# Quantifying the Limits and Success of Extractive Summarization Systems Across Domains

**Hakan Ceylan** and **Rada Mihalcea**
Department of Computer Science
University of North Texas
Denton, TX 76203
{hakan,rada}@unt.edu

**Umut Özertem**
Yahoo! Labs
701 First Avenue
Sunnyvale, CA 94089
umut@yahoo-inc.com

**Elena Lloret** and **Manuel Palomar**
Department of
Software and Computing Systems
University of Alicante
San Vicente del Raspeig
Alicante 03690, Spain
{elloret,mpalomar}@dlsi.ua.es

## Abstract

This paper analyzes the topic identification stage of single-document automatic text summarization across four different domains, consisting of newswire, literary, scientific and legal documents. We present a study that explores the summary space of each domain via an exhaustive search strategy, and finds the probability density function (pdf) of the ROUGE score distributions for each domain. We then use this pdf to calculate the percentile rank of extractive summarization systems. Our results introduce a new way to judge the success of automatic summarization systems and bring quantified explanations to questions such as why it was so hard for the systems to date to have a statistically significant improvement over the lead baseline in the news domain.

## 1 Introduction

Topic identification is the first stage of the generally accepted three-phase model in automatic text summarization, in which the goal is to identify the most important units in a document, i.e., phrases, sentences, or paragraphs (Hovy and Lin, 1999; Lin, 1999; Sparck-Jones, 1999). This stage is followed by the topic interpretation and summary generation steps where the identified units are further processed to bring the summary into a coherent, human readable abstract form. The extractive summarization systems, however, only employ the topic identification stage, and simply output a ranked list of the units according to a compression ratio criterion. In general, for most systems sentences are the preferred units in this stage, as they are the smallest grammatical units that can express a statement.

Since the sentences in a document are reproduced verbatim in extractive summaries, it is theoretically possible to explore the search space of this problem through an enumeration of all possible extracts for a document. Such an exploration would not only allow us to see how far we can go with extractive summarization, but we would also be able to judge the difficulty of the problem by looking at the distribution of the evaluation scores for the generated extracts. Moreover, the high scoring extracts could also be used to train a machine learning algorithm.

However, such an enumeration strategy has an exponential complexity as it requires all possible sentence combinations of a document to be generated, constrained by a given word or sentence length. Thus the problem quickly becomes impractical as the number of sentences in a document increases and the compression ratio decreases. In this work, we try to overcome this bottleneck by using a large cluster of computers, and decomposing the task into smaller problems by using the given section boundaries or a linear text segmentation method. As a result of this exploration, we generate a probability density function (pdf) of the ROUGE score (Lin, 2004) distributions for four different domains, which shows the distribution of the evaluation scores for the generated extracts, and allows us to assess the difficulty of each domain for extractive summarization.

Furthermore, using these pdfs, we introduce a new success measure for extractive summarization systems. Namely, given a system's average score over a data set, we show how to calculate the per-



*Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 903–911,
Los Angeles, California, June 2010. 

centile rank of this system from the corresponding pdf of the data set. This allows us to see the true improvement a system achieves over another, such as a baseline, and provides a standardized scoring scheme for systems performing on the same data set.

## 2 Related Work

Despite the large amount of work in automatic text summarization, there are only a few studies in the literature that employ an exhaustive search strategy to create extracts, which is mainly due to the prohibitively large search space of the problem. Furthermore, the research regarding the alignment of abstracts to original documents has shown great variations across domains (Kupiec et al., 1995; Teufel and Moens, 1997; Marcu, 1999; Jing, 2002; Ceylan and Mihalcea, 2009), which indicates that the extractive summarization techniques are not applicable to all domains at the same level.

In order to automate the process of corpus construction for automatic summarization systems, (Marcu, 1999) used exhaustive search to generate the best *Extract* from a given *(Abstract, Text)* tuple, where the best *Extract* contains a set of clauses from *Text* that have the highest similarity to the given *Abstract*.

In addition, (Donaway et al., 2000) used exhaustive search to create all the sentence extracts of length three starting with 15 TREC Documents, in order to judge the performance of several summary evaluation measures suggested in their paper.

