A Simple, Concise, Query-based Approach to News Article Summarization Using Sentence Scoring

Megan Thornton
Computer Science Dept.
Brigham Young University
Provo, Utah, USA
mamberly@gmail.com

Sophie Gao
Computer Science Dept.
Brigham Young University
Provo, Utah, USA
sophiegao99@gmail.com

Yiu-Kai Ng
Computer Science Dept.
Brigham Young University
Provo, Utah, USA
ng@compsci.byu.edu

***Abstract*—With the increasing amount of information being digitized and the growing connectedness of the world, access to news and their intricated information is becoming more vital. Because of this growing need, creating news article summaries is becoming an increasingly important task to allow people to access essential information quickly. However, current summarization approaches require complex, taxing algorithms that cannot be seamlessly adopted for others to implement at the speed that we need. To remedy this, we have designed an elegant approach that allows the utilizing technology to quickly employ a multinomial classifier and sentence scoring of news articles to help with querying and filtering news to allow users to obtain a brief, efficient summary of what the articles entail. The multinomial classifier achieves very effective classification of news articles for summarization. Using various complementary sentence scores, we are able to accurately determine sentences that provide the most informative contents with respect to a user query Q. Through the use of this classification and summarization, we allow information of Q to be readily available. Experimental results verify that our news article summarization approach is *effective* and *efficient* in creating high-quality summaries. In addition, the conducted empirical study demonstrates that our summarization approach outperform a significant number of DUC summarizers.**

***Index Terms*—News; classes; summaries; sentence scoring**

I. Introduction

With the increasing access to technology and subsequent growing percentage of information being accessible online, the need for news to be reachable via smart phones or quickly on a tablet or laptop is becoming more vital. However, “the human attention span is continuously decreasing, and the amount of time a person wants to spend on reading is declining” [19]. We are motivated to increase the access to online news in a simple, concise way so that readers are offered a short summary on up-to-date events without feeling the burden of reading through multiple sources of information to be informed on the details about what is going on in the world, which is the design goal of our news article classification and summarization system.

In designing our system, the first problem to deal with is that we are bombarded by too much information. There is no doubt that a lot of information are available but the biggest problem is to find the desired one. People who want updates on sports, politics, or business don’t have the patient or time to search or scroll through pages of different news articles. To help resolve this problem, we first categorize news articles using a simple algorithm. Automating categorization of articles is the first step to help filter the mass of information so that we can easily query the articles and find what an user is likely looking for.

Once we have news articles on the same topic categorized, the next step is to filter particular articles that capture the desired information need specified by a user in a query Q. A subset of articles belonged to the same category that contain the information need indicated in Q are extracted to create a summary. Such a simple query-based categorization and summarization algorithm, however, is not presently available.

A summary takes on the lengthy, wordy, related articles and provides a few concise sentences that allow users looking for particular information to either obtain the desired information from those sentences or know which article they should access for additional information. Using summarization, we can cut down the time for retrieving desired information.

Our query-based summarization approach is performed by prioritizing sentences in retrieved articles based on their degree of relevance in the articles. The *degree of relevance* of a sentence S is determined by different sentence features, which include the (i) number of *significant words* in S, (ii) *degree of similarity* of S with respect to other sentences in the retrieved articles, (iii) *coverage of information* revealed in S, (iv) *non-redundancy* of S in terms of its content, and (v) temporal dimension of S. Such a summary captures the user’s information need in a concise way. We have verified that our summarizer generates high-quality summaries and significantly outperforms well-known summarizers on news articles.

II. Related Work

News article summarization is not a new concept. However, as society’s attention span gets lower [19], the need for summarization has become more vital to allow ordinary people to get the news they need in the time and way they need. As a result, there has been a fair amount of research on the subject. For the most part, all of these works follow a pattern of (i) text retrieval, (ii) significant sentence scoring, (iii) significant sentence selection, and (iv) summarization creation. Steps (ii) and (iii) are the steps that vary the most from study to study.

Singh et al. [19] use word-to-vector embedding to score sentences and bidirectional and unidirectional LTSM models to put the highest scored sentences together. Ou et al. [14] scored each sentence but applied a more event-based framework using supervised learning, sentence labels, and word extraction. In order to avoid redundancy, they labeled each sentence and maximized each. Word extraction allowed them to get the optimal ordering for Step (iii) listed above.

Many existing works focus specifically on the words in each sentence to score them. Malhotra et al. [11] used name entities to score each sentence and used sentence similarity to reduce redundant information but focused heavily on noun phrases, cardinal numbers, thematic terms, and anchor text. Bouras and Vassilis [2] also focused on noun retrieval and the words themselves in each sentence through pre-labeling and prioritizing nouns and words in the title for their scoring.

Hyoungil et al. [5] also used keywords found in the article's title, focusing on the frequency and position of the keywords in a sentence for their scoring. They, however, focused on summarization through mobile devices, so they tried to limit processing time. To do this, they used a Binary Independence Model and Statistical Relevance Weighting to estimate the importance of keyword candidates. They also use the title to find keywords and determined that a significant sentence was one with the frequency and position of keywords within it.

