrieved or routed from a text database, the Internet, or a news source, according to a query or a user-defined profile. Built for information analysts, the system uses clustering techniques to subdivide the documents into meaningful topics and themes, and to separate unrelated material. XDoX presents the user with two kinds of overall summary, one with more detail related to the complexity of the document set, and one with fewer details and limited length. In addition, a summary of each topical cluster is available via a Graphical User Interface. The GUI also allows the user to view individual passages or full documents. Thus the system answers both the indicative and the exploratory needs of our customers.

XDoX has been developed over the last 2 years with several intermediate designs considered. An earlier version of the system (Stein et al, 2000) produced clusters of documents according to their mutual similarity, which was based primarily on straightforward term overlap between documents. In addition, WordNet lexical database (Fellbaum et al., 1998; Miller, 1995) was used to facilitate the matching of synonyms and other related terms. This early work focused on evaluation of various document clustering techniques but we found that most known clustering methods could not alone support an effective summarizer – whether or not WordNet is used, the resulting clusters were often of poor quality, formed around common but semantically unimportant terms.

Our new approach improves the quality of the clusters and the summaries chiefly by implementing concept-based clustering and summarization. Instead of clustering entire documents, we cluster passages (Mitra et al., 1997), or sequences of text, which usually correspond to the natural para-

graphs designed by the author or editor, or that may be obtained automatically (Hearst, 1994). Our similarity metric is based on *n*-gram matching rather than just single-term overlap. For example, a word in a sequence of six words (or a 6-gram) receives proportionately more weight than the same word occurring in two distinct three-word sequences (or tri-grams), which in turn is weighted more than if it were found in three bi-grams, and so on. Individual term weights are computed at document level using a variant of pivoted document length normalization metric (Singhal et al., 1996) and added to *n*-gram weights.

We have found that our new approach produces excellent summaries in most cases. The summaries are based on high-quality clusters that form around significant common concepts or themes that occur repeatedly across the set of documents. The paragraph is a useful semantic unit for conceptual clustering, because most writers view a paragraph as a topical unit, and organize their thoughts accordingly. A few examples of themes detected in large document sets are: "lie detector test" (a theme occurring in a set of documents on labor relations), "South Georgia and South Sandwich Islands" (a theme found in a set of documents on Falkland oil exploration), "blood alcohol levels" (a theme detected in the set on automobile accidents), etc.

2. An Outline of XDoX Architecture

The XdoX system is written in Java, except for our single-document summarizer, which is written in C++. The system has several distinct modules. Their basic tasks, inputs and outputs are shown in Figure 1.

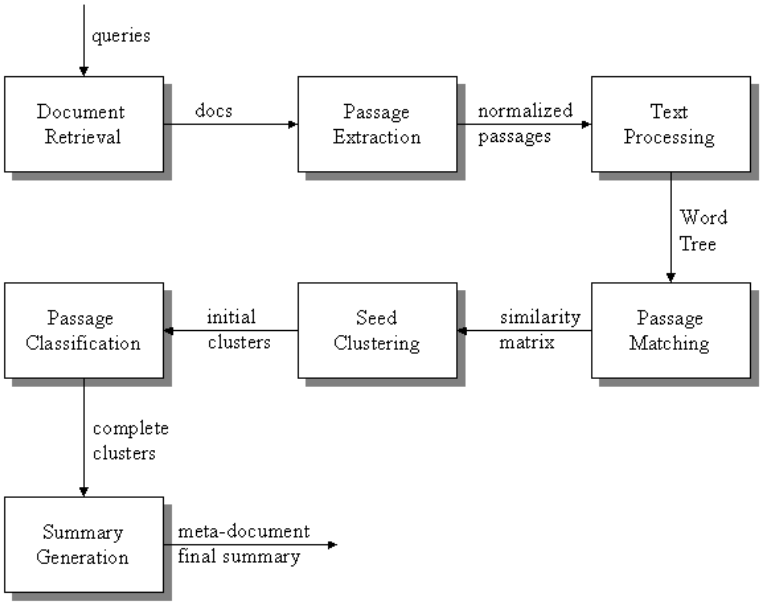

Figure 1. XDoX System Architecture

3. Text Processing

As shown above, documents are first chunked into paragraphs according to existing paragraph boundaries. SGML tags, lists of keywords, author by-lines, news sources, locations, etc. are all removed. Subtitles and one-line paragraphs are merged with the following text. Titles of articles are eligible to be clustered, except when they have fewer than three terms.

