# Combining Syntax and Semantics for Automatic Extractive Single-Document Summarization

Araly Barrera and Rakesh Verma

University of Houston, Houston TX USA,
Computer Science Department
abarrera7@uh.edu, rmverma@cs.uh.edu

**Abstract.** The goal of automated summarization is to tackle the "information overload" problem by extracting and perhaps compressing the most important content of a document. Due to the difficulty that single-document summarization has in beating a standard baseline, especially for news articles, most efforts are currently focused on multi-document summarization. The goal of this study is to reconsider the importance of single-document summarization by introducing a new approach and its implementation. This approach essentially combines syntactic, semantic, and statistical methodologies, and reflects psychological findings that pinpoint specific selection patterns as humans construct summaries. Successful summary evaluation results and baseline out-performance are demonstrated when our system is executed on two separate datasets: the Document Understanding Conference (DUC) 2002 data set and a scientific magazine article set. These results have implications not only for extractive and abstractive single-document summarization, but could also be leveraged in multi-document summarization.

## 1 Introduction

The Internet age brings forth an alarming rate of text documents (from news articles to electronic books to scientific papers, etc.), making it difficult for people to cope. Since as early as the 1950s, automated summarization of documents has been studied in an effort to alleviate an information overload problem considered to exist even then. The goal of this area of research is simply to reduce the vast amounts of information into compact summaries so that users can locate the most important pieces of information more easily from the "haystack."

Two main methods of summarization are [14]: 1) *abstractive* - the construction of original sentences from one's own thoughts, understanding, and experiences, and 2) *extractive* - the selection of most salient source sentences. Due to the complex linguistic and real-world knowledge required for truly abstractive summaries, extractive summarization has become a more popular choice for computation and is the focus of this study.

Although summarization has been studied for almost 50 years now, there has been a decline in recent research on single document summarization. In 2001-02 the Document Understanding Conference[1] (DUC) proposed the task of creating

[1] http://duc.nist.gov

A. Gelbukh (Ed.): CICLing 2012, Part II, LNCS 7182, pp. 366–377, 2012.

100-word summaries of individual news articles but soon after dropped single document summarization competitions to move on to multi-document extraction and update summarization. This, according to [21,18], was due to the fact that no system [in DUC 2001-2002] could outperform the baseline with statistical significance. The baseline, an extract consisting of the first portions of a document, has been generally accepted as a good representation of a news-article summary. Outperforming baseline standards essentially indicates a summarizer of high-quality, but for many researchers, the notion of single-document summarization remains that of an underperformer and essentially a more difficult task than multi-doc summarization [21,17].

In this work we revisit the important problem of single-document summarization and reconsider the performance of the baseline in a different context, viz., scientific magazine articles. We design a new and robust approach for single document summarization that ranks an article's sentences based on semantics, overall word popularity, and sentence position. We subject it to intensive experiments using two datasets: scientific magazine articles, and the DUC 2002 news collection from NIST. We compare our approach and its implementation against the baseline(s), the popular MEAD summarizer available on the internet [19], TextRank sentence extraction [16], and, for news data, the systems that participated in the DUC 2002 competition. We show that: (i) our system outperforms all the systems including the baselines, and (ii) for scientific article dataset, our system beats the baselines by a wide, statistically significant margin. For news articles, our system beats the baseline, but not by a statistically significant margin. Hence, our results also demonstrate that the baseline's presumed superiority *so far* only holds for news data.

The organization of the rest of this paper is as follows. Section 2 presents our system and its implementation and Section 3 describes the data sets used for system trials. Section 4 provides the evaluation methodology, Section 5 the results and Section 6 some perspective on the results. Section 7 discusses the related work and Section 8 concludes the paper.

## 2 Method and System Overview

As a whole, our system is designed to handle both syntactic and semantic qualities of a document's text. It implements part-of-speech (POS) tagging[2], named entity recognition[3], stopword removal[4], TextRank word extraction[16] for word popularity ranking, SenseLearner[5] for word disambiguation, a parser for heading recognition and filtering [6] and the popular WordNet [6] database tool for deeper word analysis. Figure 1 illustrates the entire process of our system.

[2] Stanford POS Tagger: `http://nlp.stanford.edu/software/tagger.shtml`

[3] Stanford NER Tagger: `http://wwwnlp.stanford.edu/software/CRF-NER.shtml`

[4] `http://search.cpan.org/creamyg/Lingua-StopWords-0.09/lib/Lingua/StopWords.pm`

[5] R. Mihalcea and A. Csomani. SenseLearner: Word Sense Disambiguation for all Words in Unrestricted Text. ACL, 2005.