While some researchers cared about processing time and power abilities, others ignored them and utilized more complex algorithms. Nallapati et al. [12] used a Recursive Neural Network utilizing two RNNs, one for word attention and another for sentence attention. Singh et al. [18] scored their sentences using frequencies, syllables, proper nouns, capital letters, heading matching, sentence length, and sentence position but used various algorithms to prioritize and come up with the best summary. They used logistic regression, decision trees, random forest, neural networks, xgboost, and SVM.

III. Our News Article Summarization Approach

In contrast to existing approaches, we design a simple summarization system by avoiding the adaptation of complex machine learning algorithms so that the complicated process of partitioning, training, and testing data are not required. Instead, we consider features of sentences and words, such as *sentence significant factor*, to determine what information in a news article is significant and what to include/exclude in a summary.

A. The Document Classifier

Prior to creating a summary on news articles in response to a user query Q, we first extract existing news articles of the same category to which Q belongs. To accomplish this task, we classify a corpus of news articles into their respective categories, such as sports, technology, business, politics, and entertainment, which are pre-defined. With these categories, we can better filter and choose news articles for a user based on Q for summarization. We have chosen the Multinomial Naïve Bayes (MNB) classifier, since it is simple and effective, to perform the classification. The first step of the classification is to convert archive news articles into an event space.

1) Multinomial Event Space: An event summary is a set of possible events (or outcomes) from some process. A probability is assigned to each event in the event space, and the sum of the probabilities over all of the events in the event space must equal one. To create this event space, we total all the word frequencies for each document, i.e., news article in our case, in a category and create a vector where each dimension is the probability to find that word in the category.

$$P(c|d) = \frac{P(d|c)P(c)}{\sum_{c \in C} P(d|c)P(c)} = \frac{\Pi_{i=1}^{|n|} P(w_i|c)P(c)}{\sum_{c \in C} \Pi_{i=1}^{|n|} P(w_i|c)P(c)}$$

$$Class(d) = arg\ max_{c \in C} P(c|d), \text{ and } P(c) = \frac{N_c}{N} \quad (1)$$

where $P(c|d)$ is the probability of *document* d in class c, $P(d|c)$ is the probability that d is observed given c, $P(w_i|c)$ is the probability of the word w_i given c, $P(c)$ is the probability of observing c, C is the total number of classes, $|n|$ is the total number of distinct words in d, N_c is the number of training documents in c, and N is the number of training documents.

2) The Multinomial Classifier: The MNB classifier is a linear classifier, suitable for classification with discrete features, such as word counts for text classification. The multinomial distribution requires integer feature counts as defined below.

$$P(d|c) = \Pi_{w \in V} P(w|c)^{tf_{w,d}}, \text{ and } P(w|c) = \frac{tf_{w,c} + 1}{|c| + |V|} \quad (2)$$

where $tf_{w,d}$ is the frequency of occurrence of word w in document d, $tf_{w,c}$ is the frequency of occurrence of w in class c, $|c|$ is the number of words in the documents of c, and V is the number of distinct words in the corpus of documents.

B. Our Multi-documents Summarization Approach

Given a user query Q, we create a summary from news articles in a MNB-generated cluster C to which Q belongs by (i) downloading and cleaning the top-100 articles (see Section III-C1) retrieved from C based on their TF-IDF value with respect to Q, since 100 articles is an *ideal* set for creating summaries [4], (ii) identifying and associating all (pro)nouns in the retrieved top-100 articles with their referents (see Section III-C2), (iii) assigning each sentence S in the top-100 articles a score, denoted RS, which reflects the *relative significance* of S in capturing the key concepts covered in the top-100 articles with respect to Q according to a set of sentence features (see Section III-D), (iv) re-weighting the sentences in Step (iii) based on their temporal dimensions to capture the flow of events (explained in Section III-E), (v) choosing the top-M ($\geq$ 1) sentences (based on their re-weighted scores) from the top-100 articles, such that $(\sum_{i=1}^{M-1} L_i) < 9 \times Size$ and $(\sum_{i=1}^{M} L_i) \geq 9 \times Size$, where L_i is the number of words in a sentence i of the top-100 articles and $Size$ is approximately 10% of the number of words in the top-100 articles[1], (vi) clustering the M sentences to yield *sentence clusters* using the Hierarchical Agglomerative Clustering (HAC) algorithm based on word-correlation factors[2] (see Section III-D2), and (vii) selecting the top-N sentences from the sentence clusters

[1] We first select sentences with a total number of words of length 9 $\times Size$ and then trim the number of sentences to create a multi-news-article summary.

[2] *Word-correlation factors* quantify the *similarity* (*degree of closeness*) of two words in terms of their semantic meaning.

created in Step (vi) such that $\sum_{i=1}^{N-1} L_i < 250\ words$ and $\sum_{i=1}^{N} L_i \geq 250\ words$. If the number of sentences N to be selected for a summary is *less* than the number of created *sentence clusters*, then one from each of the sentence clusters with the highest re-weighted RS score is chosen.

