Enhancing Single-document Summarization by Combining RankNet and Third-party Sources

Krysta M. Svore
Microsoft Research
1 Microsoft Way
Redmond, WA 98052
ksvore@microsoft.com

Lucy Vanderwende
Microsoft Research
1 Microsoft Way
Redmond, WA 98052

Christopher J.C. Burges
Microsoft Research
1 Microsoft Way
Redmond, WA 98052

Abstract

We present a new approach to automatic summarization based on neural nets, called NetSum. We extract a set of features from each sentence that helps identify its importance in the document. We apply novel features based on news search query logs and Wikipedia entities. Using the RankNet learning algorithm, we train a pair-based sentence ranker to score every sentence in the document and identify the most important sentences. We apply our system to documents gathered from CNN.com, where each document includes highlights and an article. Our system significantly outperforms the standard baseline in the ROUGE-1 measure on over 70% of our document set.

1 Introduction

Automatic summarization was first studied almost 50 years ago by Luhn (Luhn, 1958) and has continued to be a steady subject of research. Automatic summarization refers to the creation of a shortened version of a document or cluster of documents by a machine, see (Mani, 2001) for details. The summary can be an abstraction or extraction. In an abstract summary, content from the original document may be paraphrased or generated, whereas in an extract summary, the content is preserved in its original form, i.e., sentences. Both summary types can involve sentence compression, but abstracts tend to be more condensed. In this paper, we focus on producing fully automated single-document extract summaries of newswire articles.

To create an extract, most automatic systems use linguistic and/or statistical methods to identify key words, phrases, and concepts in a sentence or across single or multiple documents. Each sentence is then assigned a score indicating the strength of presence of key words, phrases, and so on. Sentence scoring methods utilize both purely statistical and purely semantic features, for example as in (Vanderwende et al., 2006; Nenkova et al., 2006; Yih et al., 2007).

Recently, machine learning techniques have been successfully applied to summarization. The methods include binary classifiers (Kupiec et al., 1995), Markov models (Conroy et al., 2004), Bayesian methods (Daumé III and Marcu, 2005; Aone et al., 1998), and heuristic methods to determine feature weights (Schiffman, 2002; Lin and Hovy, 2002). Graph-based methods have also been employed (Erkan and Radev, 2004a; Erkan and Radev, 2004b; Mihalcea, 2005; Mihalcea and Tarau, 2005; Mihalcea and Radev, 2006).

In 2001–02, the Document Understanding Conference (DUC, 2001), issued the task of creating a 100-word summary of a single news article. The best performing systems (Hirao et al., 2002; Lal and Ruger, 2002) used various learning and semantic-based methods, although no system could outperform the baseline with statistical significance (Nenkova, 2005). After 2002, the single-document summarization task was dropped.

In recent years, there has been a decline in studies on automatic single-document summarization, in part because the DUC task was dropped, and in part because the task of single-document extracts may be counterintuitively more difficult than multi-

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 448–457, Prague, June 2007.

document summarization (Nenkova, 2005). However, with the ever-growing internet and increased information access, we believe single-document summarization is essential to improve quick access to large quantities of information. Recently, CNN.com (CNN.com, 2007a) added "Story Highlights" to many news articles on its site to allow readers to quickly gather information on stories. These highlights give a brief overview of the article and appear as 3–4 related sentences in the form of bullet points rather than a summary paragraph, making them even easier to quickly scan.

Our work is motivated by both the addition of highlights to an extremely visible and reputable online news source, as well as the inability of past single-document summarization systems to outperform the extremely strong baseline of choosing the first n sentences of a newswire article as the summary (Nenkova, 2005). Although some recent systems indicate an improvement over the baseline (Mihalcea, 2005; Mihalcea and Tarau, 2005), statistical significance has not been shown. We show that by using a neural network ranking algorithm and third-party datasets to enhance sentence features, our system, NetSum, can outperform the baseline with statistical significance.

Our paper is organized as follows. Section 2 describes our two studies: summarization and highlight extraction. We describe our dataset in detail in Section 3. Our ranking system and feature vectors are outlined in Section 4. We present our evaluation measure in Section 5. Sections 6 and 7 report on our results on summarization and highlight extraction, respectively. We conclude in Section 8 and discuss future work in Section 9.

2 Our Task

In this paper, we focus on single-document summarization of newswire documents. Each document consists of three highlight sentences and the article text. Each highlight sentence is human-generated, but is based on the article. In Section 4 we discuss the process of matching a highlight to an article sentence. The output of our system consists of purely extracted sentences, where we do not perform any sentence compression or sentence generation. We leave such extensions for future work.