[6] Link Grammar Parser: `http://www.link.cs.cmu.edu/link/`

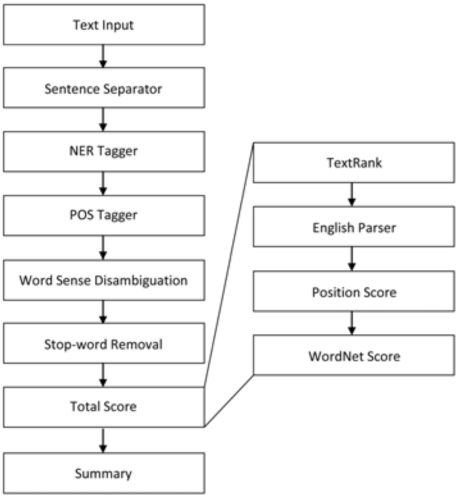

**Fig. 1.** Block Diagram of our system

The focus of this section is on our system's sentence scoring algorithm, which has a major influence on the extraction of a document's sentences for the construction of a summary. This method consists of assigning a score to each sentence that is the aggregate of three key weighted scores: 1) A **TextRank score**, based on the TextRank keyword extraction algorithm [16] to rank popularity of words within a document and to exploit the presence of these words in document sentences. 2) A **WordNet score** utilizing the WordNet [6] lexical database in three different models proposed for semantic prioritization. 3) A **Position score** which exploits a sentence's relative position within the text, a feature that humans naturally use for extraction.

### 2.1 Total Sentence Score

A final sentence score is assigned to each sentence as a linear combination of the Position ($P$), WordNet ($WN$), and TextRank ($T$) scores using the equation:

$$TotalScore(S_i) = w_1 P(S_i) + w_2 WN(S_i) + w_3 T(S_i) \tag{1}$$

where $w_1 + w_2 + w_3 = 1$. Essentially, resulting top scoring sentences are selected and used as the document's final summary, whose size would depend on a compression rate constraint[7] specified for the task at hand.

[7] See Section 5 for the different compression rates used for the datasets analyzed in this study.

### 2.2 TextRank Score

Our system implements the TextRank algorithm [16] to extract important keywords from a text document and also to determine the word's weight of importance within the entire document. The TextRank keyword extraction algorithm is a graph-based ranking model for graphs generated from text and is primarily based on PageRank [4]. Those words containing most co-occurring connections to other words in a graph are thus ranked with greater weights and thus considered most popular. Optimal results have been found when using only nouns and adjectives in this implementation. *The primary purpose of using this function is by giving higher weight to sentences containing a larger quantity of these popularly-used words as a means of selecting more thematic information.* The TextRank score ($T$) as mentioned in equation (1) used in our system for a sentence $S_i$, which is a multiset (or bag) of words, $w$, is the following:

$$T(S_i) = \frac{\sum_{w \in Nouns(S_i) \cup Adjectives(S_i)} I(w)}{|S_i|} \tag{2}$$

where *I(w)* computes the word's importance, as detailed in [16]. The TextRank score for each sentence is normalized by dividing $T(S_i)$ with the maximum TextRank score of all sentences.

### 2.3 Position Score

Our system explores three position models. *The first position model is based on the assumption that sentences near the beginning and end of a document are more likely to be included in effective summaries.* This assumption is accomplished through the following cosine position score model, $P_{cos}$, which in previous empirical testing yielded superior results:

$$P_{cos}(S_i) = \frac{\cos \frac{2\pi x}{k-1} + \alpha - 1}{\alpha} \tag{3}$$

where $\alpha$ is the *dent factor* ($\alpha = 2$ was used in the evaluations described below based on optimal results obtained in prior experiments). The idea is that as $\alpha$ increases, the $P_{cos}$ score becomes more equally distributed, and as $\alpha$ decreases, $P_{cos}$ becomes more concentrated to value one at the beginning and end of a document. Here, $k$ represents the total number of sentences in the document and $x$ is the position of sentence $S_i$ within the document. The first sentence in the text document would have an $x$ value of 0 and the last sentence an $x$ value of $k - 1$.