We start with $9 \times Size$ words in creating a cluster summary, since Schlesinger et al. [17] claim that $9 \times Size$ words are required to generate a sufficient, distinct-content summary. For a generated multi-document summary, we (i) extract mutual content across the documents while avoiding *repetition*, (ii) capture *unique* (*related*, respectively) information in the documents, and (iii) allow the user to *click* on a sentence in the summary to view the corresponding news articles.

C. Pre-processing News Articles

After categorizing news articles into their respective classes and prior to analyzing the contents of the top-100 articles for creating a summary with respect to a user query, we proceed to remove non-essential information presented in the articles and perform the co-reference resolution on the text hereafter.

1) Document Cleaning: Text in each one of the top-100 retrieved news articles is first segmented into sentences using a short list of end-of-sentence punctuation marks and a list of common abbreviations, such as “i.e.”, to ensure reliable identification of sentence boundaries. Hereafter, each sentence is converted into a sequence of word tokens using the Conexor Parser[3]. For each word token, its *Doc(ument)_ID*, *Sent(ence)_ID*, *word form* (in the text), *stem* (created by the Porter stemming algorithm), and *creation date* are stored. The *Doc_ID* and *Sent_ID* identify the document from where the sentences are extracted, the *stem* of a word is used in different *similarity* formulas, and the *date* is used for re-weighting the sentences in a summary based on their *temporal dimension*.

2) Co-Reference Resolution: Co-reference resolution is the task of finding all expressions that refer to the same entity in a text, i.e., determining which (common) (pro)noun phrases refer to which real-world entity as given in a news article. Consider the sentence S, “I study computer science. It is a very demanding major.” In solving the co-reference problem, the pronoun “it” is replaced by “computer science”. In summarization, it is required to replace a (pro)noun in a sentence with its referencing entity, since sentences in the summary can lose their original orders and yield a *false* indication of what the (pro)noun refers to. Co-reference resolution does not only help with the summary creation, but it is important for computing a sentence relevance score. We used a github repository *neuralcoref*[4] to perform the co-reference resolution.

D. Sentence Scoring

Each sentence S in the top-100 retrieved news articles T is assigned a *relevance score*, denoted RS, which indicates its relative significance in capturing the contents of the articles in T. To compute the *relevance score* of S, we utilize the *features* presented between Section III-D1 and Section III-D6.

1) Significance Factor: In our news summarization approach, we rank each sentence in a top-100 news article to be summarized using a *significance factor* [3] and to select the top sentences for the summary. The *significance factor* for a sentence relays how significant a sentence is based on the significance of the words in the sentence. *Significant words* are defined as words of medium frequency in the document, where *medium* means that the frequency is between predefined high-frequency and low-frequency cutoff values. Intuitively, higher scores are given to sentences with more *significant words*. Given that $f_{d,w}$ is the *frequency* of word w in document d, then w is a significant word if (i) it is not a stopword, which eliminates the high-frequency, non-essential words, and (ii)

$$f_{d,w} \geq \begin{cases} 7 - 0.1 \times (25 - Z) & \text{if } Z < 25 \\ 7 & \text{if } 25 \leq Z \leq 40 \\ 7 + 0.1 \times (Z - 40) & \text{otherwise} \end{cases} \quad (3)$$

where Z is the number of sentences in d, and 25 and 40 are the low- and high-frequency cutoff values, respectively.

Once we know which words in a news article are significant, we can calculate the significance factor (SF) of a sentence S, which is defined as follows.

$$SF(S) = \frac{|significant\text{-}words|^2}{|S|} \quad (4)$$

where $|S|$ is the number of words in S and $|significant\text{-}words|$ is the number of significant words in S.

2) Sentence Similarity: In order to avoid choosing sentences that are (very) similar to be included in a summary, we prioritize sentences that are unique based on the *word-correlation factors* (wcf) of the words in each sentence of the collection of top-100 news articles T. The *degree of similarity* of a sentence S_i with respect to the others in T, denoted $Sim(S_i)$, indicates the relative degree of S_i in capturing the overall semantic *content* of T. We compute $Sim(S_i)$ using (i) the wcf of every word in S_i and words in each remaining sentence S_j in T and (ii) the *Odds ratio* = $\frac{p}{1-p}$ [8].

$$Sim(S_i) = \frac{\sum_{j=1, i\neq j}^{|S|} \sum_{k=1}^{n} \sum_{l=1}^{m} wcf(w_k, w_l)}{1 - \sum_{j=1, i\neq j}^{|S|} \sum_{k=1}^{n} \sum_{l=1}^{m} wcf(w_k,\ w_l)} \quad (5)$$

where $|S|$ is the number of sentences in T, n (m, respectively) is the number of words in S_i (S_j, respectively), and w_k (w_l, respectively) is a word in S_i (S_j, respectively).

