# Document Map and WN-SUM: A New Framework for Automatic Text Summarization and a First Implementation[1]

Rakesh Verma and Felix Filozov[2]
Computer Science Department
University of Houston
Houston, TX, 77204, USA
http://www.cs.uh.edu

UH-CS-10-03
April 22, 2010

**Keywords**: Automatic summarization, extractive summarization, semantics-based approach

## Abstract

Automatic text summarization is a very challenging task whose goal is to compress text to its most important parts, therefore providing an indispensable aid for the user in dealing with a large body of text documents. Statistical methods of summarization incorporate minimum semantics and hence are suboptimal on summarization tasks. We propose a new, flexible and modular summarization framework called Document Map in this paper and a first implementation of it called WN-SUM. WN-SUM is a hybrid between semantic and statistical methodology. WordNet is employed by WN-SUM for semantic information. For statistical information, sentence position and topic relation are used. As used, WordNet makes a small improvement to WN-SUM. Nonetheless, the history of the Document Understanding Conferences (DUC) shows that even small improvements in this area are hard to achieve and can be significant. If WordNet is included, WN-SUM provides a Ngram(1,1) score of 0.45329, while WN-SUM without WordNet yields a score of 0.44312. WN-SUM and MEADDemo (another text summarization system) were tested against each other and scored Ngram(1,1) of 0.45329 and 0.44034, respectively. Both scored higher than the baseline, 0.42774.

[1] Thanks to NSF grants CCF 0306475, OCI-0453498 and CNS 0755500
[2] Felix Filozov is with the Polytechnic University of New York University, Brooklyn, NY 11201.

# Document Map and WN-SUM: A New Framework for Automatic Text Summarization and a First Implementation*

Rakesh M. Verma and Felix Filozov
Computer Science Department
University of Houston, Houston, TX 77204
rverma@uh.edu
713-743-3348/Fax: 713-743-3335

**Abstract**

Automatic text summarization is a very challenging task whose goal is to compress text to its most important parts, therefore providing an indispensable aid for the user in dealing with a large body of text documents. Statistical methods of summarization incorporate minimum semantics and hence are suboptimal on summarization tasks. We propose a new, flexible and modular summarization framework called Document Map in this paper and a first implementation of it called WN-SUM. WN-SUM is a hybrid between semantic and statistical methodology. WordNet is employed by WN-SUM for semantic information. For statistical information, sentence position and topic relation are used. As used, WordNet makes a small improvement to WN-SUM. Nonetheless, the history of the Document Understanding Conferences (DUC) shows that even small improvements in this area are hard to achieve and can be significant. If WordNet is included, WN-SUM provides a Ngram(1,1) score of 0.45329, while WN-SUM without WordNet yields a score of 0.44312. WN-SUM and MEADDemo (another text summarization system) were tested against each other and scored Ngram(1,1) of 0.45329 and 0.44034, respectively. Both scored higher than the baseline, 0.42774.

## 1 Introduction

The explosive growth of the world wide web and the increase in web "authors" have led to a growing need for people to deal with an overwhelming number of text documents on a daily basis. Examples of text documents include newspaper articles, discussion forums posts, research papers, etc. Thus automatic text summarization will increasingly be an indispensable aid in the future. The goal of automatic text summarization, which is a very challenging task, is to condense text documents into their essence, and

*Research supported in part by NSF grants CCF-0306475, OCI-0453498 and CNS-0755500

display the result to the user. The consequence of achieving this goal would be the ability for the users to effectively manage more textual data in less or equal time.

Automatic text summarization methods are of two types: abstractive and extractive. Abstractive methods may construct a summary using sentences that are not necessarily in the original document or some other abstract representation of the document, e.g., [4]. Extractive methods on the other hand rely only on sentences in the original document. In this paper, the focus is on extractive methods so henceforth we will not refer to any abstractive methods from the literature and whenever we speak of a summarization method the term extractive is implicit.