In preparation for comparing *n*-grams in passages, stopwords are removed and the remaining words are stemmed, using the Porter algorithm. Each stem is mapped to a unique integer code. All the stemmed words are placed into a simple binary `Word Tree` structure, using Java's red-black `TreeMap` class. The tree contains a node for each stem. The value for each node is an ordered list of occurrences of the stem, including positional information (document, paragraph, location in the paragraph) as well as the code of the following stem. The `Word Tree` can be constructed in linear time.

3.1. Document-level term weighting

We chose a term weighting scheme that uses average term frequency in a document as the normalization factor. In the function

$$\frac{1+\log(tf)}{1+\log(average(tf))},$$

tf(t) is the actual frequency of term *t* in document *D*. This normalization method has been shown to perform 5.7% better than maximum term frequency based normalization for 200 TREC queries on the entire TREC collection (Singhal et al., 1996). Combining this *tf* factor with the pivoted normalization used in SMART system, we arrive at the weighting strategy:

$$\frac{\frac{1+\log(tf)}{1+\log(average(tf))}}{(1-c)average(L)+c\times L},$$

where *L* is the number of unique terms in document *D* and *c* is a constant between 0 and 1. This weighting scheme is related to BM25 used in Okapi system (Robertson et al, 1995), which has been reported to perform consistently better than the standard cosine normalization in document retrieval applications.

4. Passage similarity metric based on n-grams

After the text processing module has built the `Word Tree` and computed term weights, the passage matching module compares passages and assigns similarity scores to every pair of passages in the document set (not including pairs from the same document). The output is a table of paragraph pairs and their similarity values, represented as a matrix.

4.1. A fast algorithm for matching *n*-grams

In order to compare large numbers of documents efficiently using *n*-gram matching, we chose to work with a very small subset of all possible substrings in the documents: we look only at *n*-grams, where *n* is 1 to 6, that are actually matched somewhere among the passages in the document set. Our goal was to minimize the number of comparisons made, and so we construct *n*-grams on the fly, in a bottom-up manner, from the `Word Tree` data structure. The algorithm is as follows:

For each document D_i (D_1 to D_{n-1}):

a. Get the first word *w* in the first paragraph.
b. From the Word Tree, get a list *a* of all instances of *w* occurring in documents D_{i+1} to D_n. These are matching 1-grams. List *a* becomes the start of a "live *n*-grams" list that grows

and shrinks as matching *n*-grams are built and removed from the list when completed. (Java's `ArrayList` class is handy for this purpose.)

c. For each word *v* which follows *w* in D_i:
   1) From the Word Tree, get a list *b* of all instances of *v* in documents D_{i+1} to D_n. Each *v* either continues an n-gram, *wv,* or begins a new 1-gram.
   2) Taking advantage of the sequential order of both lists, add or insert each *v* from list *b* in the proper place in list *a.* If an *n*-gram in list *a* cannot be extended by an occurrence in list *b,* or if it would create a sequence longer than *n,* then the *n*-gram is removed from the list and stored in a matrix structure.

Invariants: 1) The list of live *n*-grams, as well as each new list of word occurrences, remains in sequential order, 2) the final word of each *n*-gram in the list is the same, and 3) each finished *n*-gram is the longest possible *n*-gram. The goal is to find the best matches between any two paragraphs, where "best" is defined as maximum length. For example, the phrase *Federal Reserve Board Chairman Alan Greenspan*, when found in two passages, is counted as a 6-gram and not six 1-grams, or three bi-grams, or any of the other possible combinations for this phrase. Sometimes overlaps must be eliminated, as in this example:

[doc 1, par 1] *Terrorist attacks (on) American targets . . . American targets.*

[doc 5, par 6] *Terrorist attacks . . . Terrorist attacks (on) American targets . . . American targets . . . attacks.*

These two sample passages produce only two *n*-gram matches: a 4-gram: *Terrorist attacks (on) American targets* and a bi-gram: *American targets*.