The word-correlation factors, i.e., wcf, in our $word\text{-}similarity$ matrix, denoted $WS\text{-}matrix$, is a 54,625 × 54,625 symmetric matrix. The wcf of any two non-stop, stemmed words i and j in $WS\text{-}matrix$ is computed using the (i) *frequency* of co-occurrence and (ii) relative *distances* of i and j in each document in which they occur (as shown in Equation 6). WS-matrix was constructed using the Wikipedia collection[5] with 930,000 documents written by more than 89,000 authors on various topics and writing styles.

$$wcf(i,j) = \frac{\sum_{D \in Wiki} \left(\frac{\sum_{k_i \in D} \sum_{k_j \in D} \frac{1}{d(k_i,k_j)+1}}{N_i \times N_j} \right)}{|Wiki|} \quad (6)$$

[3] connexor.eu/technology/machinese/demo

[4] github.com/huggingface/neuralcoref

[5] en.wikipedia.org/wiki/Wikipedia:Database_download

where $|Wiki|$ is the number of documents in the Wikipedia collection, i.e., $Wiki$, $d(k_i, k_j)$ denotes the *distance* (i.e., the number of words in) between words i and j or their stems in a Wiki document D in which they co-occur, and N_i (N_j, respectively) is the number of times word i (j, respectively) and its *stems* appeared in D.

Compared with WordNet[6] in which each pair of words is not assigned a *similarity weight*, word-correlation factors offer a more sophisticated measure of word similarity.

3) Label-Sentence Similarity: *Label-Sentence Similarity* (LSS) measures the similarity between *sentences* in a news article and their *category labels*, such as *technology, business, politics, entertainment*, and *sports*. Using category labels, we extract the 10 most common nouns found in each category and consider these eleven words, i.e., the category label + the 10 most common words, to describe the labels of a category. Using LSS, we can determine how well a sentence S relays its category label L using the Vector Space Model (VSM) [3].

$$LSS(S) = sim(L, S) = \frac{\sum_{i=1}^{N} w_{i,S} \times w_{i,L}}{\sqrt{\sum_{i=1}^{N} w_{i,S}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,L}^2}} \quad (7)$$

where $w_{i,S}$ ($w_{i,L}$, respectively) is the weight of word i in S (L, respectively) and is defined as $w_{i,S} = tf(i,S) \times idf(i)$ ($w_{i,L} = tf(i,L) \times idf(i)$, respectively), and N is the total number of distinct keywords in the category labeled L.

The *higher* the LSS value of S is, the *higher* is the degree of S in reflecting the topic covered in the category labeled L. Based on LSS, we can prioritize words that focus on what the user actually cares about. For example, if a user is interested in reading an article about a sports team, we should avoid creating a summary that includes a sentence about some of the new technology in a sports stadium unrelated to the team, but information about (e.g., a game) the team involved instead.

4) Named Entity: An entity can be any word or a series of words that consistently references to the same concept. Named Entity Recognition (NER) focuses on (i) determining if a word w is part of a named entity and (ii) assigning w to the correct entity. In Natural Language Processing (NLP), this can be accomplished by categorizing each word w into a category, such as a person, organization, time, location, object, etc., assuming that w belongs to a name entity. To determine the named entity weight, denoted $NE(S)$, of a sentence S, we consider the number of named entities in S. By summing the number of named entities in S, we can prioritize sentences that are more informative, i.e., with more named entities, than others.

$$NE(S) = \frac{\sum_{i=1}^{|E|} f(E_i)}{f(E)} \quad (8)$$

where $|E|$ is the number of named entities in S, $f(E_i)$ is the frequency of occurrence of entity E_i in the top-100 news article collection T, and $f(E)$ is the sum of the frequency of occurrence of all named entities in T. A sentence that contains a named entity usually captures more useful information in a document than sentences that do not contain any [13].

5) Sentence Length: We penalize sentences that are either *too short* (with less than 15 words) or *too long* (with more than 30 words) [16]. Short sentences are detrimental to our summarization task, since they require some introduction or do not have as much information included, whereas long sentences have a higher probability of discussing multiple topics and can be found somewhere else in a document. We compute the *Sentence Length*, denoted SL, of a sentence S as

$$SL(S) = \begin{cases} -1 & \text{if } |S| < 15 \text{ or } |S| > 30 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $|S|$ is the number of (stop)words in S.

6) Sentence Location: In the literature, it is well-known that the *first sentence* in a document d is a *topic sentence*, which means that the information contained in the sentence is often the most engaging and important to the reader. The *last sentence* of d, on the other hand, is part of the conclusion that summarizes the points made throughout d and focuses on the significance of the information stated in d. Because these sentences have the highest likelihood of possessing useful information, we regard them higher than the remaining sentences in d. We assign the location value of sentence S as

$$SP(S) = \begin{cases} 1 & \text{if } S \text{ is the } 1^{st} \text{ sentence of the } 1^{st} \text{ paragraph} \\ & \text{or the } \textit{last} \text{ sentence of the } \textit{last} \text{ paragraph in } d \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

7) CombMNZ: Based on the respective scores of the features discussed in Sections III-D1 through III-D6 that are computed for each sentence of a news article, we rank all the sentences in the top-100 news articles accordingly. To compute a single score on which the cumulative effect of the six different features of each sentence are used for ranking propose, we rely on the CombMNZ model. CombMNZ is a well-established data fusion method for combining multiple ranked lists on an item I, i.e., a sentence in our case, to determine a *joint* ranking of I, a well-known rank-aggregation task.

$$CombMNZ_I = \sum_{c=1}^{N} I^c \times |I^c > 0|, \text{ where } I^c = \frac{S^I - I^c_{min}}{I^c_{max} - I^c_{min}} \quad (11)$$

where N is the number of ranked lists to be fused, which is *six* in our case, I^c is the normalized score of I in the ranked list c, and $|I^c > 0|$ is the number of non-zero, normalized scores of I in the lists to be fused. Prior to computing the ranking score of a sentence S, we transform the original scores in each feature ranked list of S into a *common range* [0, 1] such that S^I is the score of I in the ranked list c to be normalized, I^c_{max} (I^c_{min}, respectively) is the maximum (minimum, respectively) score available in c.

