Computational Linguistics and Chinese Language Processing
Vol. 13, No. 2, June 2008, pp. 141-156

Multiple Document Summarization Using Principal Component Analysis Incorporating Semantic Vector Space Model

Om Vikas*, Akhil K Meshram*, Girraj Meena*, and Amit Gupta*

Abstract

Text Summarization is very effective in relevant assessment tasks. The Multiple Document Summarizer presents a novel approach to select sentences from documents according to several heuristic features. Summaries are generated modeling the set of documents as Semantic Vector Space Model (SVSM) and applying Principal Component Analysis (PCA) to extract topic features. Pure Statistical VSM assumes terms to be independent of each other and may result in inconsistent results. Vector space is enhanced semantically by modifying the weight of the word vector governed by Appearance and Disappearance (Action class) words. The knowledge base for Action words is maintained by classifying the words as Appearance or Disappearance with the help of Wordnet. The weights of the action words are modified in accordance with the Object list prepared by the collection of nouns corresponding to the action words. Summary thus generated provides more informative content as semantics of natural language has been taken into consideration.

Keywords: Principal Component Analysis (PCA), Semantic Vector Space Model (SVSM), Summarization, Topic Feature, Wordnet

1. Introduction

With the advent of the information revolution, electronic documents are becoming a principal media of business and academic information. The Internet is being populated with hundreds of thousands of electronic documents each day. In order to fully utilize these on-line documents effectively, it is crucial to be able to extract the main idea of these documents. Having a Text Summarization system would thus be immensely useful in serving this need. Multiple Document Summarization System aids to provide the summary of a document set that

*Indian Institute of Information Technology and Management, Gwalior, India- 474010
E-mail: {omvikas, akhil, girrajmeena, amitgupta}@iiitm.ac.in

[Received December 30, 2006; Revised July 14, 2007; Accepted July 19, 2008]

contains documents which belong to same topic. It can also be used to generate the summary of a single document.

In the present work, we propose a method of text summarization that uses semantics of data in order to form efficient and relevant summary. Summary is generated by constructing Statistical Vector Space Model **(3.1)** and then modifying it using the concept of Action words to form Semantic Vector Space Model **(3.2)**. Action Words are identified using the Action Word Classifier which makes use of Wordnet [Kedar *et al.*] in order to analyze the semantics of word.

Principal Component Analysis **(3.3)** is then applied on SVSM to reduce the dimension of multidimensional data sets. Singular Value Decomposition (SVD) is carried out on SVSM as a part of PCA to yield singular values and eigen vectors. Backprojection is then performed to project the documents onto the eigen space yielding projected values of documents which are henceforth compared with the singular values to yield the most relevant document/topic. Sentence Extraction **(3.4)** from multiple document sets has been assigned weight on the basis of keywords obtained from the most important document/topic. Sentences with higher weight are taken to form a summary.

2. Related Work

Various multiple document summarization systems already exist. This document summarizer is based on Kupeic 95 [Kupeic *et al.* 1995] which is a method of training a Bayesian classifier to recognize sentences that should belong in a summary. The classifier estimates the probability that a sentence belongs in a summary given a vector of features that are computed over the sentence. It identifies a set of features that correspond to the absence/presence of certain words or phrases and avoids the problem of having to analyze sentence structure. Their work focused on analyzing a single document at a time. Since then, there has been lot of work on the related problem of Multiple-document Summarization [Regina *et al.* 1999; Radev *et al.* 1998], where a system summarizes multiple documents on the same topic. For example, a system might summarize multiple news accounts of the recent massacre in Nepal; into a single document. Our hypothesis is that the similarities and differences between documents of the same type (*e.g.* bios of CS professors, earnings releases, etc.) provide information about the features that make a summary informative. The intuition is that the ‘information content’ of a document can be measured by the relationship between the document and a corpus of related documents. To be an informative summary, an abstract has to capture as much of the ‘information content’ as possible. To gain a handle on the problem of capturing the relationship between a document and a corpus, we examined several papers on Multiple-Document Summarization [Regina *et al.* 1999; Radev *et al.* 1998, 2000, 2004; Otterbacher *et al.* 2002]. However, we found most of their approaches were not applicable to our problem since they are mostly trying to match sentences of the same meaning to align multiple documents. The MEAD summarizer [Radev *et al.* 2000, 2001], which was developed at the University of Michigan and at the Johns Hopkins University 2001 Summer Workshop on Automatic Summarization, produces summaries of one or more source articles (or a 'cluster' of topically related articles).