The above algorithm may easily be modified for other-size passage matching, sentence matching, or even phrase matching tasks, such as may be required in clustering of dynamic or streaming data, or in question answering. The running time depends on both the total number of terms and the number of distinct terms in the document set. With a smaller ratio of total terms to distinct terms, fewer operations must be performed. The average number of times each term appears in the document set is $a = n/m$, where n = *total number of terms* and m = *number of distinct terms*. The number of operations required is approximately:

$$m(\frac{a(a-1)}{2}),$$

which gives us a best-case running time of $O(n)$, when all terms are distinct, and a worst-case running time of $O(n^2)$, when all terms are the same. The actual number of operations also depends on factors such as the variation in the number of occurrences of each distinct term, as well as the distribution of terms across the documents.

The result of the *n*-gram matching stage is a matrix structure that contains all the common *n*-grams shared between each pair of paragraphs.

4.2. Passage Similarity Scores

For computing the similarity between any two passages, we use a cosine coefficient function, modified according to *n*-gram weights. Because it would be cumbersome to assign a weight to every *n*-gram we find, we weight individual terms instead, dividing the *n*-gram weight among all the terms in the *n*-gram. The weight of term T_i in passage X_i is the weight of T_i in document X (as described above) plus the *n*-gram weight. The *n*-gram weight is given as $(n_i)/n^2$, where n_i is the length of the

n-gram of which term T_i is an element, and *n* is the length of the maximum *n*-gram. When *n* is 6, for example, a term that is part of a bi-gram receives a weight premium of 0.056 in addition to its document term weight. Every term in a matching 6-gram gets an additional 0.167 weight. The final passage similarity function is as follows:

$$sim(X,Y) = \frac{\sum_{j=1}^{t} x_{ij} \times y_{ij}}{\sqrt{\sum_{j=1}^{t} (x_{ij})^2 \times \sum_{j=1}^{t} (y_{ij})^2}},$$

In the above, x_{ij} is the weight of term T_j in paragraph X_i and y_{ij} is the weight of term T_j in paragraph Y_i. When the scoring is finished, we have a similarity matrix whose $(i,j)^{th}$ entry gives the similarity between the i^{th} and j^{th} passages.

5. Seed-clustering algorithm

In order to form seed clusters we apply the well-known complete-link algorithm to our similarity matrix (Willett, 1988). This algorithm becomes computationally expensive when used over large numbers of documents, each of which may have many passages to score. We have found it both practical and effective to run the complete-link only to the point at which we reach a target number of candidate seed clusters. We want a target that will allow the clusters to illustrate a good distribution of sub-themes or concepts in the document set, one that avoids over-generalization on the one hand and too much detail on the other. For most sets of documents, a good target is $\log_2 N$, where *N* is the number of documents. Initially, each passage is a cluster. We run the algorithm as follows:

1. Merge the most similar two clusters (clusters *i* and *j*).
2. Update the similarity matrix to reflect the pairwise similarity between the new cluster *(ij)* and the original clusters. We remove all the entries for *i* and *j* and replace them with new *ij* entries.
3. Repeat steps 1 and 2 until the target number of seed clusters is reached.

We add the restrictions that a seed cluster must contain three or more passages, and that there must be at least two common terms among the first three passages in the cluster. Clusters with larger common stem sets are preferred, as well as clusters whose common stem sets do not overlap much with another cluster. After the target is reached, some candidate seed clusters may be weeded out for failing to meet these criteria.

By having the system stop after reaching a specified number of clusters, we remove another difficulty inherent in the complete-link algorithm: choosing a proper threshold. When this choice is left up to the user, he or she must often make several tries before reaching a reasonable number of clusters. If the threshold is high, too many small clusters are formed around narrow or incidental concepts. If the threshold is too low, the clusters are fewer and larger, and the topics become fuzzy or obscured.

However, if the system is set to aim for a target number of candidate seed clusters, it will adapt its performance to the specific makeup of the document set at hand: it will adjust for document sets that happen not to have many subtopics; and it will likewise adjust for document sets that cover a wider range of themes or concepts.