E. Temporal Dimension

The temporal dimension is the last step we consider to give priorities to (sentences in) news articles that are newer. The temporal dimension considers the number of days since a news article was published, multiplied by a decay rate. For our decay

[6] wordnet.princeton.edu

rate, we used 0.5, which can be adjusted based on the priority given to newer articles.

Temporal dimension is used, since the information captured in a set of news articles might have been dynamically changed over time, such as a disaster in news. An updated news article contains the most recent development, i.e., information, compared with its older editions. We account for the *temporal dimension* in a set of news articles by re-weighting each sentence in a news article based on its *timestamp*, the date when it was last updated. The relevant score weight of each sentence S, denoted $RS(S)$, is modified based on its temporal dimension weight, denoted $TD(S)$.

$$RS_T(S) = RS(S) \times TD(S) \tag{12}$$

where S is a sentence in the top-100 news articles, and $TD(S)$ is a time-based weight of S. The *earlier* a news article in T which includes S is published, the *smaller* the $TD(S)$ is. Since *exponential average* is extensively used in time-series prediction, we use the *decay rate formula* in computing $TD(S)$, which decreases the sentence weight exponentially based on *time* [20] and is defined as

$$TD(S) = DecayRate^{\frac{y-t}{24}} \tag{13}$$

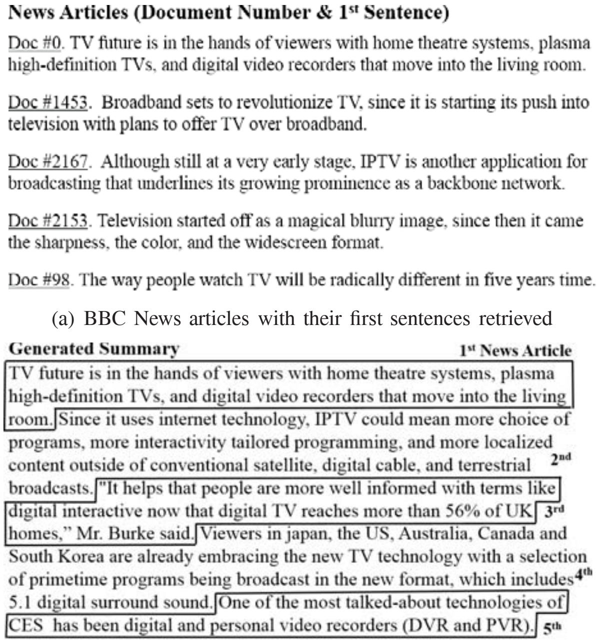
where y is the current time (i.e., day, hour, and minute), t is the publication time of the news article that includes S[7], $(y - t)$ is the time gap in hours, and *DecayRate* is a variable experimentally set to 0.5 [20].

F. Creating Sentence Clusters

Before selecting sentences for creating the summary Sum of the set of top-100 news articles T with respect to a query, we cluster the top-M (≥ 1) ranked sentences (based on their re-weighted RS scores) in T, where M is *nine* times the length of Sum, using the Hierarchical Agglomerative Clustering (HAC) algorithm [3]. The HAC algorithm initially assigns each sentence to a *singleton* sentence cluster. Hereafter, it repeatedly merges sentence clusters until a specified termination criterion is satisfied. Since the HAC algorithm relies on a *similarity metric* among sentences in any two sentence clusters for merging clusters, we use the *Sim* measure, as defined in Equation 5 with the first summation removed, to compute the similarity between any two sentences in two (intermediate) sentence clusters. To determine the termination criterion for HAC, we implement the algorithm in [1] to define the *optimal number* of *sentence clusters* in T to be generated by HAC, i.e.,

$$F(q) = \frac{\sum_{k=1}^{q} \sum_{i=1}^{m} \sum_{p=1}^{m} d_{ip} x_{ik} x_{pk}}{\sum_{i=1}^{m} \sum_{k=1}^{q} \sum_{p=1}^{m} \sum_{l=1, l \neq k}^{q} d_{ip} x_{ik} x_{pl}} \tag{14}$$

where q is the number of intermediate sentence clusters, m is the number of sentences in T, d_{ip} is the *Euclidean distance* between sentences i and p, and x_{ik} (x_{pk} and x_{pl}, respectively) is a Boolean variable that indicates whether sentence i is in the sentence cluster k (p in k and p in l, respectively).