Our Summarizer works on the documents belonging to same topic. It is strongly motivated by the analogy between this problem and the problem of face identification, where a system learns features for facial identification by applying PCA to find the characteristic eigenfaces [Turk *et al.* 1991; Pentland *et al.* 1994; Moon *et al.* 2001].

3. New Methodology

Any set of documents dealing with the same subject is decomposed using Vector Space model. The important keywords can be extracted from the Vector Space Model using a threshold. Such keywords are called thematic keywords which are based on statistics. Important sentences can be extracted and a summary can be made using thematic keywords. We propose a new methodology for multiple document summarization by enhancing the VSM using semantics and identifying topic features based keywords to make the multiple document summary. The approach is:

1. Statistical VSM construction from Multiple Document Set.
2. Semantic VSM generation using the concept of Contextual Action Words using Wordnet.
3. Application of PCA on Semantic VSM to reduce the dimension of the multidimensional data set yielding the most important Keywords.
4. Score Sentences based on several features such as sentence length cut-off feature, position feature, keyword weight, etc.
5. Generation of Summary extracting the sentences with high Score.

3.1 Statistical VSM Construction

The Multiple Document Summarizer models the set of documents related to the same topic as the Statistical Vector Space Model based on several heuristics. The simplest way to transform a document into a vector is to define each unique word as a feature. The weight of a feature being decided based on the contribution of various parameters such as Cue-phrase Keywords, topic keywords and term frequency in document. The weight of the feature is being termed as Feature Combination.

The vector representations of the documents; collectively define an n-dimensional vector space (where each document is an nx1 vector). The m document vectors taken as the columns of an nxm matrix D, define a linear transformation into the vector space.

3.2 Semantic VSM Construction

The existing vector space model is statistical in nature. This vector space is input to a number of tools and processes like a summarizer and information retrieval system. PCA/SVD Technique has been applied earlier for Summarization based on statistical vector space [Gong *et al*. 2001]. Some times this statistically generated model is unable to define the context. Keywords identified by a statistical model can be non-contextual in nature. Therefore, an effort is to be made in the direction of identification of contextual keywords and modification of existing model so that it can be more helpful and contextual for various applications like text summarization and text retrieval.

To identify the contextual keywords, we try to exploit human psychology. In any article, we identify that those words are important which either give a sense of either appearance or disappearance of any object/event. Thus, after we have the pure statistical vector space we need to enhance the vector space semantically by modifying the weights of the word vector by identifying the Appearance and Disappearance (ACTION class) words. To do so, we need to have a knowledgebase (KB) with some seed wordlist which belongs to appearance or disappearance. Following, are the steps involved in the semantic vector space model.

1. Get the tf matrix, T from existing document D.
2. Identify the set of action words, A from the given tf matrix, T, (number of action words =n).
3. Find the associated object list O_i for action word A_i; $A_i \varepsilon A$, $0<i<n$.
4. Find contextual objects Co from Object list O1,O2,….,On.
5. Modify weight of contextual objects in T to form semantic vector space S{T}.

3.2.1 Identification of Action words

Action words are the backbone of the semantic vector space model.

Definition: Action words are verbs that are used to strengthen the way experiences are presented whether it is expressing positive or negative experience.

With the help of Wordnet, the terms from the tf (term frequency) matrix which belong to the ACTION class can be easily classified. The algorithm uses a seed word list to identify the action words.

Definition: Seed Word List is the collection of action words. (Appendix A)

Whenever a term from the tf matrix is fetched, it is matched against seed word list. If it is matched, then the fetched term is action word; otherwise, synonyms of the fetched terms are matched.

```
Given Input: T = {t1,t2,…..,tn}.
List type: A = { }, integer type: depth
Do:   for every t ε T
          depth = 0;
          match(t,seedwrdlst)
          if found then A = A U t.
          else, not found
              if(depth == 0)
                  match(extractsynonym (t),seedwordlist)
                depth = 1
              else
                continue
            endif
        endif
      endfor
Output : A = {t1,t2,….tm}.
```