6. Classification algorithm

All remaining passages are classified as satellites around the seed clusters. For this stage we perform M-bin classification, where M is the number of seeds. If a passage has no similarity to any of the seeds, it is placed into a miscellaneous "trash cluster". Passages in a cluster are presented in descending order: seed passages come first, in the order in which they were added to the cluster, so that those with the tightest similarity to one another are shown first. Next come the satellite passages, ordered according to their degree of similarity with the seed cluster.

7. Generating 2 kinds of summaries

After the clusters are formed, we create a "meta-document", selecting one of the highest-scoring, or most characteristic, passages from each cluster, and concatenating them together. Next, our single-document summarizer creates a summary of this meta-document, using the query terms, if any, as the "title". The user, then, has two types of summary to view. The meta-document is more suitable for groups of documents that describe similar but isolated events, such as alcohol-related traffic accidents, or different people who have something in common, such as winners of the Nobel Peace Prize. The summary of the meta-document is more appropriate for groups of documents, which are all related to the same topic or event, such as oil exploration in the Falkland Islands, or the U.S. Presidential election in the year 2000.

8. DUC Participation

The Albany/GE team participated in both the single-document and the multi-document tracks in DUC 2001. For cross-document summarization, the XDoX system works best on large document sets that have multiple themes. For sets that have fewer than 20 documents, as in the DUC data, our system works well with some parameter adjustments, as long as the documents contain sufficient common concepts around which to form clusters. We have seen excellent results from document sets pertaining to a single event, topic, or person, such as *gun control*, *the eruption of Mt. Pinatubo, the Charles Keating scandal,* or *the Rodney King beating*. XDoX has difficulty generating a longer, more detailed summary when there are 10 or fewer documents concerning isolated incidents, such as gas explosions or earthquakes, or when there is little repetition (see more examples in the appendix). We are considering augmenting the system with selective use of our single-document summarizer (Strzalkowski et al., 1999).

The following examples were generated by the XDoX system from the DUC 2001 data.

<multi size="50" docset="d06a">

A panel that investigated the Los Angeles Police Department after officers were videotaped beating a motorist has decided not to seek Chief Daryl Gates' resignation and neither blamed nor cleared him.

In mid-June, a lawsuit against the Los Angeles Police Department was settled when police agreed to stop using a martial-arts weapon, nunchakus, while arresting anti-abortion advocates.

</multi>

<multi size="50" docset="d13c">

President Bush on Monday nominated Clarence Thomas, a conservative Republican with a controversial record on civil rights, to replace retiring Justice Thurgood Marshall on the Supreme Court.

CLARENCE THOMAS; Born: June 23, 1948, in Pinpoint, Ga. Education: B.A. from Holy Cross College, 1971; J.D. from Yale Law School, 1974.

</multi>

<multi size="100" docset="d22d">

Firefighters in California, Michigan, Montana, Wyoming and Utah battled holiday weekend fires, which blackened more than 6,000 acres of forest and wilderness areas.

Elsewhere in Utah, the Uinta Canyon fire had burned 3,850 acres 20 miles north of Roosevelt in the Ashley National Forest. Forest Service spokeswoman Cece Stewart said three helicopters scattered incendiary bombs made of chemically treated plastic balls on an unburned 200-acre area between fire lines and the main fire in an effort to stop the fire's advance.

</multi>

<multi size="100" docset="d39g">

Officials estimate the tunnel trains may carry 28 million passengers in the first year of operation. Eurotunnel doesn't expect a profit until the end of the century.

In London, the conservative Daily Express newspaper noted today that Britons will be able to walk to France for the first time since the Ice Age.

The tunnel's cost has soared from an initial estimate of $9.4 billion to $16.7 billion, including an extra $1.97 billion in case of unforeseen cost overruns.

Eurotunnel PLC announced Oct. 8 that it had reached an agreement with its banks on $3.5 billion in new credit. More than 200 banks are involved in financing the world's costliest tunnel.

</multi>

<multi size="400" docset="d16c">

A major earthquake rumbled through the jungles of southern Sudan on Sunday near the war-torn region's largest city, but officials said there appeared to be no casualties and little damage.

The Richter scale is a measure of ground motion as recorded on seismographs. Every increase of one number means a tenfold increase in magnitude. Thus a reading of 7.5 reflects an earthquake 10 times stronger than one of 6.5.