News Articles (Document Number & 1st Sentence)

Doc #0. TV future is in the hands of viewers with home theatre systems, plasma high-definition TVs, and digital video recorders that move into the living room.

Doc #1453. Broadband sets to revolutionize TV, since it is starting its push into television with plans to offer TV over broadband.

Doc #2167. Although still at a very early stage, IPTV is another application for broadcasting that underlines its growing prominence as a backbone network.

Doc #2153. Television started off as a magical blurry image, since then it came the sharpness, the color, and the widescreen format.

Doc #98. The way people watch TV will be radically different in five years time.

(a) BBC News articles with their first sentences retrieved

Generated Summary **1st News Article**

TV future is in the hands of viewers with home theatre systems, plasma high-definition TVs, and digital video recorders that move into the living room. Since it uses internet technology, IPTV could mean more choice of programs, more interactivity tailored programming, and more localized content outside of conventional satellite, digital cable, and terrestrial **2nd** broadcasts. "It helps that people are more well informed with terms like digital interactive now that digital TV reaches more than 56% of UK **3rd** homes," Mr. Burke said. Viewers in japan, the US, Australia, Canada and South Korea are already embracing the new TV technology with a selection of primetime programs being broadcast in the new format, which includes **4th** 5.1 digital surround sound. One of the most talked-about technologies of CES has been digital and personal video recorders (DVR and PVR). **5th**

(b) The Summary generated using the news articles in Figure 1(a)

Fig. 1. The summary generated for the user query "TV future", using news articles in the *chosen* "Technology" category of the BBC news article dataset

We set the tolerable limit ε (= 0.005), which is determined experimentally, as the stopping criteria of the HAC algorithm. We start with the number of sentences in a sentence cluster to be created by HAC being two. Hereafter, we iteratively apply HAC to determine the new sentence clusters and compute the new $F(q)$ value in Equation 14, in which the numerator computes the *intra-document similarity*, whereas the denominator calculates the *inter-document similarity*. When $\frac{(F(q)-F(q+1))}{F(2)} < \varepsilon$, for $q > 2$, the HAC algorithm terminates.

G. Generating News Article Summaries

We select sentences from each *sentence cluster SC* created by HAC to be included in the summary Sum of the top-100 news articles T iteratively. The first sentence S to be chosen is from a SC with the highest (re-weighted) RS value in T and the sentence with the next highest RS value in each remaining sentence cluster is chosen in order. After the first round, we choose the next sentence $S^{'}$ from a SC in the same order, but with the *lowest similarity score* relative to its most-recently chosen sentence S in SC, which is computed as the *sum* of the word-correlation factors between each word in $S^{'}$ and S. Using this selection strategy, we ensure that selected sentences are *distinct* in contents, which avoids *redundancy*, and maximizes the *coverage* of the information included in Sum. The selection terminates whenever the length of the newly-selected sentence and other sentences that are already included in Sum exceeds *250* words in T, which is recommended by the *Text Analysis Conference (TAC)* for a multi-document summary[8]. Figure 1 shows the summary generated for the user query "TV future", using the news articles in the "Technology" category.

[7] If a sentence contains a date, then it overrides the publication time of the document, since it explicitly states the time of the information presented in the sentence.

[8] nist.gov/tac

TABLE I
DATASETS USED FOR EVALUATING THE QUALITY OF OUR SUMMARIES

Dataset	DUC 05	DUC 06	DUC 07
Number of Clusters	50	50	45
Number of Docs/Cluster	232	328	257
Data Source	TDT	AQUAINT	AQUAINT

IV. EXPERIMENTAL RESULTS

To assess the performance of our news article summarization approach, we first determined the datasets used for the empirical study and adopted a statistical approach to identify the ideal number of appraisers and queries required for validating the grammatical correctness, referential clarity, anti-redundancy, structure and coherence, and responsiveness quality of generated summaries. We also compared the summarized information between our approach, DUC summarizers, and Google and measured the time for generating our summaries.

A. The Datasets

We present the datasets used for analyzing the quality of our created news summaries, each of which captures the contents of the news articles retrieved in response to a user query.

Generic multi-document, news article summarization analysis has been one of the designated tasks of DUC 2005, DUC 2006, and DUC 2007, each of which is an open benchmark dataset created and archived by the Document Understanding Conference, DUC[9]. We used all three datasets for evaluating news article summaries created by us. Table I shows the properties of the three datasets, where TDT[10] and AQUAINT[11] are corpora from where the DUC datasets are created.

NIST assessors, who organized DUC and created each news article dataset as shown in Table I, selected various topics and chose a set of news articles relevant to each topic. Given a DUC topic T and a collection of news articles U belonged to T, a summarization approach to be evaluated is expected to create a brief (approximately 250 words from multiple documents), well-organized, and fluent summary that captures the key concepts covered in U on T. The summary is compared with the *reference summaries* of U, which were created by NIST assessors, to analyze its quality.

B. Number of Appraisers and Test Queries Used for the Controlled Experiments

We determine the ideal number of appraisers and test queries used in evaluating our news summarization approach so that the performance evaluation is *reliable* and *objective*.