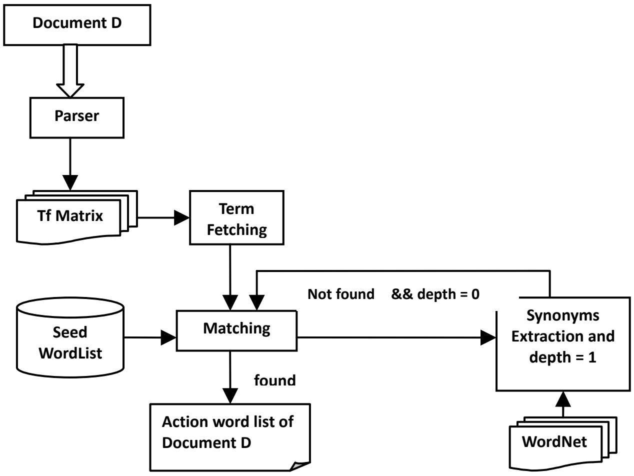

Figure 1. ACTION Word Classifier

To decide whether the word belongs to action list or not, we have to build a seed wordlist and compare them with standard meaning. For example, let '**devastation**' be the word to be decided as action or not. After searching in WordNet, the following meanings were obtained:

- **desolation** (an event that results in total destruction)

- **ravaging**, (plundering with excessive damage and destruction)
- **destruction,** (the termination of something by causing so much damage to it that it cannot be repaired or no longer exists)

From the first and last meaning, it clearly lies in the phenomenon of appear/disappear so it will be appended into the seed list along with its Synset.

3.2.2 Finding the Objects of the Action

Merely acquiring the ACTION words doesn't provide the semantic to the vector space. We have to find whether these words are really important. The importance of the word can be estimated by the application of the word in the article. Objects corresponding to the Action Words and their weight in Statistical VSM have to be identified in order to determine the extent of relevancy of Action Words. The Objects are the Nouns or Adjectives for the Action. The nearest Noun for Verb is identified using POS Tagger and termed as Object of Action. Only those sentences are to be chosen which contain action words.

Today *broke* fire in Delhi. (Action is verb)

Today/NN *broke*/VBD **fire**/NN in/IN Delhi/NNP

Destruction of material *happens* due to this fire.

Destruction//NN of/IN **material**/NN happens/VBZ due/JJ to/TO this/DT fire/NN ./.

Many suffered from the *broken* glass in the road. (Action is Adjective)

Many/JJ suffered/VBD from/IN the/DT *broken*/JJ **glass**/NN in/IN the/DT road/NN. /.

The authority *arrives* here soon.

The/DT **authority**/NN arrives/VBZ here/RB soon/RB. /.

Table 1. Action-Object List

Action word	**Objects**	
broke	fire, glass	
arrives	Authority	
destruction	material	

The bold ones are selected as objects for the Action. The action-object list is prepared by the help of POS tagger and Contextual Action Words are determined.

3.2.3 Classification of Contextual Words

Contextual words are being defined as those action words which are applied to the important object. The Weight of Action Word is being taken as the maximum weight amongst all the objects corresponding to the given Action Word. The weight obtained is added to the weight of the corresponding Action Words in Statistical VSM yielding Semantic Vector Space Model for the given set of Documents.

If we take an example of a single document:

Today broke fire in Delhi. Mass Destruction of material happens due to this fire. Many suffered from the broken glass in the road. The authority arrives here soon. Till now there is report of any casualties in these fire except from few injures. Thanks for the local communities for help.

the Vector Space generated on the basis of term frequency feature is

Table 2. Vector Space of Single Document (1 x 20)

Broke	0.1889	Injuries	0.1889
Fire	0.5669	Thanks	0.1889
Delhi	0.1889	Local	0.1889
Mass	0.1889	communities	0.1889
Destruction	0.1889	Help	0.1889
Material	0.1889	Authority	0.1889
Happens	0.1889	Arrives	0.1889
Suffered	0.1889	Report	0.1889
Broken	0.1889	Casualties	0.1889
glass	0.1889	Road	0.1889

The Action Word List obtained corresponding to the above example is: broke, destruction, and arrives. Now, the Action-Object list is prepared by identifying the Object words in which the ACTION words are acted.

Table 3. Object-Action List for example above

Action word	**Objects**
broke	today, fire, glass
destruction	Material
arrives	Authority

Each ACTION word has been given weight as per the contextual word obtained corresponding to it.