Finally, the study most similar to ours was done by (Lin and Hovy, 2003), who used the articles with less than 30 sentences from the DUC 2001 data set to find *oracle extracts* of 100 and 150 ($\pm 5$) words. These extracts were compared against one summary source, selected as the one that gave the highest inter-human agreement. Although it was concluded that a 10% improvement was possible for extractive summarization systems, which typically score around the lead baseline, there was no report on how difficult it would be to achieve this improvement, which is the main objective of our paper.

## 3 Description of the Data Set

Our data set is composed of four different domains: newswire, literary, scientific and legal. For all the domains we used 50 documents and only one summary for each document, except for newswire where we used two summaries per document. For the newswire domain, we selected the articles and their summaries from the DUC 2002 data set,[1]. For the literary domain, we obtained 10 novels that are literature classics, and available online in text format. Further, we collected the corresponding summaries for these novels from various websites such as CliffsNotes (www.cliffsnotes.com) and SparkNotes (www.sparknotes.com), which make available human generated abstracts for literary works. These sources give a summary for each chapter of the novel, so each chapter can be treated as a separate document. Thus we evaluate 50 chapters in total. For the scientific domain, we selected the articles from the medical journal *Autoimmunity Reviews*[2] were selected, and their abstracts are used as summaries. Finally, for the legal domain, we gathered 50 law documents and their corresponding summaries from the European Legislation Website,[3] which comprises four types of laws - *Council Directives*, *Acts*, *Communications*, and *Decisions* over several topics, such as society, environment, education, economics and employment.

| Domain | $\mu_{Dw}$ | $\mu_{Sw}$ | $\mu_R$ | $\mu_C$ | $\mu_{Cw}$ |
|---|---|---|---|---|---|
| Newswire | 641 | 101 | 84% | 1 | 641 |
| Literary | 4973 | 1148 | 77% | 6 | 196 |
| Scientific | 1989 | 160 | 92% | 9 | 221 |
| Legal | 3469 | 865 | 75% | 18 | 192 |

Table 1: Statistical properties of the data set. $\mu_{Dw}$, and $\mu_{Sw}$ represent the average number of words for each document and summary respectively; $\mu_R$ indicates the average compression ratio; and $\mu_C$ and $\mu_{Cw}$ represent the average number of sections for each document, and the average number of words for each section respectively.

Although all the summaries are human generated abstracts for all the domains, it is worth mentioning that the documents and their corresponding summaries exhibit a specific writing style for each domain, in terms of the vocabulary used and the length of the sentences. We list some of the statistical properties of each domain in Table 1.

[1] http://www-nlpir.nist.gov/projects/duc/data.html
[2] http://www.elsevier.com/wps/product/cws_home/622356
[3] http://eur-lex.europa.eu/en/legis/index.htm

## 4 Experimental Setup

As mentioned in Section 1, an exhaustive search algorithm requires generating all possible sentence combinations from a document, and evaluating each one individually. For example, using the values from Table 1, and assuming 20 words per sentence, we find that the search space for the news domain contains approximately $\binom{32}{5} \times 50 = 10,068,800$ summaries. The same calculation method for the scientific domain gives us $\binom{99}{8} \times 50 = 8.56 \times 10^{12}$ summaries. Obviously the search space gets much bigger for the legal and literary domains due to their larger text size.

In order to be able to cope with such a huge search space, the first thing we did was to modify the ROUGE 1.5.5[4] Perl script by fixing the parameters to those used in the DUC experiments,[5] and also by modifying the way it handles the input and output to make it suitable for streaming on the cluster.

The resulting script evaluates around 25-30 summaries per second on an Intel 2.33 GHz processor. Next, we streamed the resulting ROUGE script for each (document, summary) pair on a large cluster of computers running on an Hadoop Map-Reduce framework.[6] Based on the size of the search space for a (document, summary) pair, the number of computers allocated in the cluster ranged from just a few to more than one thousand.