1) The Number of Appraisers: In statistics, two types of errors, Types I and II, are defined [7]. Type I errors, also known as α errors or *false positives*, are the *mistakes* of *rejecting* a null hypothesis when it is true, whereas Type II errors, also known as β errors or *false negatives*, are the *mistakes* of *accepting* a null hypothesis when it is false. We apply the formula in [7] below to determine the ideal number of appraisers, n, which is dictated by the probabilities of occurrence of Types I and II errors, to evaluate news article summaries created by us.

$$n = \frac{(Z_{\frac{\alpha}{2}} + Z_{\beta})^2 \times 2\sigma^2}{\triangle^2} + \frac{(Z_{\frac{\alpha}{2}})^2}{2} \qquad (15)$$

where $\triangle$ is the *minimal expected difference* to compare our news summarization approach with NIST assessors, which is set to 1 in our study as we expect our news article summarization approach to generate high-quality news summaries as good as the ones created by NIST assessors; σ^2 is the *variance*[12] of the generated summaries, which is 2.15 in our study (see discussion in the next paragraph); α (β, respectively) denotes the probability of making a Type I (II, respectively) error, which is set to be 0.05 (0.20, respectively), and 1 - β determines the probability of a false null hypothesis that is correctly rejected, and Z is the value assigned to the standard *normal distribution* of generated summaries (see the explanations on setting the α and β values given later). Based on the standard normal distribution, when α = 0.05, $Z_{\frac{\alpha}{2}}$ = 1.96, and when β = 0.20, Z_β = 0.84.

We conducted an experiment using a randomly sampled 100 test queries on news extracted from the *AOL query log*[13] to determine the value of σ^2. We chose only 100 queries, since the *minimal expected difference* and *variance*, which are computed on a *simple random sample*, do not change with a larger sample set of queries. σ^2 is computed by averaging the sum of the square difference between the mean and the actual number of *useful* summaries[14] created for each one of the 100 test queries. We obtained $\sigma^2 = 2.15$ for news summaries.

The values of α and β are set to be 0.05 and 0.20, respectively, which imply that we have 95% *confidence* on the correctness of our analysis and that the *power* (i.e., probability of avoiding false negatives/positives) of our statistical study is 80%. According to [9], 0.05 is the commonly-used value for α, whereas 0.80 is a conventional value for 1 - β, and a test with β = 0.20 is considered to be statistically powerful. Based on the values assigned to the variables in Equation 15, the ideal number of appraisers used for our study is

$$n = \frac{(1.96 + 0.84)^2 \times 2 \times 2.15}{1^2} + \frac{1.96^2}{2} \cong 36 \qquad (16)$$

The results collected from the 36 appraisers are expected to be comparable with the results that are obtained by the actual population [7], i.e., web users who query web search engines.

2) The Number of Test Queries: To determine the ideal number of test queries to be included in the controlled experiments, we rely on two different variables: (i) the *average attention span* of an adult and (ii) the *average number of search queries* that a person often creates in one session when

[9] nlpir.nist.gov/projects/duc/

[10] projects.ldc.upenn.edu/TDT/

[11] ldc.upenn.edu/Catalog/docs/LDC2002T31/

[12] *Variance* is widely used in statistics, along with standard deviation (which is the square root of the variance), to measure the average dispersion of the scores in a distribution.

[13] The logs of AOL (gregsadetsky.com/aol-data/) include 50 million queries created by millions of AOL users between 03/01/06 and 05/31/06, and the AOL logs are available for public use.

[14] A summary is considered *useful* if it is of high quality (4 or 5 on a 5-point scale) as defined by DUC.

using a web search engine. As mentioned in [15], the average attention span of an adult is between twenty to thirty minutes. Furthermore, Jansen et al. [6], who have evaluated web users' behavior especially on (i) the amount of time web users spend on a web search engine, (ii) the average size of users' queries, and (iii) the average number of queries submitted by a user, estimate that the average number of queries created by each user in one session on a web search engine is about 2.8. Based on these studies, each appraiser was asked to evaluate our summarization approach using *three* queries, since evaluating the summaries on the retrieved results of each one of the three queries takes approximately 30 minutes, which falls into an adult time span. We randomly selected *108* (= 36×3) news queries from the DUC datasets for evaluating our summaries.

C. Performance Evaluation of Our Classification Approach

In evaluating the performance of our MNB classifier in categorizing news articles, we used the BBC news dataset[15] which includes a dozen categories and thousands of news articles in each category. The accuracy of our MNB classifier is *89%*, which indicates that almost 9 out of 10 articles are correctly classified and the classification task is highly accurate.

D. Performance Measures of Our Summarization Approach

We have recruited 36 students who were English major enrolled at our university as the appraisers to evaluate the *quality* of our news *summaries*. These student appraisers, who responded to our solicitation for conducting the empirical study through their teachers, were familiar with English writing. Majority of them have taken at least two English writing classes, including technical writing, and could provide unbiased evaluations on summarization. Out of the 36 students, 17 of them were graduating seniors, 12 of them were juniors, and the remaining ones were sophomore in academic standing.