### 1.1 Related Work

Many automatic text summarization systems employ a vast array of statistical methods, e.g., see [1, 5, 9] and the references cited therein. These methods usually treat text documents as a bag of words with no order, or meaning. Using this idea, many systems were developed to be modestly successful. However, a sentence is more than just a collection of unordered words. Each sentence carries meaning, and a truly good summary can be constructed only if meaning is incorporated into the system. Recently, some research groups have started experimenting with incorporating some semantics into their systems, e.g., see the proceedings of the Document Understanding Conference for the last three years [10, 11, 12] at `http://www-nlpir.nist.gov/projects/duc/pubs.html`. For instance, WordNet has been used to build lexical chains of word synonyms for sentence filtering. We use WordNet in a novel way for sentence filtering in our framework called Document Map.

The rest of the paper is organized as follows. Section 2 presents the summarization framework, Document Map. Section 3 gives an overview of WN-SUM, a first implementation of Document Map followed by a section for each component of the scoring algorithm used in WN-SUM. Section 7 gives the scoring formulas. Section 8 presents some evaluation results and Section 9 concludes the paper.

## 2 Document Map

We first describe our framework for single document summarization, later we will extend it to multiple documents. Given a compression ratio for the summary defined as the size of the desired summary (in words or sentences) divided by the size of the text document in the same unit, our framework for summarization, Document Map, envisions document summarization as a task consisting of two stages. In the first stage, sentences in a document are classified using a scoring algorithm into two groups: the thematic sentences and the evidentiary sentences. We refer to sentences that assert or restate the main themes or topics of the document as *thematic sentences*, and the sentences that modify or provide evidence in favor of (or against) the themes or topics of the document as *evidentiary sentences*. In the second stage, we must choose a subset (possibly all) of the thematic sentences and a corresponding subset (possibly empty) of the evidentiary sentences to construct a summary of the document. The second stage proceeds by marking the positions of the thematic sentences in the text, hence the name Document Map. One of our hypotheses in Document Map is that the evidentiary sentences in between thematic sentence $i$ and thematic sentence $i+1$ in the original text provide evidence for (or against) the $i^{th}$ thematic sentence. Henceforth we refer to this group of evidentiary sentences as the *evidentiary region for thematic sentence* $i$.

The reason for not necessarily selecting all the thematic sentences is that there may be too much redundancy if all of the thematic sentences are selected. The idea is to choose enough thematic sentences, subject to the compression ratio, to provide good coverage of the themes of the article. If some space remains based on the compression ratio, then for the remaining space it is better to select evidentiary sentences rather than redundant thematic ones. The idea is to provide new information to the user not available in the thematic sentences alone and to improve the coherence and readability of the summary generated. Again in choosing the evidentiary sentences we choose the ones that are not subsumed by the thematic sentences and are the top scoring sentences in the evidentiary region of a thematic sentence. In Document Map, top scoring evidentiary sentences are selected starting from the evidentiary region of the top scoring thematic sentence and so on until there is no more remaining space for the summary.

To implement the above summarization framework, one needs to specify: the algorithm that identifies themes of the article, the sentence scoring algorithm and the coverage or the redundancy checking algorithm.

There is a fair body of work on identifying the themes or topics of an article and so there are a number of options available. We currently use the TextRank algorithm [9] for identifying the topics, which is easy to implement and appears to be robust (see the experiments on this problem in [9]). However, if a careful comparison with other algorithms is done and a different winner is found, our framework is modular and can use a different algorithm painlessly. In Document Map our hypothesis is that sentences matching the topics of the article and containing more general words are more likely to be thematic sentences. Hence we propose to use named entity extraction algorithms to identify sentences that contain too specific information and we use WordNet in an attempt to quantify the generality of words in a sentence.

A possible coverage algorithm could be based on a syntactic or semantic subsumption checking algorithm. We say that sentence $s$ *subsumes* sentence $t$ if the fraction of information of sentence $t$ present in sentence $s$ is $\geq 1/2+\epsilon$, where $\epsilon > 0$ and the fraction of information of sentence $s$ present in sentence $t$ is $\leq 1/2-\epsilon$. Of course, the fraction $1/2$ just represents a possible choice out of a range of possible choices in the half-open interval $[1/2, 1)$. The measure of information could be purely syntactic, e.g., the number of keywords in a sentence or it could incorporate semantic information as well such as through word sense disambiguation. Further, we could include or exclude named entities from the information measure. We leave these two choices open in Document Map.

Given a subsumption checking algorithm we construct a graph in which the nodes represent thematic sentences and a directed edge from vertex $u$ to vertex $v$ represents that $u$ subsumes $v$. Then, we run a greedy approximation algorithm for vertex cover, which is well known to be NP-complete but has an efficient 2-approximation algorithm [2], in such a directed graph. Before we discuss the sentence scoring algorithm we describe how our summarization framework can be applied to multiple documents.