We develop two separate problems based on our document set. First, can we extract three sentences that best "match" the highlights as a whole? In this task, we concatenate the three sentences produced by our system into a single summary or *block*, and similarly concatenate the three highlight sentences into a single summary or *block*. We then compare our system's block against the highlight block. Second, can we extract three sentences that best "match" the three highlights, such that ordering is preserved? In this task, we produce three sentences, where the first sentence is compared against the first highlight, the second sentence is compared against the second highlight, and the third sentence is compared against the third highlight. Credit is not given for producing three sentences that match the highlights, but are out of order. The second task considers ordering and compares sentences on an individual level, whereas the first task considers the three chosen sentences as a summary or block and disregards sentence order. In both tasks, we assume the title has been seen by the reader and will be listed above the highlights.

3 Evaluation Corpus

Our data consists of 1365 news documents gathered from CNN.com (CNN.com, 2007a). Each document was extracted by hand, where a maximum of 50 documents per day were collected. The documents were hand-collected on consecutive days during the month of February.

Each document includes the title, timestamp, story highlights, and article text. The timestamp on articles ranges from December 2006 to February 2007, since articles remain posted on CNN.com for up to several months. The story highlights are human-generated from the article text. The number of story highlights is between 3–4. Since all articles include at least 3 story highlights, we consider only the task of extracting three highlights from each article.

4 Description of Our System

Our goal is to extract three sentences from a single news document that best match various characteristics of the three document highlights. One way to identify the best sentences is to rank the sentences

TIMESTAMP: 1:59 p.m. EST, January 31, 2007

TITLE: Nigeria reports first human death from bird flu

HIGHLIGHT 1: Government boosts surveillance after woman dies

HIGHLIGHT 2: Egypt, Djibouti also have reported bird flu in humans

HIGHLIGHT 3: H5N1 bird flu virus has killed 164 worldwide since 2003

ARTICLE: 1. Health officials reported Nigeria's first cases of bird flu in humans on Wednesday, saying one woman had died and a family member had been infected but was responding to treatment. 2. The victim, a 22-year old woman in Lagos, died January 17, Information Minister Frank Nweke said in a statement. 3. He added that the government was boosting surveillance across Africa's most-populous nation after the infections in Lagos, Nigeria's biggest city. 4. The World Health Organization had no immediate confirmation. 5. Nigerian health officials earlier said 14 human samples were being tested. 6. Nweke made no mention of those cases on Wednesday. 7. An outbreak of H5N1 bird flu hit Nigeria last year, but no human infections had been reported until Wednesday. 8. Until the Nigerian report, Egypt and Djibouti were the only African countries that had confirmed infections among people. 9. Eleven people have died in Egypt. 10. The bird flu virus remains hard for humans to catch, but health experts fear H5N1 may mutate into a form that could spread easily among humans and possibly kill millions in a flu pandemic. 11. Amid a new H5N1 outbreak reported in recent weeks in Nigeria's north, hundreds of miles from Lagos, health workers have begun a cull of poultry. 12. Bird flu is generally not harmful to humans, but the H5N1 virus has claimed at least 164 lives worldwide since it began ravaging Asian poultry in late 2003, according to the WHO. 13. The H5N1 strain had been confirmed in 15 of Nigeria's 36 states. 14. By September, when the last known case of the virus was found in poultry in a farm near Nigeria's biggest city of Lagos, 915,650 birds had been slaughtered nationwide by government veterinary teams under a plan in which the owners were promised compensation. 15. However, many Nigerian farmers have yet to receive compensation in the north of the country, and health officials fear that chicken deaths may be covered up by owners reluctant to slaughter their animals. 16. Since bird flu cases were first discovered in Nigeria last year, Cameroon, Djibouti, Niger, Ivory Coast, Sudan and Burkina Faso have also reported the H5N1 strain of bird flu in birds. 17. There are fears that it has spread even further than is known in Africa because monitoring is difficult on a poor continent with weak infrastructure. 18. With sub-Saharan Africa bearing the brunt of the AIDS epidemic, there is concern that millions of people with suppressed immune systems will be particularly vulnerable, especially in rural areas with little access to health facilities. 19. Many people keep chickens for food, even in densely populated urban areas.

Figure 1: Example document containing highlights and article text. Sentences are numbered by their position. Article is from (CNN.com, 2007b).

using a machine learning approach, for example as in (Hirao et al., 2002). A train set is labeled such that the labels identify the best sentences. Then a set of features is extracted from each sentence in the train and test sets, and the train set is used to train the system. The system is then evaluated on the test set. The system learns from the train set the distribution of features for the best sentences and outputs a ranked list of sentences for each document. In this paper, we rank sentences using a neural network algorithm called RankNet (Burges et al., 2005).