The following function was the linear position score model used, $P_{lin}$, for an individual sentence in a document. *The assumption in using this model is based on efforts to prioritize sentences closer to only the top portions of a document.* This is accomplished through the following scoring model:

$$P_{lin}(S_i) = 1 - \frac{x}{k} \tag{4}$$

where $x$ and $k$ represent the same values as in the cosine position score equation (3) above. Essentially, as the $x$ value increases, the score decreases, giving higher weight to the sentences at top portions of a document.

A third position score function was designed for our system based on a correlation and regression analysis performed by us (we omit this for lack of space here) on data obtained from a previous cognitive experiment [13]. Essentially, a set of four scientific articles articles (that either contained heading or not) were assigned to a group of people who were asked to make short summaries from these. Of all different factors analyzed from this data, sentences closer to preceding *signaling devices* such as headings or titles, were found to be mostly correlated with human sentence extraction.

*The purpose of this scoring algorithm is therefore to prioritize sentences closer to topic headings, a condition that as we've seen, has a strong effect in sentence extraction decisions made by humans.* Of all four article analyzed, we found the following equation to model this correlation best (we omit details of this analysis but present here the equation used as a means of making closer human extractions):

$$P(S_i) = -19 \ln(d_i) + 51.926 \tag{5}$$

where $d_i$ represents the positional distance of sentence $S_i$ from a previous signaling device, such as a title, or heading encountered in the document. The position scores of each sentence are normalized by dividing with the maximum respective position score.

### 2.4 WordNet Score

The WordNet score method (so named due to the use of WordNet [6]) is the major word analysis component of the our system. One approach to a sentence's WordNet score is to determine the combination of a sentence's noun and verb score as a means of selecting a document's most *thematic* sentences. Informally, the noun and verb scores (*NS* and *VS*) determine the location of nouns and verbs, respectively, within the WordNet hypernymy graphs. *Hypernyms* here are words that by definition, are general representatives of other words. For instance, the word *dog* would be considered a hypernym to the word *poodle*, and *poodle* a *hyponym* to *dog* since *poodle* is a *type* of dog. *The purpose of this scoring algorithm is to prioritize sentences containing nouns and verbs closer to the their root forms since these could lead to the most thematic sentences.* The first WordNet score (*WN*) model for an individual sentence is presented as follows:

$$WN(S_i) = 1 - \frac{VS + NS}{(|Nouns(S_i)| + |Verbs(S_i)|)^2} \tag{6}$$

Here $Noun(S_i)$ (resp. $Verbs(S_i)$) denotes the set of nouns (resp. verbs) in sentence $S_i$. *VS*, here, represents a total verb score given to the individual verbs of the sentence and their distances to their own root forms within the hypernymy tree structure. *NS*, similarly, represents a total score given to the nouns of the sentence and their distances to their roots forms (for details on NS and VS calculations, see [22]). Essentially, the more general the nouns and verbs of a sentence are (determined by a simple traversing mechanism using WordNet's hypernymy tree structures), the higher the WordNet score weight is for this model. The denominator is squared based on results from prior experiments.

*The second model presented is intended to give higher priority to sentences containing words close in meaning to the article's popular keywords (computed by TextRank keyword extraction) and any other heading keywords within the entire text document.* The reason for its use is also based on the importance of keywords in sentence extraction, but this with the intention of examining keyword semantics through WordNet synonym lists.

The computation of this WordNet model revolved around the collection of a *thematic word list*, the combination of all document headings and the top five percent popular words generated by the TextRank keyword extraction algorithm [16]. Each document sentence, $S_i$ is assigned a score based on its individual bag of words with the following equation for each word $w$:

$$score(w) = \frac{1}{2^l}$$

where $l$ is the minimum level determined when $w$ is compared in meaning to words in *thematic word list* through WordNet's synonym lists, known as synsets. For instance, if $w$ is a word found in the thematic word list then $score(w) = 1$ (level $l = 0$). Otherwise, $l$ is increased by one to $(l = 1)$ and $w$ is now compared to the entire WordNet [6] synset list of the preceding level of the thematic word list. If no match is found, synsets of preceding synsets are determined with up to a maximum of 4 levels. Say $w$ is found at $l = 3$, then $score(w) = \frac{1}{8}$. $WN_{syn}$ score for $S_i$ in *SynSem* therefore became:

$$WN_{syn}(S_i) = \sum_{w \in S_i} score_{syn}(w) \tag{7}$$

Higher WordNet scores are achieved as the closer a sentence word, $w$, is to the *thematic list*'s synonyms, where $score_{syn}(w)$ represents the $score(w)$ computed using the WordNet synset relation. A third method is also presented based on the same procedure listed above, except hypernyms sets are used in place of the synsets described in Step 4. Equation for WN using hypernyms is similarly:

$$WN_{hyp}(S_i) = \sum_{w \in S_i} score_{hyp}(w) \tag{8}$$

Higher WordNet scores are similarly achieved as the closer a sentence word, $w$, is to the *thematic list*'s hypernym, or general word-forms, list. Here, $score_{hyp}(w)$ represents the $score(w)$ computed using the WordNet hypernymy relation. All three scores, $WN$, $WN_{syn}$ and $WN_{hyp}$ scores for each sentence are normalized by dividing with the maximum respective WordNet score over all sentences.

## 3 Data Sets

Two separate datasets were used for evaluating our and other systems' performances: 1) Cognitive experiment data originating from [13] (inspiration for equation (5) of 2.3) and composed of scientific-type magazine articles along with corresponding human-generated summaries and 2) DUC 2002 (sponsored by the National Institute of Standards and Technology, NIST) newspaper article set. Note that, 2002 is the last year in which participating systems in DUC were assigned to produce single document summaries, the very task analyzed for this study. Most systems that participated in DUC 2002 are not available for download, however, the summaries they produced for the DUC 2002 competition are available to us through NIST. The other systems used for comparison in this study include MEAD [19] and TextRank sentence extraction [16].

1. **Dataset A.** *Scientific article dataset* - The data obtained from the cognitive experiment contains the original four article versions distributed to the experiment's participants and the corresponding summaries constructed by the participants. Essentially, there were two different versions $A$ and $B$ of an article titled, "Energy Problems and Solutions" assigned to readers. These, however, were distributed as article versions that contained headings (which we refer to as *YA* and *YB*) and versions that lacked headings (which we refer to as *NA* and *NB*). The idea of that study was to determine the effects that headings had on human extraction, which were found to have major impact. We test our system on this dataset along with the human-constructed summaries as model extracts and evaluate our system's performance with respect to them.
2. **Dataset B.** *DUC02* The data provided by DUC 2002 contains a total of 533 unique news articles.[8] *Different variations of our system were executed among this set as well, constructing total summary sizes of exactly 100 words per article.* Note that DUC02 data is composed of articles containing no topic headings and only one title. Hence, the position scoring method (position score equation (5)) presented could not fully exploit its intended use in this particular set.

## 4 Evaluation

ROUGE [10] evaluation scores were used to compare our system extractions (using varying scoring models presented in this paper) to each other and to those produced by MEAD and TextRank. This fully automated evaluator essentially measures content similarity between system-developed summaries and

[8] Since multiple document summarization was also a task in DUC 2002 the files are grouped together in sets and these sets overlap - a total of 34 files are repeated in the collection, which brings the number down from 567 to 533. For single document summarization it does not make sense to repeat articles and can bias the results, so we have eliminated duplicate articles.

corresponding model summaries, usually developed by humans. Of all forms of measures it utilizes, ROUGE n-gram co-occurrences between system summaries and model summaries have been of most interest to our experiments. The n-grams, in this case, would specify the number of n consecutive word units that would have to overlap between a summary sentence and a model sentence in order to be counted as a match.

Each dataset required a set of model summaries for proper evaluations to take place. For Dataset A, the top 15 selected sentences for each article version were used as models for the evaluations. In the case of Dataset B, two manually produced 100-word reference summaries (these are abstractive, not extractive) are provided for each article in the data and used for the evaluation. All the ROUGE evaluations use all the words in the summaries, i.e., we do not use stemming (word generalization) or stopword elimination.

## 5 Results

The following results illustrate various executions of our system, TextRank sentence extraction [16], and MEAD [19] on the pair of datasets used in our analysis. We compare the evaluations of those summaries to the datasets' baselines. *In the case of Dataset B (DUC02), the baseline consisted of a summary of the first 100 words of each article and for Dataset A, the baseline consisted of the first 15 sentences from the source article version.* In addition, compression rates were established as follows. *All systems were required to produce 15-sentence summaries for Dataset A and 100-word summaries for Dataset B.* These were determined based on the original summary requirements corresponding to each dataset.