**Multiple Document Summarization.** Our framework can be extended to multiple document summarization. The best way would be to apply the framework to documents individually first. We do not recommend concatenating the documents or even fusing/splicing them together. The reason is that often sentence scoring algorithms use positional information and this information is not preserved under document concatenation or splicing. Next, a subset of the overall highest scoring thematic sentences can be chosen based on the compression ratio and the coverage of themes. If space permits, we then choose the highest scoring evidentiary sentences for the thematic sentences selected from the evidentiary regions of the thematic sentences. Of course, we start with the highest scoring thematic sentence first.

We discuss the sentence scoring algorithm as part of the system we have implemented called WN-SUM. WN-SUM, described below is our first realization of this vision. In this realization, we have focused on the classification stage. The second stage is left for future research.

WN-SUM uses a hybrid model of statistics and semantics. Some statistical methods are used to decide if a sentence belongs to a summary. The first such method is sentence position. It is known that sentences in the beginning and end make a great summary (depending on the document type). Second, sentences which include important keywords are considered good summary candidates as well. Semantics are incorporated into WN-SUM by using WordNet. WordNet is used to establish how much more general a word is compared to another word. The hypothesis is that, all other scores being equal, a sentence with more general words than another sentence is considered more likely to be a thematic sentence.

# 3 System Overview

WN-SUM is made of two major components, represented in Figure 1. The left column in Figure 1 calculates a score for each sentence. Sentences with higher scores are considered better summary candidates. Each of these scores depends on the output of the second component, represented by the right column in Figure 1.

Since the first five steps of WN-SUM are the same, the following descriptions will apply to both components. The input text document is accepted as input. The document is then separated into individual sentences. Then, each named entity is marked so it can be distinguished from non-named entities. All the sentences are then passed to POS tagger, which adds a part-of-speech tag to each word. The result is then passed to a tool that disambiguates the meaning of each word. From what has been processed so far, the step "Position Score & WordNet Score" calculates position score $PS$ and WordNet score $WNS$ for each sentence (more details will be given later in the paper). Using the output of the TextRank algorithm, the topic score $TopS$ is calculated. Then, combining all three scores yields a total score $TS$. It is this score that decides if a sentence belongs to a summary. The final step is to choose $k$ highest scores and the corresponding sentences to be summary representatives.

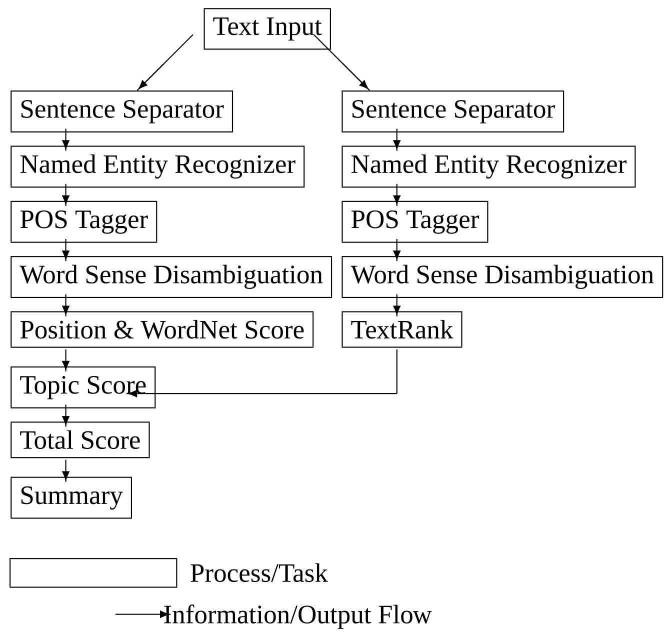



Figure 1: Components of WN-SUM

The tools used to perform named entity recognition, POS tagging, and word sense disambiguation are described below.