Although the combination of a large cluster and a faster ROUGE is enough to handle most of the documents in the news domain in just a few hours, a simple calculation shows that the problem is still impractical for the other domains. Hence for the scientific, legal, and literary domains, rather than considering each document as a whole, we divide them into sections, and create extracts for each section such that the length of the extract is proportional to the length of the section in the original document. For the legal and scientific domains, we use the given section boundaries (without considering the subsections for scientific documents). For the novels, we treat each chapter as a single document (since each chapter has its own summary), which is further divided into sections using a publicly available linear text segmentation algorithm by (Utiyama and Isahara, 2001).[7] In all cases, we let the algorithm pick the number of segments automatically.

To evaluate the sections, we modified ROUGE further so that it applies the length constraint to the extracts only, not to the model summaries. This is due to the fact that we evaluate the extracts of each section individually against the whole model summary, which is larger than the extract. This way, we can get an overall ROUGE recall score for a document extract, simply by summing up the recall scores of each section extracts. The precision score for the entire document can also be found by adding the weighted precision scores for each section, where the weight is proportional to the length of the section in the original document. In our study, however, we only use recall scores.

Note that, since for the legal, scientific, and literary domains we consider each section of a document independently, we are not performing a true exhaustive search for these domains, but rather solving a suboptimal problem, as we divide the number of words in the model summary to each section proportional to the section's length. However, we believe that this is a fair assumption, as it has been shown repeatedly in the past that text segmentation helps improving the performance of text summarization systems (yen Kan et al., 1998; Nakao, 2000; Mihalcea and Ceylan, 2007).

## 5 Exhaustive Search Algorithm

Let $E_{i_k} = S_{i_1}, S_{i_2}, ..., S_{i_k}$ be the $i^{th}$ extract that has $k$ sentences, and generated from a document $D$ with $n$ sentences $D = S_1, S_2, \ldots, S_n$. Further, let $len(S_j)$ give the number of words in sentence $S_j$. We enforce that $E_{i_k}$ satisfies the following constraints:

$$\begin{aligned} len(E_{i_k}) &= len(S_{i_1}) + \ldots + len(S_{i_k}) \geq L \\ len(E_{i_{k-1}}) &= len(S_{i_1}) + \ldots + len(S_{i_{k-1}}) < L \end{aligned}$$

where $L$ is the length constraint on all the extracts of document $D$. We note that for any $E_{i_k}$, the order of the sentences in $E_{i_{k-1}}$ does not affect the ROUGE scores, since only the last sentence may be

[4] http://berouge.com

[5] -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0

[6] http://hadoop.apache.org/

[7] http://mastarpj.nict.go.jp/ mutiyama/software/textseg/textseg-1.211.tar.gz

chopped off due to the length constraint.[8] Hence, we start generating sentence combinations $\binom{n}{r}$ in *lexicographic order*, for $r = 1...n$, and for each combination $E_{i_k} = S_{i_1}, S_{i_2}, ..., S_{i_k}$ where $k > 1$, we generate additional extracts $E'_{i_k}$ by successfully swapping $S_{i_j}$ with $S_{i_k}$ for $j = 1, ..., k-1$ and checking to see if the above constraints are still satisfied. Therefore from a combination with $k$ sentences that satisfies the constraints, we might generate up to $k-1$ additional extracts. Finally, we stop the process either when $r = n$ and the last combination is generated, or we cannot find any extract that satisfies the constraints for $r$.

## 6 Generating pdfs

Once the extracts for a document are generated and evaluated, we go through each result and assign its recall score to a range, which we refer to as a bin. We use $1,000$ equally spaced bins between $0$ and $1$. As an example, a recall score of $0.46873$ would be assigned to the bin $[0.468, 0.469]$. By keeping a count for each bin, we are in fact building a histogram of scores for the document. Let this histogram be $h$, and $h[j]$ be the value in the $j^{th}$ bin of the histogram. We then define the normalized histogram $\widehat{h}$ as:

$$\widehat{h}[j] = \frac{N}{\sum_{i=1}^{N} h[j]}\, h[j] \qquad (1)$$

where $N = 1,000$ is the number of bins in the histogram. Note that since the *width* of each bin is $\frac{1}{N}$, the Riemann sum of the normalized histogram $\widehat{h}$ is equal to 1, so $\widehat{h}$ can be used as an approximation to the underlying pdf. As an example, we show the histogram $\widehat{h}$ for the newswire document AP890323-0218 in Figure 1.