Table 1 shows ROUGE uni-gram evaluations when the systems were executed on Dataset A. For lack of space, we show results for only *YB* and *NB* (article version *B* containing headings [*YB*] and the one lacking headings [*NB*].) In the case of executions made on *YB* (Table 1(a)), our system's ROUGE scores manage to outperform those of MEAD, TextRank sentence extractions, and the baseline, most importantly. Best results were achieved in our system using topic heading filtering[9], position equation (4) and WordNet equation (6) with an F-measure of 0.71937. Executions made on the *NB* article (shown in Table 1 (b)) demonstrate similar outperformance and a highest F-measure resulting from our system of 0.65209. The same model combinations executed on both versions of the article resulted within the top 7 scoring systems of Tables 1 (a) and (b).

Table 2 illustrates ROUGE uni-grams scores on various executions made by our system for Dataset B. For lack of space, we only show a few top scoring model combinations. Optimal results here were achieved using the position model prioritizing sentences closer to distances-to-preceding headings (position equation 5), and the WordNet model exploiting synonym linkage to thematic content (WordNet equation 7), or in the case of DUC02, article titles and popular words.

[9] An option to filter headings and the inclusion of these in a final extract is an additional aspect of our system.

**Table 1.** Basic ROUGE evaluation scores for the baseline, our system, MEAD, and TextRank sentence extraction on Dataset A, showing results for articles *YB* and *NB*

| **Parameter Key for (Our System)** | | | |
|---|---|---|---|
| $N$ | Removal of topic headings in summary | $P_m$ | Position Score $P$ model $(m)$ |
| $H$ | Inclusion of topic headings in summary | $WN_m$ | WordNet Score $WN$ model $(m)$ |

**(a)** *YB*

| **Execution on *YB*** | **ROUGE uni-gram Scores** | | |
|---|---|---|---|
| *(Conditions)* | *Recall* | *Precision* | *F-mea.* |
| $(N,P_4,WN_6)$ | .74897 | .69202 | .71937 |
| $(H,P_4,WN_6)$ | .70782 | .67717 | .69216 |
| $(N,P_5,WN_7)$ | .55144 | .62617 | .58643 |
| $(N,P_5,WN_6)$ | .63786 | .54007 | .58491 |
| *Baseline* | .39506 | .61146 | .48000 |
| MEAD | .52263 | .42617 | .46950 |
| TextRank | .59671 | .36341 | .45172 |

**(b)** *NB*

| **Execution on *NB*** | **ROUGE uni-gram Scores** | | |
|---|---|---|---|
| *(Conditions)* | *Recall* | *Precision* | *F-mea.* |
| $(H,P_4,WN_6)$ | .65079 | .65339 | .65209 |
| $(N,P_4,WN_6)$ | .65476 | .63218 | .64327 |
| $(N,P_5,WN_7)$ | .59921 | .66520 | .63048 |
| $(N,P_5,WN_6)$ | .58730 | .66667 | .62447 |
| MEAD | .50794 | .42953 | .46546 |
| *Baseline* | .49603 | .43103 | .46125 |
| TextRank | .55556 | .34913 | .42879 |

**Table 2.** Basic ROUGE evaluation scores for our system on Dataset B – DUC02

| **DUC02 Dataset** | **ROUGE uni-gram Scores** | | | |
|---|---|---|---|---|
| *(Conditions)* | *Recall* | *Precision* | *F-measure* | *95% conf. int.* |
| $(N,P_5,WN_7)$ | .48159 | .45062 | .46549 | [.45753-.47260] |
| $(N,P_3,WN_7)$ | .48111 | .44995 | .46491 | [.45715-.47252] |
| $(N,P_5,WN_6)$ | .47965 | .45145 | .46491 | [.45774-.47236] |
| $(N,P_5,WN_8)$ | .47920 | .45195 | .46488 | [.45738-.47257] |
| $(N,P_4,WN_7)$ | .48091 | .44965 | .46466 | [.45689-.47236] |
| $(N,P_4,WN_6)$ | .47941 | .45113 | .46462 | [.45691-.47218] |
| $(N,P_4,WN_8)$ | .47930 | .45098 | .46450 | [.45724-.47228] |

Table 3 presents the top 7 out of 13 participating DUC02 systems compared with our system (the highest scoring from Table 2), MEAD, TextRank, and the baseline, all whose summaries contain up to 100 words only.[10] Our system using topic filtering, the closest distance-to-preceding headings position model, and the WordNet method exploiting synonyms obtains higher ROUGE F-measure scores than the baseline and all other participating systems but S28, a system which failed to produce one summary. Our system was ranked second in F-measure but according to [2,11], the recall metric can be prioritized since precision scores can be manipulated by adjusting the length of a candidate, or system, summary. If recall is only taken into consideration, then our system would rank first.