## 3.1 Named Entity Recognizer

For named entity recognition, WN-SUM uses Stanford Named Entity Recognizer 1.0. It is able to extract three classes: PERSON, LOCATION, and ORGANIZATION. According to the website: “The software provides an implementation of Conditional Random Field sequence models, of the sort pioneered by Lafferty, McCallum, and Pereira (2001), coupled with well-engineered feature extractors for Named Entity Recognition. Included are a good 3 class (PERSON, ORGANIZATION, LOCATION) recognizer (in versions with and without additional distributional similarity features) and another pair of models trained on the CoNLL 2003 English training data. The distributional similarity features improve performance but the models require considerably more memory.”

## 3.2 POS Tagger

For part-of-speech tagging, WM-SUM uses Stanford POS tagger 2006-05-21. This software is a Java implementation of the log-linear part-of-speech (POS) taggers described in [14]. The tagger was successfully used with word sense disambiguation tool.

### 3.3 Word Sense Disambiguation

For word sense disambiguation, WN-SUM uses SenseLearner 2.0 [8]. SenseLearner was trained using SemCor 2.1 database, which was compiled using WordNet 2.1. The sense tag of the word is used in locating the word in the WordNet hypernymy tree, which is explained below.

## 4 WordNet

WordNet is utilized to decide which sentences are more general, with the assumption that more general sentences are more likely to be thematic sentences. According to Fellbaum [3] "WordNet is neither a traditional dictionary nor a traditional thesaurus but combines features of both types. It resembles a thesaurus in that its building block is a synset consisting of all the words that express a given concept." According to Miller [3], "The basic semantic relation in WordNet is synonymy. Sets of synonyms (synsets) form the basic building blocks. Although synonymy is a semantic relation between word forms, the semantic relation that is the most important in organizing nouns is a relation between lexicalized concepts. It is relation of subordination (or class inclusion or subsumption), which is called hyponymy." Miller writes in [3] that (page 26): "Since a noun usually has a single hypernym, lexicographers include it in the definition." The key point to be noted is that although the hypernymy relation is defined on synsets in WordNet, and hence it could happen that a synset can have more than one hypernym, this situation is not frequent.[1] The reason is that a synset is designed to refer to a single concept and hence we need to disambiguate words in the document to find the correct synset for a noun. For instance the word plant could mean a factory in one context and could mean a tree in another context. Hence the word plant would be found in two different synsets in this case. The relation between nouns to other nouns, and verbs to other verbs is used by WN-SUM.

We use the hypernymy relation between nouns, which is defined as follows: A is a *hypernym* of B if the meaning of A encompasses the meaning of B (B is called the *hyponym*). For example, *animal* is a hypernym of *dog*, and a *dog* is a hyponym of *animal*. All nouns in WordNet are stored in a graph (that is close to a tree) that represents the hypernymy hierarchy. The word *entity* is the root of the tree, because it is believed to encompass the meaning of all other nouns. Traversing down the tree manifests more specific nouns. Figure 2 shows a very small portion of the hypernymy tree.

[1] We do take care of the situation in which there are multiple hypernyms as explained in the WordNet score subsection.

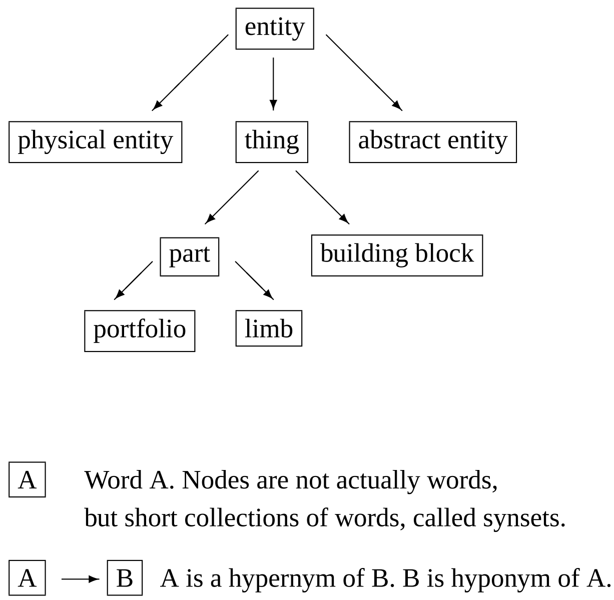



Figure 2: A tiny sample of WordNet hypernymy relation.

From observing the tree, it can be seen that more specific nouns are closer to leaves, and more general nouns are closer to the root. It is assumed that sentences that have nouns closer to the root are more likely to be thematic sentences.