We combine the normalized histograms of all the documents in a domain in order to find the pdf for that domain. This requires multiplying the value of each bin in a document's histogram, with all the other possible combinations of bin values taken from each of the remaining histograms, and assigning the result to the average bin for each combination. This can be done iteratively by keeping a moving average. We illustrate this procedure in Algorithm 1, where $K$ represents the number of documents in a domain.

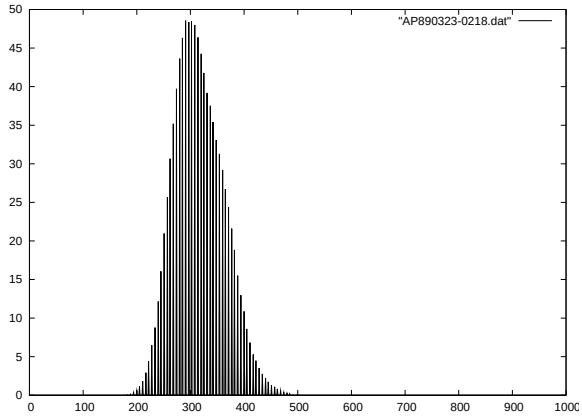


Figure 1: The normalized histogram $\widehat{h}$ of ROUGE-1 recall scores for the newswire document AP890323-0218.

**Algorithm 1** Combine $\widehat{h}^i$'s for $i = 1, \ldots, K$ to create $h_d$, the histogram for domain $d$.

```
1:  h_d := {}
2:  for i = 1 to N do
3:      h_d[i] := ĥ^1[i]
4:  end for
5:  for i = 2 to K do
6:      h_t = {}
7:      for j = 1 to N do
8:          for k = 1 to N do
9:              a = round(((k * (i − 1)) + j)/i)
10:             h_t[a] = h_t[a] + (h_d[k] * ĥ^i[j])
11:         end for
12:     end for
13:     h_d := h_t
14: end for
```

The resulting histogram $h_d$, when normalized using Equation 1, is an approximation to the pdf for domain $d$. Furthermore, we used the $round()$ function in line 9, which rounds a number to the nearest integer, as the bins are indexed by integers. Note that this rounding introduces an error, which is distributed uniformly due to the nature of the $round()$ function. It is also possible to lower the affect of this error with higher resolutions (i.e. larger number of bins). In Figure 2, we show a sample $h_d$, obtained by combining 10 documents from the newswire do-

[8] Note that we do not take the coherence of extracts into account, i.e. the sentences in an extract do not need to be sorted in order of their appearance in the original document. We also do not change the position of the words in a sentence.

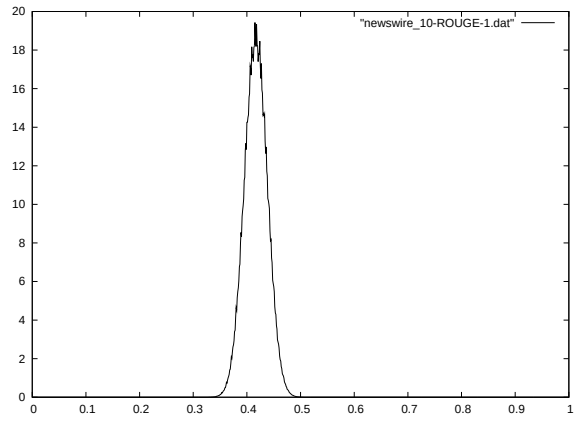


Figure 2: An example pdf obtained by combining 10 document histograms of ROUGE-1 recall scores from the newswire domain. The x-axis is normalized to [0,1].

main.

Recall from Section 4 that the documents in the literary, legal, and scientific domains are divided into sections either by using the given section boundaries or by applying a text segmentation algorithm, and the extracts of each section are then evaluated individually. Hence for these domains, we first calculate the histogram of each section individually, and then combine them to find the histogram of a document. The combination procedure for the section histograms is similar to Algorithm 1, except that in this case we do not keep a moving average, but rather sum up the bins of the sections. Note that when bin $i$ and $j$ are added, the resulting values should be expected to be half the times in bin $i + j$, and half the times in $i + j - 1$.