[10] Both manual abstracts and the system summaries are truncated to exactly 100 words whenever they exceed this limit.

**Table 3.** Basic ROUGE evaluation scores for our system, top 7 DUC02 systems, MEAD, TextRank sentence extraction, and the baseline

| **DUC02** | **ROUGE uni-gram Scores** | | | |
|---|---|---|---|---|
| *System* | *Recall* | *Precision* | *F-mea.* | *95% conf. int.* |
| S28 | .47813 | .45779 | .46729 | [.45986-.47418] |
| **Our System** | **.48159** | **.45062** | **.46549** | **[.45753-.47260]** |
| S19 | .45563 | .47748 | .46309 | [.45427-.47202] |
| *Baseline* | .47788 | .44680 | .46172 | [.45413-.46944] |
| S21 | .47543 | .44635 | .46029 | [.45209-.46802] |
| TextRank | .46165 | .43234 | .44640 | [.44004-.45348] |
| S29 | .46100 | .44557 | .45269 | [.44585-.45982] |
| S23 | .43188 | .47585 | .45018 | [.44191-.45900] |
| S27 | .45485 | .44808 | .45014 | [.44227-.45862] |
| MEAD | .44506 | .45290 | .44729 | [.43961-.45508] |
| S15 | .44805 | .43323 | .44014 | [.43203-.44799] |

## 6 Discussion

From the experimental results presented here, it is clear that our system succeeds in identifying important sentences in a text using information that is present only in the text and to do this within a summary that manages to outperform the documents' baseline. It is an unsupervised system with one caveat, the issue of weight selection, and requires no training data. When only a single article on a topic is available, we have devised unweighted schemes that deliver performance very close (F-measure to within 2-3%) to the optimal weighted schemes. For lack of space we omit these schemes and their performance here.

When a set of related articles is available, selection of weights can be done by adding a tuning module that uses a random subset of the data to find the best weight combination for the subset and then using it for the entire collection. To test this hypothesis, we conducted two experiments. In the first we took ten random samples of $\lceil \sqrt{D} \rceil$ articles from the $D = 533$ articles in DUC02 dataset and found the optimal combination of weights for each sample using ROUGE. All ten combinations of weights for the samples were in the top ten (F-measure) weight combinations for the entire DUC02 dataset. Of these ten optimal weight combinations for the samples, the two weight combinations with the highest frequencies, three times each, are the second and fifth best for the entire collection. This means that a small number of square-root size samples can give a near optimal combination of weights for the entire corpus. In a second experiment, we took 30 random samples of $\lceil log_2\ D \rceil$ documents from the 533 DUC02 documents, but here the results were not as good (a few optimal weight combinations for the sample were not in the top ten combinations for the entire dataset) as for the square-root size samples.

Our system outperforms MEAD and TextRank sentence extraction in all experiments of evaluation and is consistently higher than the baseline. Its ROUGE scores are also statistically significantly higher (through ROUGE 95% confidence intervals) than the baseline for the scientific magazine article set, where it is able take advantage of the headings in the article for its position score and the summary size restriction is on the number of sentences. When there are no headings in the articles and summary needs to be shorter (100 words versus 15 sentences), as for instance in the DUC dataset, it still beats the baseline.

## 7 Related Work

Sentence position has been considered important to summarization and information extraction ever since the late 1950s [3]. Many researchers have proposed using it for automatic summarization, e.g., see [5], [10], [20] and [14]. The importance of sentence position in *book length* documents was studied by [15], which are outside the scope of our study. Most researchers use sentence position based on their opinion of the language in which the document is written. Many use a linear function of the sentence position [9], [8] or sentence position with respect to a centroid sentence [20], others use either the first few sentences in a paragraph or the document. To our knowledge, this is the first objective study that analyzes human summary data for a “newspaper-length” article without requiring any key words[11]. Moreover, our work shows the importance of considering derived variables from the sentence position, not just the raw sentence position, and we observe a logarithmic relationship.