WordNet also stores hypernymy relationships between verbs. The graph for verbs is not an almost tree. Within the graph there might be multiple root nodes. The assumption is, again, that a verb closer to any of these roots is more general as compared to a verb closer to any of the leaves. Therefore, sentences that have verbs closer to a root are considered more likely to be thematic sentences.

# 5 Position

It has been observed that sentences in certain positions of a document are good summary candidates as well as more likely to be thematic sentences. For example, the first and last sentences of a news article are good summary candidates. The same can not be said about other types of text documents. Currently, WN-SUM makes an assumption that the input document is similar to news articles. However, it can be modified to work with other types of text documents where candidate sentences are not at the beginning or end of the document.

# 6 Topic Relevance

Sentences which contain important keywords are considered good candidates. This is considered to be the case, because important keywords could be representing ideas that are often mentioned in the text and are likely to be related to the article's themes or topics. WN-SUM uses the TextRank algorithm to extract important keywords from a text document, and also the corresponding weight of importance. A word that has 0.1 importance is considered less important than a word with 0.9 importance. TextRank is based on PageRank. It builds a graph, where each node is a word and the edge between words represents a certain relationship. From this graph the importance of each word is inferred. This is similar to PageRank, except the nodes are web pages and the edges links between the web pages.

Before running TextRank on a text document, WN-SUM strips the document of unnecessary words. The only words left alone are nouns, verbs, and adjectives (stop words are removed). With the available text, TextRank builds a graph of words and their relationship to other words. The relation is defined as follows: if two words are separated by less then N words, then an edge is created between them. For example, if the text document is "A B C D E F," and N is 2, then A and C will have an edge between them, but A and F will not, and neither will A and D.

# 7 Sentence Selection

WN-SUM calculates a total score $TS$ for each sentence. Higher scores are assumed to indicate higher likelihood of a sentence being a thematic sentence. Hence this score quantifies how good of a summary candidate it is compared to other sentences. The total score $TS$ is a linear combination of three other scores: $WNS$, $PS$, and topic score $TopS$.

## 7.1 Position Score

WN-SUM uses a model where sentences at the beginning and sentences at the end are considered good summaries. Therefore, it is desirable for each sentence to have a score in [0,1], where beginning sentences have a score close to 1, middle sentences close to 0.5, and the last sentences close to 1. The following formula provides this model:

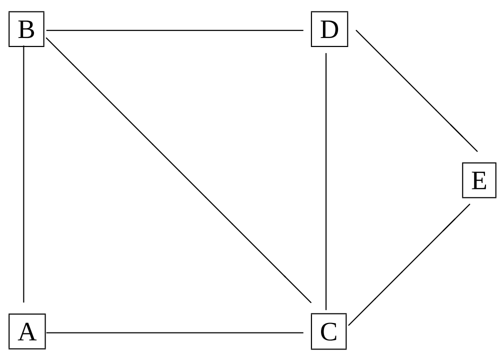
$$PS(S_i) = \frac{cos(\frac{2\pi x}{k-1}) + \alpha - 1}{\alpha}$$

$\alpha$ is the *dent factor*, and $\alpha \geq 2$. As $\alpha$ increases $PS$ of each sentence becomes more equally distributed. As $\alpha$ decreases, $PS$ is more concentrated at the beginning and end of the document. In the formula $k$ is the number of sentences in the text document, and $x$ is the position of sentence $S_i$ in the document, so the first sentence will have an $x$ of 0 and the last sentence will have an $x$ of $k - 1$.

The position score $PS$ is normalized by dividing each $PS$ value by the maximum $PS$ value.

## 7.2 Topic Relevance Score

As explained above, sentences which contain important keywords are considered good summary candidates. These sentences are said to be relevant to the topic of the document. The extraction of important keywords is facilitated by the TextRank algorithm. TextRank algorithm is related to PageRank algorithm. TextRank works by building a graph where nodes are words and edges are relations between words. If two words are separated by less then N number of words, then an edge is drawn between them. For example, suppose the text document input is “A B C D E,” and N is 2, then the graph generated by TextRank is as follows:




C has the most number of edges incident to it, therefore it will have the highest importance. TextRank uses the following formula to calculate the importance of each node (word):

$$I(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} * I(V_j)$$

$In(V_i)$ is the number of outgoing links from node $V_i$. $Out(V_j)$ is the number of outgoing links from node $V_j$. $d$ is the *damping factor* and is usually 0.85. Introducing the damping factor simulates a random jump from one node to another. This is more intuitively understood when dealing with actual web pages.