## 7 Calculating Percentile Ranks

Given a pdf for a domain, the success of a system having a ROUGE recall score of $S$ could be simply measured by finding the area bounded by $S$. This gives us the percentile rank of the system in the overall distribution. Assuming $0 \leq S \leq 1$, let $\widehat{S} = \lfloor N \times S \rfloor$, then the formula to calculate the percentile rank can be simply given as:

$$PR(S) = \frac{100}{N} \sum_{i=1}^{\widehat{S}} \widehat{h_d}[i] \qquad (2)$$

| | ROUGE-1 | | | |
|---|---|---|---|---|
| Domain | $\mu$ | $\sigma$ | max | min |
| Newswire | 39.39 | 0.87 | 65.70 | 20.20 |
| Literary | 45.20 | 0.47 | 63.90 | 28.40 |
| Scientific | 45.99 | 0.68 | 71.90 | 24.20 |
| Legal | 72.82 | 0.28 | 82.40 | 62.80 |
| | ROUGE-2 | | | |
| Domain | $\mu$ | $\sigma$ | max | min |
| Newswire | 11.57 | 0.79 | 37.40 | 1.60 |
| Literary | 5.41 | 0.34 | 16.90 | 1.80 |
| Scientific | 10.98 | 0.60 | 33.30 | 1.30 |
| Legal | 28.74 | 0.29 | 40.90 | 19.60 |
| | ROUGE-SU4 | | | |
| Domain | $\mu$ | $\sigma$ | max | min |
| Newswire | 15.33 | 0.69 | 38.10 | 6.40 |
| Literary | 13.28 | 0.30 | 24.30 | 6.90 |
| Scientific | 16.13 | 0.50 | 35.80 | 6.20 |
| Legal | 35.63 | 0.25 | 45.70 | 28.70 |

Table 2: Statistical properties of the pdfs

## 8 Results

The ensemble distributions of ROUGE-1 recall scores per document are shown in Figure 3. The ensemble distributions tell us that the performance of the extracts, especially for the news and the scientific domains, are mostly uniform for each document. This is due to the fact that documents in these domains, and their corresponding summaries, are written with a certain conventional style. There is however a little scattering in the distributions of the literary and the legal domains. This is an expected result for the literary domain, as there is no specific summarization style for these documents, but somehow surprising for the legal domain, where the effect is probably due to the different types of legal documents in the data set.

The pdf plots resulting from the ROUGE-1 recall scores are shown in Figure 4.[9] In order to analyze the pdf plots, and better understand their differences, Table 2 lists the mean ($\mu$) and the standard deviation ($\sigma$) measures of the pdfs, as well as the average minimum and maximum scores that an extractive summarization system can get for each domain.

By looking at the pdf plots and the minimum and maximum columns from Table 2, we notice that for

[9] Similar pdfs are obtained for ROUGE-2 and ROUGE-SU4, even if at a different scale.

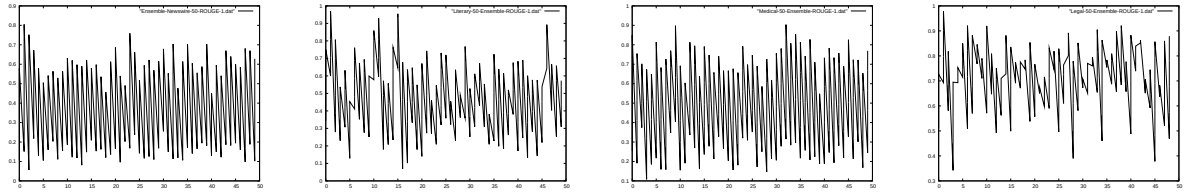

Figure 3: ROUGE-1 recall score distributions per document for Newswire, Literary, Scientific and Legal Domains, respectively from left to right.