4.1 RankNet

From the labels and features for each sentence, we train a model that, when run on a test set of sentences, can infer the proper ranking of sentences in a document based on information gathered during training about sentence characteristics. To accomplish the ranking, we use RankNet (Burges et al., 2005), a ranking algorithm based on neural networks.

RankNet is a pair-based neural network algorithm used to rank a set of inputs, in this case, the set of sentences in a given document. The system is trained on pairs of sentences (S_i, S_j), such that S_i should be ranked higher or equal to S_j. Pairs are generated between sentences in a single document, not across documents. Each pair is determined from the input labels. Since our sentences are labeled using ROUGE (see Section 4.3), if the ROUGE score of S_i is greater than the ROUGE score of S_j, then (S_i, S_j) is one input pair. The cost function for RankNet is the probabilistic cross-entropy cost function. Training is performed using a modified version of the back propagation algorithm for two layer nets (Le Cun et al., 1998), which is based on optimizing the cost function by gradient descent. A similar method of training on sentence pairs in the context of multi-document summarization was recently shown in (Toutanova et al., 2007).

Our system, NetSum, is a two-layer neural net trained using RankNet. To speed up the performance of RankNet, we implement RankNet in the framework of LambdaRank (Burges et al., 2006). For details, see (Burges et al., 2006; Burges et al., 2005). We experiment with between 5 and 15 hidden nodes and with an error rate between 10^{-2} and 10^{-7}.

We implement 4 versions of NetSum. The first version, NetSum(b), is trained for our first summarization problem (b indicates block). The pairs are generated using the maximum ROUGE scores l_1 (see Section 4.3). The other three rankers are trained to identify the sentence in the document that best matches highlight n. We train one ranker, NetSum(n), for each highlight n, for $n = 1, 2, 3$, resulting in three rankers. NetSum(n) is trained using pairs generated from the $l_{1,n}$ ROUGE scores between sentence S_i and highlight H_n (see Section 4.3).

4.2 Matching Extracted to Generated Sentences

In this section, we describe how to determine which sentence in the document best matches a given highlight. Choosing three sentences most similar to the three highlights is very challenging since the highlights include content that has been gathered across sentences and even paragraphs, and furthermore include vocabulary that may not be present in the text. Jing showed, for 300 news articles, that 19% of human-generated summary sentences contain no matching article sentence (Jing, 2002). In addition, only 42% of the summary sentences match the content of a single article sentence, where there are still semantic and syntactic transformations between the summary sentence and article sentence.. Since each highlight is human generated and does not exactly match any one sentence in the document, we must develop a method to identify how closely related a highlight is to a sentence. We use the ROUGE (Lin, 2004b) measure to score the similarity between an article sentence and a highlight sentence. We anticipate low ROUGE scores for both the baseline and NetSum due to the difficulty of finding a single sentence to match a highlight.

4.3 ROUGE

Recall-Oriented Understudy for Gisting Evaluation (Lin, 2004b), known as ROUGE, measures the quality of a model-generated summary or sentence by comparing it to a "gold-standard", typically human-generated, summary or sentence. It has been shown that ROUGE is very effective for measuring both single-document summaries and single-document headlines (Lin, 2004a).

ROUGE-N is a N-gram recall between a model-generated summary and a reference summary. We use ROUGE-N, for $N = 1$, for labeling and evaluation of our model-generated highlights.[1] ROUGE-1 and ROUGE-2 have been shown to be statistically similar to human evaluations and can be used with a single reference summary (Lin, 2004a). We have only one reference summary, the set of human-generated highlights, per document. In our work, the reference summary can be a single highlight sentence or the highlights as a block. We calculate ROUGE-N as

$$\frac{\sum_{gram_j \in R \cap S_i} Count(gram_j)}{\sum_{gram_j \in R} Count(gram_j)}, \quad (1)$$

where R is the reference summary, S_i is the model-generated summary, and N is the length of the N-gram $gram_j$.[2] The numerator cannot excede the number of N-grams (non-unique) in R.

We label each sentence S_i by its ROUGE-1 score. For the first problem of matching the highlights as a block, we label each S_i by l_1, the maximum ROUGE-1 score between S_i and each highlight H_n, for $n = 1, 2, 3$, given by $l_1 = \max_n(R(S_i, H_n))$.