The importance of keywords or key phrases for summarization is also well-recognized since at least [5]. Many researchers have proposed using it, e.g., ( [9], [16]) among others. Although WordNet was used before in summarization, e.g, in SUMMARIST [7] for the task of topic interpretation, the usage is quite different from that of our methods and our WordNet scoring methodology is new to the best of our knowledge.

## 8 Conclusions

In this paper we have described the implications of basing a single-document summarization system on combining new syntactic and semantic techniques for sentence scoring. Results have demonstrated topic heading relevance to the overall position, and semantic linkages have produced effective summaries when experimented on both the DUC02 newswire and the scientific magazine article sets.

Our approach is easily adapted to specific domains that have ontologies available. There are several interesting directions for future work: the incorporation of heuristics that optimize the score of a summary given a size constraint, sentence compression (e.g., [1]), and criteria for measuring inter-sentence redundancy and its minimization. We have recently extended this approach to multi-document summarization and are currently evaluating it. Extensions of our approach on stronger semantic summary evaluations are other avenues for the future.

[11] Lin and Hovy’s work on optimum position policy [12] requires a corpus along with key words.

## References

1. Angheluta, R., Mitra, R., Jing, X., Moens, M.-F.: K.U. Leuven Summarization System at DUC 2004. Available on the Web (2004)
2. Arora, R., Ravindran, B.: Latent Dirichlet Allocation and Singular Value Decomposition based Multi-Document Summarization. In: ICDM 2008: Proceedings of the 2008 Eighth IEEE Int'l Conf. on Data Mining, pp. 713–718 (2008)
3. Baxendale, P.: Machine-made Index for Technical Literature - An Experiment. IBM Journal of Research Development 2(4), 354–361 (1958)
4. Brin, S., Page, L.: The Anatomy of Large-scale Hypertextual Web Search Engine. Computer Networds and ISDN Systems 30, 1–7 (1998)
5. Edmundson, H.: New Methods in Automatic Extraction. Journal of ACM 16(2), 264–285 (1969)
6. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1998)
7. Hovy, E., Lin, C.: Automatic Text Summarization in SUMMARIST. In: Mani, Maybury, M. (eds.) Adv. in Text Summarization, vol. 1. MIT Press (1999)
8. Ishikawa, K.: Trainable Automatic Text Summarization Using Segmentation of Sentence. In: Proceedings of the Third NTCIR Workshop (2003)
9. Li, S., Wang, W., Wang, C.: TAC 2009 Update Summarization Task of ICL. In: Text Analysis Conference 2008 (2008)
10. Lin, C.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of Workshop on Text Summarization Post-Conference Workshop (ACL 2004), Barcelona, Spain (2004)
11. Lin, C., Hovy, E.: Automatic Evaluation of Summaries Using n-gram Co-occurrence Statistics. In: HTL-NAACL (2003)
12. Lin, C.-Y., Hovy, E.H.: Identifying topics by position. In: ANLP, pp. 283–290 (1997)
13. Lorch, R., Lorch, E.: Effects of Headings of Text Recall and Summarization. Contemporary Educational Psychology 21, 261–278 (1996)
14. Mani, I., Maybury, M.: Advances in Automatic Summarization. MIT Press, Cambridge (1999)
15. Mihalcea, R., Ceylan, H.: Explorations in Automatic Book Summarization. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2007), Prague (2007)
16. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 (March 2004)
17. Nenkova, A.: Automatic Text Summarization of Newswire: Lessons Learned from the document understanding conference. In: AAAI, pp. 1436–1441 (2005)
18. Nenkova, A.: A General Introduction to Automatic Summarization (2009), http://webcast.jhu.edu/mediasite/Viewer/?peid=8cd235b1699a457f9c776c12d4925408
19. Radev, D., Allison, T.: Mead - a Platform for Multidocument Multilingual Text Summarization. In: LREC (2004)
20. Radev, D., Jing, H., Stys, M., Tam, D.: Centroid-based Summarization of Multiple Documents. Information Proc. and Mgmt. 40, 919–938 (2004)
21. Svore, K.M., Vanderwende, L., Burges, C.J.C.: Enhancing Single-document Summarization by Combining RankNet and Third-Party Sources. In: EMNLP-CoNLL, pp. 448–457 (2007)
22. Verma, R., Filozov, F.: Document Map and WN-Sum: A new framework for automatic text summarization and a first implementation. Technical Report UH-CS-10-03, University of Houston Computer Science Dept. (2010)