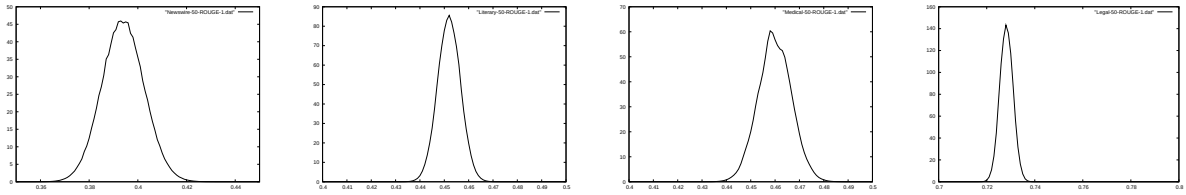

Figure 4: Probability Density Functions of ROUGE-1 recall scores for the Newswire, Literary, Scientific and Legal Domains, respectively from left to right. The resolution of the x-axis is increased to 0.1.

all the domains, the pdfs are long-tailed distributions. This immediately implies that most of the extracts in a summary space are clustered around the mean, which means that for automatic summarization systems, it is very easy to get scores around this range. Furthermore, we can judge the hardness of each domain by looking at the standard deviation values. A lower standard deviation indicates a steeper curve, which implies that improving a system would be harder. From the table, we can infer that the legal domain is the hardest while the newswire is the easiest.

Comparing Table 2 with the values in Table 1, we also notice that the compression ratio affects the performance differently for each domain. For example, although the scientific domain has the highest compression ratio, it has a higher mean than the literary and the newswire domains for ROUGE-1 and ROUGE-SU4 recall scores. This implies that although the abstracts of the medical journals are highly compressed, they have a high overlap with the document, probably caused by their writing style. This was in fact confirmed earlier by the experiments in (Kupiec et al., 1995), where it was found out that for a data set of 188 scientific articles, 79% of the sentences in the abstracts could be perfectly matched with the sentences in the corresponding documents.

Next, we confirm our experiments by testing three different extractive summarization systems on our data set. The first system that we implement is called *Random*, and gives a random score between 1 and 100 to each sentence in a document, and then selects the top scoring sentences. The second system, *Lead*, implements the lead baseline method which takes the first $k$ sentences of a document until the length limit is reached. Finally, the last system that we implement is *TextRank*, which uses a variation of the PageRank graph centrality algorithm in order to identify the most important sentences in a document (Page et al., 1999; Erkan and Radev, 2004; Mihalcea and Tarau, 2004). We selected TextRank as it has a performance competitive with the top systems participating in DUC '02 (Mihalcea and Tarau, 2004). We would also like to mention that for the literary, scientific, and legal domains, the systems apply the algorithms for each section and each section is evaluated independently, and their resulting recall scores are summed up. This is needed in order to be consistent with our exhaustive search experiments.

The ROUGE recall scores of the three systems are shown in Table 3. As expected, for the literary and legal domains, the *Random*, and the *Lead* systems score around the mean. This is due to the fact that the leading sentences for these two domains do not indicate any significance, hence the *Lead* system just behaves like *Random*. However for the scientific and newswire domains, the leading sentences do have

| | ROUGE-1 | | |
|---|---|---|---|
| Domain | Random | Lead | TextRank |
| Newswire | 39.13 | 45.63 | 44.43 |
| Literary | 45.39 | 45.36 | 46.12 |
| Scientific | 45.75 | 47.18 | 49.26 |
| Legal | 73.04 | 72.42 | 74.82 |
| | ROUGE-2 | | |
| Domain | Random | Lead | TextRank |
| Newswire | 11.39 | 19.60 | 17.99 |
| Literary | 5.33 | 5.41 | 5.92 |
| Scientific | 10.73 | 12.07 | 12.76 |
| Legal | 28.56 | 28.92 | 31.06 |
| | ROUGE-SU4 | | |
| Domain | Random | Lead | TextRank |
| Newswire | 15.07 | 21.58 | 20.46 |
| Literary | 13.21 | 13.28 | 13.81 |
| Scientific | 15.92 | 17.12 | 17.85 |
| Legal | 35.41 | 35.55 | 37.64 |