For the second problem of matching three sentences to the three highlights individually, we label each sentence S_i by $l_{1,n}$, the ROUGE-1 score between S_i and H_n, given by $l_{1,n} = R(S_i, H_n)$. The ranker for highlight n, NetSum(n), is passed samples labeled using $l_{1,n}$.

4.4 Features

RankNet takes as input a set of samples, where each sample contains a label and feature vector. The labels were previously described in Section 4.3. In this section, we describe each feature in detail and motivate in part why each feature is chosen. We generate 10 features for each sentence S_i in each document, listed in Table 1. Each feature is chosen to identify characteristics of an article sentence that may match those of a highlight sentence. Some of the features such as position and N-gram frequencies are commonly used for scoring. Sentence scoring based on

[1] We use an implementation of ROUGE that does not perform stemming or stopword removal.

[2] ROUGE is typically used when the length of the reference summary is equal to length of the model-generated summary. Our reference summary and model-generated summary are different lengths, so there is a slight bias toward longer sentences.

Symbol	Feature Name
$F(S_i)$	Is First Sentence
$Pos(S_i)$	Sentence Position
$SB(S_i)$	SumBasic Score
$SB_b(S_i)$	SumBasic Bigram Score
$Sim(S_i)$	Title Similarity Score
$NT(S_i)$	Average News Query Term Score
$NT_+(S_i)$	News Query Term Sum Score
$NT_r(S_i)$	Relative News Query Term Score
$WE(S_i)$	Average Wikipedia Entity Score
$WE_+(S_i)$	Wikipedia Entity Sum Score

Table 1: Features used in our model.

sentence position, terms common with the title, appearance of keyword terms, and other cue phrases is known as the Edmundsonian Paradigm (Edmundson, 1969; Alfonesca and Rodriguez, 2003; Mani, 2001). We use variations on these features as well as a novel set of features based on third-party data.

Typically, news articles are written such that the first sentence summarizes the article. Thus, we include a binary feature $F(S_i)$ that equals 1 if S_i is the first sentence of the document: $F(S_i) = \delta_{i,1}$, where δ is the Kronecker delta function. This feature is used only for NetSum(b) and NetSum(1).

We include sentence position since we found in empirical studies that the sentence to best match highlight H_1 is on average 10% down the article, the sentence to best match H_2 is on average 20% down the article, and the sentence to best match H_3 is 31% down the article.[3] We calculate the position of S_i in document D as

$$Pos(S_i) = \frac{i}{\ell}, \tag{2}$$

where $i = \{1, \ldots, \ell\}$ is the sentence number and ℓ is the number of sentences in D.

We include the SumBasic score (Nenkova et al., 2006) of a sentence to estimate the importance of a sentence based on word frequency. We calculate the SumBasic score of S_i in document D as

$$SB(S_i) = \frac{\sum_{w \in S_i} p(w)}{|S_i|}, \tag{3}$$

where $p(w)$ is the probability of word w and $|S_i|$ is the number of words in sentence S_i. We calculate $p(w)$ as $p(w) = \frac{Count(w)}{|D|}$, where $Count(w)$ is the number of times word w appears in document D and $|D|$ is the number of words in document D. Note that the score of a sentence is the average probability of a word in the sentence.

We also include the SumBasic score over bigrams, where w in Eq 3 is replaced by bigrams and we normalize by the number of bigrams in S_i.

We compute the similarity of a sentence S_i in document D with the title T of D as the relative probability of title terms $t \in T$ in S_i as

$$Sim(S_i) = \frac{\sum_{t \in S_i} p(t)}{|S_i|}, \tag{4}$$

where $p(t) = \frac{Count(t)}{|T|}$ is the number of times term t appears in T over the number of terms in T.

The remaining features we use are based on third-party data sources. Previously, third-party sources such as WordNet (Fellbaum, 1998), the web (Jagalamudi et al., 2006), or click-through data (Sun et al., 2005) have been used as features. We propose using news query logs and Wikipedia entities to enhance features. We base several features on query terms frequently issued to Microsoft's news search engine http://search.live.com/news, and entities[4] found in the online open-source encyclopedia Wikipedia (Wikipedia.org, 2007). If a query term or Wikipedia entity appears frequently in a CNN document, then we assume highlights should include that term or entity since it is important on both the document and global level. Sentences containing query terms or Wikipedia entities therefore contain important content. We confirm the importance of these third-party features in Section 7.