Table 4. Contextual Action List

Action word	**weight factor (wt)**
broke	Max (0.1889,0.5669) = 0.5669
destruction	0.1889
arrives	0.1889

The Statistical VSM is now modified and the Semantic VSM is being generated as follows

Table 5. Semantic Vector Space Model

Broke	***0.7558***	Injuries	0.1889
Fire	0.5669	Thanks	0.1889
Delhi	0.1889	Local	0.1889
Mass	0.1889	communities	0.1889
Destruction	***0.3778***	Help	0.1889
Material	0.1889	Authority	0.1889
Happens	0.1889	***Arrives***	***0.3778***
Suffered	0.1889	Report	0.1889
Broken	0.1889	Casualties	0.1889
glass	0.1889	Road	0.1889

Similarly, the model is extended for multiple documents. This Semantic Vector Space Model is used further to determine important Keywords and henceforth, the summary.

3.3 Principal Component Analysis

Principal Component Analysis (PCA) [Michael *et al.* 2003] is used to reduce the multidimensional datasets to lower dimensions for analysis. Singular Value Decomposition (SVD) [Michael *et al.* 2003] is carried out on Semantic VSM to find the principal components of Vector Space. The singular value decomposition (SVD) of matrix A_{mxn} is the factorization $A=U\sum V^T$, where U and V are orthogonal, and $\sum$= diag $(\sigma_{1,\ldots,}\sigma_r)$, r= min (m,n), with $\sigma_1 \geq \sigma_2 \geq \ldots\ldots \geq \sigma_r \geq 0$. The columns of V are the ‘hidden’ dimensions that we are looking for. The diagonal of $\sum$ are the singular values which are the weights for the new set of basis vectors. $\sum$ is symmetric, its singular values are its eigen values and its basis vectors are the eigen vectors.

Given an eigen vector e, we can find the corresponding dimension in document space.

$$\overrightarrow{d} = D.\overrightarrow{e}$$

After determining out the dimension of eigen vector in document space, backprojection of d^* is carried out. Commonly, composing a vector in terms of the principal components is called backprojection. Since our principal components or eigen documents are all orthogonal vectors, this is easy to accomplish. Let E be the matrix formed from the eigen documents then vector p is the document projected onto the eigenspace.

$$\vec{p} = E^T \vec{d}$$

Relevance of the topic/document is calculated by dividing projected component by the corresponding Singular Value. Metrics thus obtained is arranged in decreasing order excluding out the negative metrics. Main topic/Document is the one with highest metric value.

After selecting the main topic, we now need the topic keywords. We simply take the eigen document vector corresponding to main document and select the words with high weight. These are the set of Keywords which are of high relevance in summary.

3.4 Sentence Extraction

To identify sentences that should belong to summary, several features have been taken into consideration.

- **Sentence-Length Cut off Feature** – If the sentence length is greater than 4 words, only then it is taken into consideration.
- **Position Feature –** Sentences have been given some weight based on their position in the paragraph whether it is in initial, middle or final.
- **Keywords -** Sentence weight also depends not only on the number of keywords present in it but also the weight of each keyword.
- **Upper Case Feature -** Sentences containing upper case words have been given additional weight as it is probable that they may contain proper nouns.

Sentences with higher weight are taken as the relevant sentences for the summary and arranged in the order they appear in the document yielding the required summary. The rearrangement becomes a challenge in the case of multiple documents. In that case, sentences are kept at the position at which they appear in original document (initial/middle/final). This rearrangement technique provides fair results.

4. Implementation

The Multiple Document Summarization System is implemented in Java using JAMA (Java Matrix Package) and WVTool (Word Vector Tool) packages. JAMA is used to perform all the matrix operations as computing SVD, eigen vector, Backprojection, etc. WVTool is used to generate the Statistical Vector Space Model taking input as the Multiple Document Set or a Single document based on user requirement.

5. Evaluation of Summarizer

The present section will focus on the accuracy of the proposed summarization method. The accuracy of the method was examined on both single as well as multiple document summaries:

5.1 Single Document summary

Text belonging to different areas was taken. Summaries to the same texts were made by sentence extractions by different people. Based on the set of the summaries, we ranked sentences of the texts.

We then carried out the summarization process using our algorithm, the Auto Summarizer in MS Word, and the Gnome Summarizer and compared their agreement on the extracted sentences with the human sentence extractions.

The results are given in the following table.

Table 6. Summarization Algorithm Results

Article #	Our Summarizer	MS Word Summarizer	Gnome Summarizer
Science (789 words)	60.0%	50.0%	70.0%
Geography (725 words)	55.56%	33.33%	22.5%
History (557 words)	70.0%	50.0%	48.5%
Average accuracy	**61.85%**	**44.44%**	**47%**

On an average, we get an average accuracy of 61.85% and improvement of 39.17% with respect to MS Word Summarizer.