Table 3: ROUGE recall scores of the Lead baseline, TextRank, and Random sentence selector across domains

| | ROUGE-1 | | |
|---|---|---|---|
| Domain | Random | Lead | TextRank |
| Newswire | %39.18 | %99.99 | %99.99 |
| Literary | %62.89 | %62.89 | %97.90 |
| Scientific | %42.30 | %95.56 | %99.87 |
| Legal | %79.47 | %16.19 | %99.99 |
| | ROUGE-2 | | |
| Domain | Random | Lead | TextRank |
| Newswire | %39.57 | %99.99 | %99.99 |
| Literary | %42.20 | %54.32 | %94.34 |
| Scientific | %35.6 | %96.03 | %99.79 |
| Legal | %36.68 | %75.38 | %99.99 |
| | ROUGE-SU4 | | |
| Domain | Random | Lead | TextRank |
| Newswire | %40.68 | %99.99 | %99.99 |
| Literary | %46.39 | %46.39 | %96.84 |
| Scientific | %36.37 | %97.69 | %99.94 |
| Legal | %23.53 | %42.00 | %99.99 |

Table 4: Percentile rankings of the Lead baseline, TextRank, and Random sentence selector across domains

importance so the *Lead* system consistently outperforms *Random*. Furthermore, although *TextRank* is the best system for the literary, scientific, and legal domains, it gets outperformed by the *Lead* system on the newswire domain. This is also an expected result as none of the single-document summarization systems were able to achieve a statistically significant improvement over the lead baseline in the previous Document Understanding Conferences (DUC).

The ROUGE scoring scheme does not tell us how much improvement a system achieved over another, or how far it is from the upper bound. Since we now have access to the pdf of each domain in our data set, we can find this information simply by calculating the percentile rank of each system using the formula given in Equation 2.

The percentile ranks of all three systems for each domain are shown in Table 4. Notice how different the gap is between the scores of each system this time, compared to the scores in Table 3. For example, we see in Table 3 that *TextRank* on scientific domain has only a 3.51 ROUGE-1 score improvement over a system that randomly selects sentences to include in the extract. However, Table 4 tells us that this improvement is in fact 57.57%.

From Table 4, we see that both *TextRank* and the *Lead* system are in the 99.99% percentile of the newswire domain although the systems have 1.20, 1.61, and 1.12 difference in their ROUGE-1, ROUGE-2, and ROUGE-SU4 scores respectively. The high percentile for the *Lead* system explains why it was so hard to improve over these baseline in previous evaluations on newswire data (e.g., see the evaluations from the Document Understanding Conferences). Furthermore, we see from Table 2 that the upper bounds corresponding to these scores are 65.7, 37.4, and 38.1 respectively, which are well above both the *TextRank* and the *Lead* systems. Therefore, the percentile rankings of the *Lead* and the *TextRank* systems for this domain do not seem to give us clues about how the two systems compare to each other, nor about their actual distance from the upper bounds. There are two reasons for this: First, as we mentioned earlier, most of the summary space consists of *easy* extracts, which make the distribution long-tailed.[10] Therefore even though we have quite a bit of systems achieving high scores, their number is negligible compared to the millions of extracts that are clustered around the mean. Secondly, we need a higher resolution (i.e. larger number of bins) in constructing the pdfs in order to be able to

[10]This also accounts for the fact that even though we might have two very close ROUGE scores that are not statistically significant, their percentile rankings might differ quite a bit.

see the difference more clearly between the two systems. Finally, when comparing two successful systems using percentile ranks, we believe the use of error reduction would be more beneficial.

As a final note, we also randomly sampled extracts from documents in the scientific and legal domains, but this time without considering the section boundaries and without performing any segmentation. We kept the number of samples for each document equal to the number of extracts we generated from the same document using a divide-and-conquer approach. We evaluated the samples using ROUGE-1 recall scores, and obtained pdfs for each domain using the same strategy discussed earlier in the paper. The resulting pdfs, although they exhibit similar characteristics, they have mean values ($\mu$) around 10% lower than the ones we listed in Table 2, which supports the findings from earlier research that segmentation is useful for text summarization.

## 9 Conclusions and Future Work

In this paper, we described a study that explores the search space of extractive summaries across four different domains. For the news domain we generated all possible extracts of the given documents, and for the literary, scientific, and legal domains we followed a divide-and-conquer approach by chunking the documents into sections, handled each section independently, and combined the resulting scores at the end. We then used the distributions of the evaluations scores to generate the probability density functions (pdfs) for each domain. Various statistical properties of these pdfs helped us asses the difficulty of each domain. Finally, we introduced a new scoring scheme for automatic text summarization systems that can be derived from the pdfs. The new scheme calculates a percentile rank of the ROUGE-1 recall score of a system, which gives scores in the range [0-100]. This lets us see how far each system is from the upper bound, and thus make a better comparison among the systems. The new scoring system showed us that while there is a 20.1% gap between the upper bound and the lead baseline for the news domain, closing this gap is difficult, as the percentile rank of the lead baseline system, 99.99%, indicates that the system is already very close to the upper bound.