We collected several hundred of the most frequently queried terms in February 2007 from the news query logs. We took the daily top 200 terms for 10 days. Our hypothesis is that a sentence with a higher number of news query terms should be a better candidate highlight. We calculate the average probability of news query terms q in S_i as

$$NT(S_i) = \frac{\sum_{q \in S_i} p(q)}{|q \in S_i|}, \tag{5}$$

[3] Though this is not always the case, as the sentence to match H_2 precedes that to match H_1 in 22.03% of documents, and the sentence to match H_3 precedes that to match H_2 in 29.32% of and precedes that to match H_1 in 28.81% of documents.

[4] We define an entity as a title of a Wikipedia page.

where $p(q)$ is the probability of a news term q and $|q \in S_i|$ is the number of news terms in S_i. $p(q) = \frac{Count(q)}{|q \in D|}$, where $Count(q)$ is the number of times term q appears in D and $|q \in D|$ is the number of news query terms in D.

We also include the sum of news query terms in S_i, given by $NT_+(S_i) = \sum_{q \in S_i} p(q)$, and the relative probability of news query terms in S_i, given by $NT_r(S_i) = \frac{\sum_{q \in S_i} p(q)}{|S_i|}$.

We perform term disambiguation on each document using an entity extractor (Cucerzan, 2007). Terms are disambiguated to a Wikipedia entity only if they match a surface form in Wikipedia. Wikipedia surface forms are terms that disambiguate to a Wikipedia entity and link to a Wikipedia page with the entity as its title. For example, "WHO" and "World Health Org." both refer to the World Health Organization, and should disambiguate to the entity "World Health Organization". Sentences in CNN document D that contain Wikipedia entities that frequently appear in CNN document D are considered important. We calculate the average Wikipedia entity score for S_i as

$$WE(S_i) = \frac{\sum_{e \in S_i} p(e)}{|e \in S_i|}, \quad (6)$$

where $p(e)$ is the probability of entity e, given by $p(e) = \frac{Count(e)}{|e \in D|}$, where $Count(e)$ is the number of times entity e appears in CNN document D and $|e \in D|$ is the total number of entities in CNN document D.

We also include the sum of Wikipedia entities, given by $WE_+(S_i) = \sum_{e \in S_i} p(e)$.

Note that all features except position features are a variant of SumBasic over different term sets. All features are computed over sentences where every word has been lowercased and punctuation has been removed after sentence breaking. We examined using stemming, but found stemming to be ineffective.

5 Evaluation

We evaluate the performance of NetSum using ROUGE and by comparing against a baseline system. For the first summarization task, we compare against the baseline of choosing the first three sentences as the block summary. For the second highlights task, we compare NetSum(n) against the baseline of choosing sentence n (to match highlight n). Both tasks are novel in attempting to match highlights rather than a human-generated summary.

We consider ROUGE-1 to be the measure of importance and thus train our model on ROUGE-1 (to optimize ROUGE-1 scores) and likewise evaluate our system on ROUGE-1. We list ROUGE-2 scores for completeness, but do not expect them to be substantially better than the baseline since we did not directly optimize for ROUGE-2.[5]

For every document in our corpus, we compare NetSum's output with the baseline output by computing ROUGE-1 and ROUGE-2 between the highlight block and NetSum and between the highlight block and the block of sentences. Similarly, for each highlight, we compute ROUGE-1 and ROUGE-2 between highlight n and NetSum(n) and between highlight n and sentence n, for $n = 1, 2, 3$. For each task, we calculate the average ROUGE-1 and ROUGE-2 scores of NetSum and of the baseline. We also report the percent of documents where the ROUGE-1 score of NetSum is equal to or better than the ROUGE-1 score of the baseline.

We perform all experiments using five-fold cross-validation on our dataset of 1365 documents. We divide our corpus into five random sets and train on three combined sets, validate on one set, and test on the remaining set. We repeat this procedure for every combination of train, validation, and test sets. Our results are the micro-averaged results on the five test sets. For all experiments, Table 3 lists the statistical tests performed and the significance of performance differences between NetSum and the baseline at 95% confidence.

6 Results: Summarization

We first find three sentences that, as a block, best match the three highlights as a block. NetSum(b) produces a ranked list of sentences for each document. We create a block from the top 3 ranked sentences. The baseline is the block of the first 3 sentences of the document. A similar baseline outper-

[5] NetSum can directly optimize for any measure by training on it, such as training on ROUGE-2 or on a weighted sum of ROUGE-1 and ROUGE-2 to optimize both. Thus, ROUGE-2 scores could be further improved. We leave such studies for future work.