Topic score $TopS$ is the score given to each sentence depending on how relevant it is to the topic. It is computed as follows:

$$TopS(S_i) = \frac{\sum_{w \in Nouns(S_i) \cup Adjectives(S_i)} I(w)}{|S_i|}$$

$TopS$ is normalized by dividing each $TopS$ by the maximum $TopS$.

## 7.3 WordNet Score

The calculation of $WNS$ is divided into two parts. First, $NounScore$ is computed, then $VerbScore$.

$NounScore$ demonstrates where in the hierarchy tree are nouns in a sentence located. $VerbScore$ shows where in the hypernymy graph are verbs located.

First, to compute $NounScore$ the following formula is used:

$$NounScore(S_i) = \sum_{w \in Nouns(S_i)} \frac{D_n(w)}{LN_{max}}$$

$Nouns(S_i)$ is the set of all nouns of sentence $S_i$. $D_n(w)$ is the distance between the root node (*entity*), and word $w$. $LN_{max}$ is the *maximum path length* in the noun hypernymy tree. Note the use of the maximum to handle the situation of multiple hypernyms and consequently multiple paths to the root node. Second, to compute $VerbScore$ the following formula is used:

$$VerbScore(S_i) = \sum_{w \in Verbs(S_i)} \frac{D_v(w)}{LV_{max}}$$

$Verbs(S_i)$ is the set of all verbs of sentence $S_i$. $D_v(w)$ is the distance between the closest root node, and word $w$. $LV_{max}$ is the maximum path length in the verb hypernymy graph. $WNS$ is the length normalized average of $NounScore$ and $VerbScore$. The result is as follows:

$$WNS(S_i) = 1 - \frac{VerbScore + NounScore}{(|Nouns(S_i)| + |Verbs(S_i)|)^2}$$

$WNS$ is normalized by dividing each $WNS$ by the maximum $WNS$. Named entities receive a $NounScore$ of 1, indicating that they are the bottom of the noun hypernymy tree. The increase in the $NounScore$ later decreases $WNS$ when subtraction from 1 is done.

## 7.4 Total Score

Finally, a single score is assigned to each sentence, which is a linear combination of $PS$, $TopS$, and $WNS$. The following formula expresses this idea:

$$TS(S_i) = w_1 PS(S_i) + w_2 TopS(S_i) + w_3 WNS(S_i)$$

## 7.5 Choosing the weights

We took a small sample of about 35 documents from the DUC 2002 collection and generated summaries for these documents using 21 different combinations of the weights $w_1, w_2$, and $w_3$ subject to the constraint $w_1 + w_2 + w_3 = 1$ and $w_i \in [0, 1]$ with a step size of 0.2. These summaries were then evaluated using ROUGE 1.5 [6]. Based on the Rouge-1 average F-score the top five combinations in nonincreasing order of score were: 020008, 080002, 040006, 060004 and 100000, where axbycz means $w_1 = a.x$, $w_2 = b.c$ and $w_3 = c.z$. A surprising result of this experiment is that the topic score weight is found to be 0 for the top five combinations. The top ten combinations were then used on a larger sample of 533 documents from the DUC 2002 collection.[2] Results of this larger experiment are described below.

## 8 Evaluation

ROUGE 1.5 [6] was used to evaluate WN-SUM against other standards. According to a study reported in [7], Ngram(1,1) correlates highly with human subjects. Hence it was employed as a relative score. All systems were made to produce a summary of approximately 100 words by adjusting the corresponding compression rates. Four systems are evaluated:

- WN-SUM - system presented by this paper with $w_1 = 0.2, w_2 = 0.4$, and $w_3 = 0.4$.
- WN-SUM(Without WordNet) - system presented by the paper, where $WNS$ is ignored, and only $TopS$ and $PS$ are considered. As a result,

  $$TS(S_i) = w_1 PS(S_i) + w_2 TopS(S_i),$$

  where $w_1$ is 0.4, and $w_2$ is 0.6.
- MEAD - A demo on the web of the MEAD system [13].
- Random Baseline - A program that traverses through the text document and picks a word with probability 0.5. The traversal is done sentence by sentence, starting from sentence 0. If 100 words are collected without reaching the end, the program terminates anyway.