5.2 Multiple Documents Summary

The set of Documents belonging to “Introduction to Web crawler” were taken and then summary was generated using the proposed algorithm, and it was observed that the summary thus generated was in coherence with most of the documents. The input documents set consisting of documents related to the topic for summarization has been shown in Table 7.

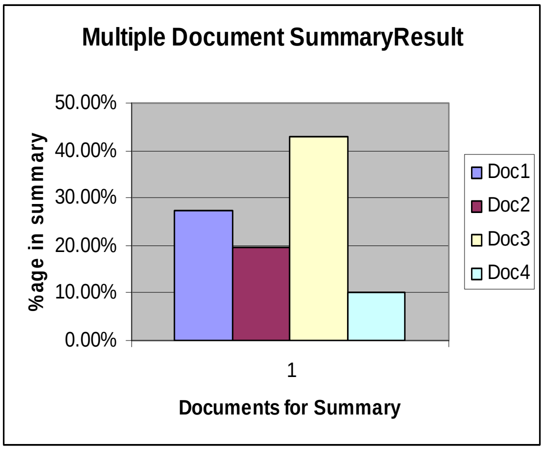

Figure 2. Contribution percentage

Table 7. Input Set for Multiple Documents Summary

Doc No.	Title of Doc	Doc Length
Doc1	Introduction to Crawler Architecture	1076 words
Doc2	Developing Web Search Engine	890 words
Doc3	Overview of Web Crawler	945 words
Doc4	Future of Search Engines	970 words

The cause of the low contribution of Doc4 to the summary generated was observed to be sentences with fewer keywords in them with respect to sentences from other documents, resulting in a low score of sentence.

6. Conclusion and Future Work

As seen from the results, the proposed method works better for various domains and, by using Semantic VSM instead of Statistical VSM; the summary obtained has become more informational and meaningful. Moreover, this method can be used to generate single as well as multiple document summaries.

The following areas in Multiple Document Summarization System require improvement:

1. Rearrangement of Extracted Sentences in the case of Multiple Documents Summarization to form an effective summary.
2. Enhance Flexibility of the system to generate a summary of multiple documents not necessarily belonging to the same topic.
3. Develop better methodology to incorporate the ACTION word score into Statistical VSM.
4. Evaluation of the system on large data samples.

References

Barzilay, R., “Information fusion in the context of multi-document summarization,” Phd. Thesis, Columbia University, 2003.

Bellare, K., A. Das Sarma, A. Das Sarma, N. Loiwal, V. Mehta, G. Ramakrishnan, and P. Bhattacharya, Generic Text Summarization using WordNet, http://i.stanford.edu/~anishds/publications/lrec04/lrec04.ps.

Berry, M. W., S. T. Dumais, and G. W. Obrien, “Using linear algebra for intelligent information-retrieval,” *Siam Review,* 37, 1995, pp. 573-95.

Golub, G., and C. Van Loan, *Matrix Computations*, Baltimore: Johns Hopkins Univ Press, 1996.

Gong, Y., and X. Liu, “Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis,” *SIGIR 2001*, pp. 19-25.

Jessup, E. R., and D. C. Sorensen, “A parallel algorithm for computing the singular-value decomposition of a matrix,” *Siam Journal on Matrix Analysis and Applications,* 15, 1994, pp. 530-548.

Jolliffe, I. T., *Principal Component Analysis*, New York: Springer, 1986.

Kupiec, J., J. Pedersen, and F. Chen, “A Trainable Document Summarizer,” In *Proceedings of the 18th ACM-SIGIR Conference*, 1995, pp. 68-73.

Moon, H., and P. J. Phillips, “Computational and Performance aspects of PCA-based Face Recognition Algorithms,” *Perception,* 30, 2001, pp. 303-321.

Otterbacher, J. C., A. J. Winkel, and D. R. Radev, The Michigan Single and Multidocument Summarizer for DUC 2002, http://www-nlpir.nist.gov/projects/duc/pubs/2002papers/umich_otter.pdf.

Pentland, A., B. Moghaddam, and T. Starner, “View-Based and Modular Eigenspaces for Face Recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 21-23 June, 1994, Seattle, Washington, USA, pp. 84-91.

Radev, D. R., and K. R. McKeown, ”Generating natural language summaries from multiple on-line sources,” *Computational Linguistics*, 24(3), 1998, pp. 469-500.