System	Av. ROUGE-1	Av. ROUGE-2
Baseline	0.4642 ± 0.0084	0.1726 ± 0.0064
NetSum(b)	**0.4956** ± 0.0075	0.1775 ± 0.0066

Table 2: Results on summarization task with standard error at 95% confidence. Bold indicates significance under paired tests.

	ROUGE-1			ROUGE-2		
System	1	2	3	1	2	3
NetSum(b)	**x**	**x**	**x**	**x**	o	o
NetSum(1)	**x**	**x**	**x**	o	o	o
NetSum(2)	**x**	**x**	**x**	**x**	o	**x**
NetSum(3)	**x**	**x**	**x**	**x**	**x**	**x**

Table 3: Paired tests for statistical significance at 95% confidence between baseline and NetSum performance; 1: McNemar, 2: Paired t-test, 3: Wilcoxon signed-rank. "**x**" indicates pass, "o" indicates fail. Since our studies are pair-wise, tests listed here are more accurate than error bars reported in Tables 2–5.

forms all previous systems for news article summarization (Nenkova, 2005) and has been used in the DUC workshops (DUC, 2001).

For each block produced by NetSum(b) and the baseline, we compute the ROUGE-1 and ROUGE-2 scores of the block against the set of highlights as a block. For 73.26% of documents, NetSum(b) produces a block with a ROUGE-1 score that is equal to or better than the baseline score. The two systems produce blocks of equal ROUGE-1 score for 24.69% of documents. Under ROUGE-2, NetSum(b) performs equal to or better than the baseline on 73.19% of documents and equal to the baseline on 40.51% of documents.

Table 2 shows the average ROUGE-1 and ROUGE-2 scores obtained with NetSum(b) and the baseline. NetSum(b) produces a higher quality block on average for ROUGE-1.

Table 4 lists the sentences in the block produced by NetSum(b) and the baseline block, for the articles shown in Figure 1. The NetSum(b) summary achieves a ROUGE-1 score of 0.52, while the baseline summary scores only 0.36.

System	Sent. #	ROUGE-1
Baseline	S_1, S_2, S_3	0.36
NetSum(b)	S_1, S_7, S_{15}	**0.52**

Table 4: Block results for the block produced by NetSum(b) and the baseline block for the example article. ROUGE-1 scores computed against the highlights as a block are listed.

7 Results: Highlights

Our second task is to extract three sentences from a document that best match the three highlights in order. To accomplish this, we train NetSum(n) for each highlight $n = 1, 2, 3$. We compare NetSum(n) with the baseline of picking the nth sentence of the document. We perform five-fold cross-validation across our 1365 documents. Our results are reported for the micro-average of the test results. For each highlight n produced by both NetSum(n) and the baseline, we compute the ROUGE-1 and ROUGE-2 scores against the nth highlight.

We expect that beating the baseline for $n = 1$ is a more difficult task than for $n = 2$ or 3 since the first sentence of a news article typically acts as a summary of the article and since we expect the first highlight to summarize the article. NetSum(1), however, produces a sentence with a ROUGE-1 score that is equal to or better than the baseline score for 93.26% of documents. The two systems produce sentences of equal ROUGE-1 scores for 82.84% of documents. Under ROUGE-2, NetSum(1) performs equal to or better than the baseline on 94.21% of documents.

Table 5 shows the average ROUGE-1 and ROUGE-2 scores obtained with NetSum(1) and the baseline. NetSum(1) produces a higher quality sentence on average under ROUGE-1.

The content of highlights 2 and 3 is typically from later in the document, so we expect the baseline to not perform as well in these tasks. NetSum(2) outperforms the baseline since it is able to identify sentences from further down the document as important. For 77.73% of documents, NetSum(2) produces a sentence with a ROUGE-1 score that is equal to or better than the score for the baseline. The two systems produce sentences of equal ROUGE-1 score for 33.92% of documents. Under ROUGE-2, NetSum(2) performs equal to or better than the baseline

System	Av. ROUGE-1	Av. ROUGE-2
Baseline(1)	0.4343 ± 0.0138	0.1833 ± 0.0095
NetSum(1)	**0.4478** ± 0.0133	0.1857 ± 0.0085
Baseline(2)	0.2451 ± 0.0128	0.0814 ± 0.0106
NetSum(2)	**0.3036** ± 0.0117	**0.0877** ± 0.0107
Baseline(3)	0.1707 ± 0.0103	0.0412 ± 0.0069
NetSum(3)	**0.2603** ± 0.0133	**0.0615** ± 0.0075

Table 5: Results on ordered highlights task with standard error at 95% confidence. Bold indicates significance under paired tests.