Furthermore, except for the literary domain, the percentile rank of the *TextRank* system is also very close to the upperbound. This result does not suggest that additional improvements cannot be made in these domains, but that making further improvements using only extractive summarization will be considerably difficult. Moreover, in order to see these future improvements, a higher resolution (i.e. larger number of bins) will be needed when constructing the pdfs.

In all our experiments we used the ROUGE (Lin, 2004) evaluation package and its ROUGE-1, ROUGE-2, and ROUGE-SU4 recall scores. We would like to note that since ROUGE performs its evaluations based on the n-gram overlap between the peer and the model summary, it does not take other summary quality metrics such as coherence and cohesion into account. However, our goal in this paper was to analyze the topic-identification stage only, which concentrates on selecting the right content from the document to include in the summary, and the ROUGE scores were found to correlate well with the human judgments on assessing the content overlap of summaries.

In the future, we would like to apply a similar exhaustive search strategy, but this time with different compression ratios, in order to see the impact of compression ratios on the pdf of each domain. Furthermore, we would also like to analyze the high scoring extracts found by the exhaustive search, in terms of coherence, position and other features. Such an analysis would allow us to see whether these extracts exhibit certain properties which could be used in training machine learning systems.

## Acknowledgments

The authors would like to thank the anonymous reviewers of NAACL-HLT 2010 for their feedback.

The work of the first author has been partly supported by an award from Google, Inc. The work of the fourth and fifth authors has been supported by an FPI grant (BES-2007-16268) from the Spanish Ministry of Science and Innovation, under the project TEXT-MESS (TIN2006-15265-C06-01) funded by the Spanish Government, and the project PROMETEO Desarrollo de Tcnicas Inteligentes e Interactivas de Minera de Textos (2009/119) from the Valencian Government.

## References


Hakan Ceylan and Rada Mihalcea. 2009. The decomposition of human-written book summaries. In *CICLing '09: Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 582–593, Berlin, Heidelberg. Springer-Verlag.

Robert L. Donaway, Kevin W. Drummey, and Laura A. Mather. 2000. A comparison of rankings produced by summarization evaluation measures. In *NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 69–78, Morristown, NJ, USA. Association for Computational Linguistics.

G. Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22.

Eduard H. Hovy and Chin Yew Lin. 1999. Automated text summarization in summarist. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 81–97. MIT Press.

Hongyan Jing. 2002. Using hidden markov modeling to decompose human-written summaries. *Comput. Linguist.*, 28(4):527–543.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73, New York, NY, USA. ACM.

Chin-Yew Lin and Eduard Hovy. 2003. The potential and limitations of automatic sentence extraction for summarization. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop*, pages 73–80, Morristown, NJ, USA. Association for Computational Linguistics.

Chin-Yew Lin. 1999. Training a selection function for extraction. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 55–62, New York, NY, USA. ACM.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Daniel Marcu. 1999. The automatic construction of large-scale corpora for summarization research. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 137–144, New York, NY, USA. ACM.

Rada Mihalcea and Hakan Ceylan. 2007. Explorations in automatic book summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 380–389, Prague, Czech Republic, June. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain.

Yoshio Nakao. 2000. An algorithm for one-page summarization of a long text based on thematic hierarchy detection. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 302–309, Morristown, NJ, USA. Association for Computational Linguistics.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Karen Sparck-Jones. 1999. Automatic summarising: Factors and directions. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 1–13. MIT Press.

Simone Teufel and Marc Moens. 1997. Sentence extraction as a classification task. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scallable Text Summarization*, Madrid, Spain, July.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In *In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 491–498.

Min yen Kan, Judith L. Klavans, and Kathleen R. McKeown. 1998. Linear segmentation and segment significance. In *In Proceedings of the 6th International Workshop of Very Large Corpora*, pages 197–205.