Radev, D. R., H. Jing, and M. Budzikowska, "Centroid- based summar-ization of multiple documents: sentence extraction, util-ity based evaluation, and user studies," In *ANLP /NAACL Workshop on Summarization*, Seattle, WA, April 2000.

Radev, D., S. Teufel, H. Saggion, W. Lam, J. Blitzer, A. Celebi, H. Qi, D. Liu, and E. Drabek, "Evaluation challenges in large-scale multidocument summarization: the MEAD project," Johns Hopkins University CLSP Workshop Final Report, 2001.

Radev, D. R., H. Jing, M. Sty, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing and Management,* 40, 2004, pp. 919-38.

Salton, G., *The Smart Retrieval System,* Prentice Hall, Englewood Cliffs, N.J. 1971.

Strang, G., *Introduction to Linear Algebra,* Wellesley, MA: Wellesley Cambridge Press, 1998.

Turk, M., and A. Pentland, "Eigenfaces for Face Detection/Recognition," *Journal of Cognitive Neuroscience,* 3(1), 1991, pp. 71-86.

Wall, M. E., A. Rechtsteinen, and L. M. Rocha, "Singular Value Decomposition and Principal Component Analysis," In A Practical Approach to Microarray Data Analysis (D.P. Berrar, W. Dubitzky, M. Granzow, eds.) Kluwer: Norwell, MA, 2003, pp. 91-109, LANL LA-UR-02-4001.

Appendix A: Seed Word List

Abstracted
Achieved
Acquired
Acted
Adapted
Addressed
Administered
Advertised
Advised
Advocated
Aided
Allocated
Analyzed
Answered
Anticipated
Applied
Appraised
Approved
Arranged
Ascertained
Assembled
Assessed
Assisted
Attained
Audited
Augmented
Authored
Bolstered
Briefed
Brought
Budgeted
Built
Calculated
Cared
Charged
Chartered
Checked
Clarified
Classified
Coached
Collaborated
Collected
Comforted
Communicate
Compared
Completed
Complied
Composed
Computed
Conceived
Conducted
Conserved
Consulted
Contracted
Contributed
Converted
Cooperated
Coordinated
Copied
Correlated
Counseled
Created
Critiqued
Cultivated
Dealt
Debated
Decided
Defined
Delegated
Delivered
Destruction
Designed
Detected
Determined
Developed
Devised
Diagnosed
Directed
Discovered
Discriminated
Dispatched
Displayed
Dissected
Documented
Drafted
Drove
Edited
Eliminated
Empathized
Enabled
Enforced
Enlightened
Enlisted
Ensured
Established
Estimated
Evaluated
Examined
Exceeded
Excelled
Expanded
Expedited
Experimented
Explained
Explored
Expressed
Extracted
Facilitate
Fashioned
Financed
Fixed
Followed
Formulated
Fostered
Founded
Gained
Gathered
Gave
Generated
Governed

Guided
Handled
Headed
Helped
Identified
Illustrated
Imagined
Implemented
Improved
Improvised
Inaugurated
Increased
Indexed
Indicated
Influenced
Initiated
Inspected
Instituted
Integrated
Interpreted
Interviewed
Introduced
Invented
Inventoried
Investigated
Judged
Kept
Launched
Learned
Lectured
Led
Lifted
Listened
Located
Logged
Made
Maintained
Managed
Manipulated
Mapped
Mastered
Maximized
Mediated
Memorized
Mentored
Met
Minimized
Modeled
Modified
Monitored
Narrated
Negotiated
Observed
Obtained
Offered
Operated
Ordered
Organized
Originated
Overcame
Oversaw
Participated
Perceived
Perfected
Performed
Persuaded
Planned
Practiced
Predicted
Prepared
Presented
Prioritized
Produced
Programmed
Projected
Promoted
Proposed
Protected
Proved
Provided
Publicized
Published
Purchased
Queried
Questioned
Raised
Ran
Ranked
Rationalized
Read
Reasoned
Recorded
Received
Reduced
Referred
Related
Relied
Reported
Researched
Responded
Restored
Revamped
Reviewed
Scanned
Scheduled
Schemed
Screened
Set goals
Shaped
Skilled
Solicited
Solved
Specialized
Spoke
Stimulated
Strategized
Streamlined
Strengthened
Stressed
Studied
Substantiated
Succeeded
Summarized
Synthesized

Supervised

Supported

Surveyed

Sustained

Symbolized

Tabulated

Talked

Taught

Theorized

Trained

Translated

Upgraded

Utilized

Validated

Verified

Visualized

Won

Wrote