System	Sent. #	ROUGE-1
Baseline	S_1	0.167
NetSum(1)	S_1	0.167
Baseline	S_2	0.111
NetSum(2)	S_1	**0.556**
Baseline	S_3	0.000
NetSum(3)	S_{15}	**0.400**

Table 6: Highlight results for highlight n produced by NetSum(n) and highlight n produced by the baseline for the example article. ROUGE-1 scores computed against highlight n are listed.

84.84% of the time. For 81.09% of documents, NetSum(3) produces a sentence with a ROUGE-1 score that is equal to or better than the score for the baseline. The two systems produce sentences of equal ROUGE-1 score for 28.45% of documents. Under ROUGE-2, NetSum(3) performs equal to or better than the baseline 89.91% of the time.

Table 5 shows the average ROUGE-1 and ROUGE-2 scores obtained for NetSum(2), NetSum(3), and the baseline. Both NetSum(2) and NetSum(3) produce a higher quality sentence on average under both measures.

Table 6 gives highlights produced by NetSum(n) and the highlights produced by the baseline, for the article shown in Figure 1. The NetSum(n) highlights produce ROUGE-1 scores equal to or higher than the baseline ROUGE-1 scores.

In feature ablation studies, we confirmed that the inclusion of news-based and Wikipedia-based features improves NetSum's peformance. For example, we removed all news-based and Wikipedia-based features in NetSum(3). The resulting performance moderately declined. Under ROUGE-1, the baseline produced a better highlight on 22.34% of documents, versus only 18.91% when using third-party features. Similarly, NetSum(3) produced a summary of equal or better ROUGE-1 score on only 77.66% of documents, compared to 81.09% of documents when using third-party features. In addition, the average ROUGE-1 score dropped to 0.2182 and the average ROUGE-2 score dropped to 0.0448. The performance of NetSum with third-party features over NetSum without third-party features is statistically significant at 95% confidence. However, NetSum still outperforms the baseline without third-party features, leading us to conclude that RankNet and simple position and term frequency features contribute the maximum performance gains, but increased ROUGE-1 and ROUGE-2 scores are a clear benefit of third-party features.

8 Conclusions

We have presented a novel approach to automatic single-document summarization based on neural networks, called NetSum. Our work is the first to use both neural networks for summarization and third-party datasets for features, using Wikipedia and news query logs. We have evaluated our system on two novel tasks: 1) producing a block of highlights and 2) producing three ordered highlight sentences. Our experiments were run on previously unstudied data gathered from CNN.com. Our system shows remarkable performance over the baseline of choosing the first n sentences of the document, where the performance difference is statistically significant under ROUGE-1.

9 Future Work

An immediate future direction is to further explore feature selection. We found third-party features beneficial to the performance of NetSum and such sources can be mined further. In addition, feature selection for each NetSum system could be performed separately since, for example, highlight 1 has different characteristics than highlight 2.

In our experiments, ROUGE scores are fairly low because a highlight rarely matches the content of a single sentence. To improve NetSum's performance, we must consider extracting content across sentence boundaries. Such work requires a system to produce abstract summaries. We hope to incorporate sentence simplification and sentence splicing and merging in a future version of NetSum.

Another future direction is the identification of "hard" and "easy" inputs. Although we report average ROUGE scores, such measures can be misleading since some highlights are simple to match and some are much more difficult. A better system evaluation measure would incorporate the difficulty of the input and weight reported results accordingly.

References

E. Alfonesca and P. Rodriguez. 2003. Description of the uam system for generating very short summaries at DUC–2003. In *DUC 2003: Document Understanding Conference, May 31–June 1, 2003, Edmonton, Canada*.

C. Aone, M. Okurowski, and J. Gorlinsky. 1998. Trainable scalable summarization using robust nlp and machine learning. In *Proceedings of the 17th COLING and 36th ACL*.

C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. 2005. Learning to Rank using Gradient Descent. In Luc De Raedt and Stefan Wrobel, editors, *ICML*, pages 89–96. ACM.

C.J.C. Burges, R. Ragno, and Q. Le. 2006. Learning to rank with nonsmooth cost functions. In *NIPS 2006: Neural Information Processing Systems, December 4-7, 2006, Vancouver, CA*.

CNN.com. 2007a. Cable news network. http://www.cnn.com/.

CNN.com. 2007b. Nigeria reports first human death from bird flu. http://edition.cnn.com/2007/WORLD/africa/01/31/nigeria.bird.flu.ap/index.html?eref=edition_world.