An earthquake of 3.5 on the Richter scale can cause slight damage in the local area, 4 moderate damage, 5 considerable damage, 6 severe damage. A 7 reading is a “major” earthquake, capable of widespread heavy damage; 8 is a “great” quake, capable of tremendous damage.

In the past 35 years there have been two major earthquakes in Hokkaido. In 1968, the Tokachi-Oki earthquake measuring a 7.9 on the Richter scale killed 52 people.

The agency said the quake had a magnitude of 7.1 on the Richter scale, the same as the Oct. 17 earthquake that devastated the San Francisco Bay area. In Washington, the U.S. Geological Survey said the quake measured 7.3 on the Richter scale.

</multi>

The following summary was obtained from the set of top 100 documents retrieved from the TREC data collection with Topic 358 on the subject of alcohol related driving fatalities.

<multi size=”400” docset=”TRECtopic358”>

Drinkers beware. When the New Year begins at midnight, a tough new law will take effect making it all the more risky to drink and drive.

Under the new law that takes effect Jan 1, a driver with 0.08% or higher is presumed to be drunk; however, those with lower levels could also be cited for drunk driving.

According to court records, Giunta's blood-alcohol content was 0.29%, more than three times the level at which a motorist is considered legally drunk.

The Department clearly and specifically limited the NPRM to consideration of whether blood testing should be used for situations in which breath testing was not readily available for reasonable suspicion and post-accident tests, or in shy lung situations. For this reason, the issue raised by some commenters of whether employers should have the flexibility or discretion to use blood testing as an alternative to breath

testing, even when breath testing is readily available in reasonable suspicion and post-accident testing or even in random or pre-employment testing, is outside the scope of the rulemaking.

The author, Sen. Bill Leonard (R-Big Bear), predicted that the measure will win legislative approval and be sent to Gov. George Deukmejian, whose Administration supports it.

</multi>

9. Conclusion

Our latest version of the XDoX system, using our new passage-clustering techniques, implements a reasonable approximation of conceptual clustering. The overall quality is significantly better than that of our previous version. The multi-document summaries are more readable and coherent. In most cases the system successfully presents main points, skips over minor details, and avoids redundancy. For small document sets, however, our clustering techniques are sometimes less effective. We need to continue to explore ways of generating longer, more detailed summaries for small groups of documents that have little repetition, such as documents having to do with unrelated examples of one type of event.

References

Fellbaum, C. (Editor). 1998. WordNet – An Electronic Lexical Database. MIT Press.

Firmin, T. and M. J. Chrzanowski. 1999. An Evaluation of Automatic Text Summarization Systems. In I. Mani and M. Maybury (eds) Advances in Automatic Text Summarization. The MIT Press.

Hearst, M. 1994. Multi-paragraph segmentation of expository text. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 9–16. Las Cruces, New Mexico: Association for Computational Linguistics.

McKeown, K., and Radev, D. 1995. Generating summaries of multiple news articles. In Proceedings, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 74–82. Seattle, Washington.

Miller, G.A. 1995. WordNet: A Lexical Database. *Communication of the ACM*, 38(11):39–41.

Mitra, M., A. Singhal, and C. Buckley. 1997. Automatic text summarization by paragraph extraction. In Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain.

Robertson, S. E., S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford (1995). Okapi at TREC-3. In Harman, D., editor, The Third Text Retrieval Conference (TREC-3), pp. 219–230. National Institute of Standards and Technology Special Publication 500–225.

Singhal, A., C. Buckley, M. Mitra (1996). Pivoted Document Length Normalization. SIGIR 1996: 21–29.

Stein, G., T. Strzalkowski and B. Wise (2000). Interactive, Text-Based Summarization of Multiple Documents. *Computational Intelligence*, vol. 16(4), pp. 606–613. Blackwell Publishers.

Strzalkowski, T., G. Stein, J. Wang and B. Wise. 1999. A Robust, Practical Text Summarizer. In I. Mani and M. Maybury (eds), *Advances in Automated Text Summarization*. MIT Press, pp. 137–154.

Willett, P. 1988. Recent trends in hierarchical document clustering: A critical review. Information Processing and Management, 24(5).