Data Set - A subset of DUC 2002 data set was used in the evaluation. 19 articles were extracted, along with their 19 human written abstracts. The following table (Table 1) shows Ngram(1,1) results of WN-SUM, MEAD [13], and the random baseline.

It should be noted that the human generated summaries that the systems were compared against using Rouge are not extractive. Humans tend to use their own words to summarize the documents. This is the reason why the correlations reported might appear low to some readers. Further, the history of the Document Understanding Conferences (DUC) shows that even small improvements in the automatic summarization area are hard to achieve and can be significant.

| WN-SUM | WN-SUM(- WordNet) | MEAD | Baseline |
|---|---|---|---|
| 0.4533 | 0.4431 | 0.4403 | 0.4277 |

Table 1: Ngram(1,1) results for evaluated systems.

[2] Although the DUC 2002 collection is purported to have 567 documents there are 34 pairs of duplicate documents, hence we get only 533 unique documents.

For the larger set of 533 documents from the DUC 2002 collection we found that the top 3 combinations (from the set of top 10 combinations of weights we found in the smaller experiment) were 040006, 060004, and 040204 based on the Rouge-1 average F-score. Again the top two have a topic score weight of 0, but the third has a small positive weight of 0.2 for the topic score.

## 9 Conclusion

In this paper we have presented a new, flexible and modular summarization framework called Document Map and a first implementation called WN-SUM. As can be seen from the evaluation, WordNet has made a positive contribution to the summarization algorithm although not perhaps not as large as we had hoped for. Several possibilities may be responsible. It is possible that WordNet scores by themselves do not correlate with human judges. A preliminary test shows that this suspicion may be true, but more experiments are required. The fact that verb hypernymy is not structured in a tree poses a problem, because the notion of generality between two words is no longer clearly defined as it is within the noun hypernymy tree. Finally, a third possibility is that the method used to compute the WordNet scores is not the best possible. Again, more research is needed on this issue.

A bigger dataset and more tests would be helpful in further evaluating the system. Hopefully, multiple sources of tests will converge on one result.

It is possible to learn the weights in $TS$ by training it on a corpus, and choosing those weights that minimize a certain function. This is another possible direction for future research.

In conclusion it is clear that further study must be done to show how semantic networks, such as WordNet, can facilitate better extraction of thematic sentences for summarization. A complete implementation of our framework is an ongoing goal of our work. Note that WN-sum implements a sentence scoring algorithm, but does not compute the subsumption relation between sentences and the region idea of Document Map. The implementation of these ideas is left for the future.

## References


[1] Ping Chen and Rakesh M. Verma. A query-based medical information summarization system using ontology knowledge. In *CBMS: IEEE Symp. on Computer-based Medical Systems*, pages 37–42, 2006.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms (Second Edition)*. MIT Press/McGraw-Hill Book Co., 2001.

[3] Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database*. MIT Press, 1998.

[4] M. Fiszman, T. Rindflesch, and H. Kilicoglu. Abstraction summarization for managing the biomedical research literature. In *HLT-NAACL 2004: computational lexical semantic workshop*, 2004.

[5] Udo Hahn and Inderjeet Mani. The challenges of automatic summarization. *IEEE Computer*, 33(11):29–36, 2000.

[6] Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, 2004.

[7] Chin-Yew Lin and Eduard H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *HLT-NAACL*, 2003.

[8] Rada Mihalcea and Andras Csomai. Senselearner: Word sense disambiguation for all words in unrestricted text. In *ACL*, 2005.

[9] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. In *EMNLP 2004: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2004.

[10] NIST, editor. *DUC '05: Proceedings of the Document Understanding Conference*, 2005.

[11] NIST, editor. *DUC '06: Proceedings of the Document Understanding Conference*, 2006.

[12] NIST, editor. *DUC '07: Proceedings of the Document Understanding Conference*, 2007.

[13] Dragomir Radev, Jahna Otterbacher, Hong Qi, and Daniel Tam. Mead reducs: Michigan at duc 2003. In *Proceedings of the Third Document Understanding Conference (DUC '03)*, 2003.

[14] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*, 2003.