J. Conroy, J. Schlesinger, J. Goldstein, and D. O'Leary. 2004. Left-brain/right-brain multi-document summarization. In *DUC 2004: Document Understanding Workshop, May 6–7, 2004, Boston, MA, USA*.

S. Cucerzan. 2007. Large scale named entity disambiguation based on wikipedia data. In *EMNLP 2007: Empirical Methods in Natural Language Processing, June 28-30, 2007, Prague, Czech Republic*.

H. Daumé III and D. Marcu. 2005. Bayesian multi-document summarization at mse. In *Proceedings of MSE*.

DUC. 2001. Document understanding conferences. http://www-nlpir.nist.gov/projects/duc/index.html.

H.P. Edmundson. 1969. New methods in automatic extracting. *Journal for the Association of Computing Machinery*, 16:159–165.

G. Erkan and D. R. Radev. 2004a. Lexpagerank: Prestige in multi-document text summarization. In *EMNLP 2004: Empirical Methods in Natural Language Processing, 2004, Barcelona, Spain*.

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

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

T. Hirao, Y. Sasaki, H. Isozaki, and E. Maeda. 2002. Ntt's text summarization system for DUC–2002. In *DUC 2002: Workshop on Text Summarization, July 11–12, 2002, Philadelphia, PA, USA*.

J. Jagalamudi, P. Pingali, and V. Varma. 2006. Query independent sentence scoring approach to DUC 2006. In *DUC 2006: Document Understanding Conference, June 8–9, 2006, Brooklyn, NY, USA*.

H. Jing. 2002. Using hidden markov modeling to decompose human-written summaries. *Computational Linguistics*, 4(28):527–543.

J. Kupiec, J. Pererson, and F. Chen. 1995. A trainable document summarizer. *Research and Development in Information Retrieval*, pages 68–73.

P. Lal and S. Ruger. 2002. Extract-based summarization with simplification. In *DUC 2002: Workshop on Text Summarization, July 11–12, 2002, Philadelphia, PA, USA*.

Y. Le Cun, L. Bottou, G.B. Orr, and K.R. Müller. 1998. Efficient backprop. In *Neural Networks, Tricks of the Trade*, Lecture Notes in Computer Science LNCS 1524. Springer Verlag.

C.Y. Lin and E. Hovy. 2002. Automated multi-document summarization in neats. In *Proceedings of the Human Language Technology Conference (HLT2002)*.

C.Y. Lin. 2004a. Looking for a few good metrics: Automatic summarization evaluation — how many samples are enough? In *Proceedings of the NTCIR Workshop 4, June 2–4, 2004, Tokyo, Japan*.

C.Y. Lin. 2004b. Rouge: A package for automatic evaluation of summaries. In *WAS 2004: Proceedings of the Workshop on Text Summarization Branches Out, July 25–26, 2004, Barcelona, Spain*.

H. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

I. Mani. 2001. *Automatic Summarization*. John Benjamins Pub. Co.

R. Mihalcea and D. R. Radev, editors. 2006. *Textgraphs: Graph-based methods for NLP*. New York City, NY.

R. Mihalcea and P. Tarau. 2005. An algorithm for language independent single and multiple document summarization. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), October, 2005, Korea*.

R. Mihalcea. 2005. Language independent extractive summarization. In *ACL 2005: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, June, 2005, Ann Arbor, MI, USA*.

A. Nenkova, L. Vanderwende, and K. McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *SIGIR*, pages 573–580. ACM.

A. Nenkova. 2005. Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005), Pittsburgh, PA*.

B. Schiffman. 2002. Building a resource for evaluating the importance of sentences. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*.

J.T. Sun, D. Shen, H.J. Zeng, Q. Yang, Y. Lu, and Z. Chen. 2005. Web-page summarization using click-through data. In R. A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, and J. Tait, editors, *SIGIR*. ACM.

K. Toutanova, C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende. 2007. The pythy summarization system: Microsoft research at DUC2007. In *DUC 2007: Document Understanding Conference, April 26–27, 2007, Rochester, NY, USA*.

L. Vanderwende, H. Suzuki, and C. Brockett. 2006. Microsoft research at DUC2006: Task-focused summarization with sentence simplification. In *DUC 2006: Document Understanding Workshop, June 8–9, 2006, Brooklyn, NY, USA*.

Wikipedia.org. 2007. Wikipedia org. http://www.wikipedia.org.

W.T. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. 2007. Multi-document summarization by maximizing informative content words. In *IJCAI 2007: 20th International Joint Conference on Artificial Intelligence, January, 2007*.