Using the DUC 2005, 2006, and 2007 datasets and an evaluation guideline, which is a set of *quality questions* developed in 2001 [10], a summary created by a summarization system can be evaluated. These questions address the quality of *grammaticality*, *non-redundancy*, *referential clarity*, *structure* and *coherence*, and *responsiveness* of a created summary. These qualities are measured on a 5-point scale as suggested by DUC. For each appraiser, we provided (i) *three* of the 108 randomly-extracted DUC news queries, (ii) their respective summaries created by our summarization approach, and (iii) the set of quality questions for the appraiser to evaluate.

E. Performance Evaluation of Our News Article Summarizer Versus the Thirty DUC Summarizers

In this section, we present the experimental results that quantify the performance of our summarization approach on generating high-quality summaries. Each student appraiser evaluated the grammar, anti-redundancy, referential clarity, coherence, and responsiveness of a summary created by us.

We have collected the responses on the *quality questions* of each summary created by our summarization approach on news articles in the DUC datasets, i.e., DUC 2005-2007, which were provided by the 36 student appraisers who reviewed the summaries in response to the 108 test queries. The results are obtained by the comparisons of contents captured in summaries generated by our approach with the ones in the *reference summaries* created by the DUC experts on the same set of news articles. The results are depicted in Table II.

TABLE II
COMPARING THE QUALITY OF OUR NEWS ARTICLE SUMMARIES WITH THE REFERENCE SUMMARIES CREATED BY THE 30 DUC SUMMARIZERS

Quality	**Achieved by us**	**Outper-formed By**	**Out-perform**
Grammar	4.10	4	26
Anti-redundancy	4.62	1	29
Referential Clarity	4.85	0	30
Structure & Coherence	4.21	4	26
Responsiveness	4.38	3	27

As demonstrated in Table II, our summaries achieve the highest score (on a 5-point scale) on *referential clarity*, second highest on *non-redundancy*, forth on *responsiveness*, and fifth on *structure and coherence* and *Grammar*. The comparatively lower scores on grammar, besides structure and coherence, among the five quality measures are due to the fact that our summarization approach is *extractive*, not *abstractive* that rewrites sentences, and is not sophisticated in connecting (i.e., combining) extracted sentences in a summary. This is not a major drawback, since our summarization approach is ranked in the top 5 on each measure among the 30 summarizers.

F. Performance Evaluation of Our Summarizer Versus Google

We have also analyzed the performance evaluations provided by the thirty-six student appraisers who have compared the *summarized results* in locating desired information retrieved by our summarization approach and Google, respectively on each one of the 108 test queries (as described earlier). Once again each appraiser was asked to evaluate the same three test queries assigned to them. To accomplish the task, we created two web applications, App_1 and App_2, and posted them under our research lab website so that our student appraisers could provide their feedbacks on the results generated for their corresponding test queries.

For App_1, the application includes two pages in a panel, the *left* page displayed the (traditional) top-10 results generated by Google on one of the 108 test queries, whereas the *right* one is the summary created by the 10 documents shown on the left page. The purpose of this study is to analyze whether summaries generated our summarizer are really useful to the users who browse through search results generated by Google and enrich their search experiences. After submitting a test query and examine the results displayed on each (left/right) page, an appraiser responded to each of following questions:

1) "On which system did you spend less time locating the intended information?"
2) "Did the system on the *left* offer vital information not contained in the system on the right?"

For the first question, the responses are 14% for *Google*, 8% for our summarizer, and 78% for the same, whereas for the

[15] www.kaggle.com/sahilkirpekar/bbcnews-dataset

TABLE III
APPRAISERS' RESPONSES TO DIFFERENT TASKS POSTED AS QUERIES UNDER GOOGLE AND OUR NEWS ARTICLE SUMMARIZATION SYSTEM

Tasks (Posted as Queries on (Google & Our Summarizer)	Number of Responses	Prefer Google	Prefer *Ours*
Find News on a Newspaper	12	4	**8**
Find Information on Sports News	16	6	**10**
Find Answers to News Questions	6	2	**4**
Find Tools for News Publishing	8	**7**	1
Navigate a News Website	7	**7**	0

second question, 23% said 'Yes' and 77% said 'No.' Based on the responses, we conclude that the appraisers have found our summaries are comparable with the snippet results generated by Google in terms of *usefulness* and *informativeness*.

For App_2, the application requires the involved appraisers to (i) first *create a new set of pre-determined news search queries*, and (ii) *submit the query* to both Google and our summarization system. Hereafter, the appraisers were asked to answer the question, "Which system helped you perform this task faster?" The tasks, the number of responses for each task, and their answers to the predefined set of questions to be addressed are shown in Table III. The responses have verified that summaries created by our summarizer on results of news queries for different tasks were highly regarded by the appraisers than the results generated by Google, with the exception of the two tasks, "Find Tools for News Publishing" and "Navigate to a News Website." The results are anticipated, since summaries created by our summarizer include information on news but exclude URL links to download them, which are provided in the results generated by Google for its users to access. Moreover, finding the URL of a website W using its name provided by the user is a strength of Google, while a summary on W offers no such value.

Even though the empirical study of App_2 reflects that our summarizer cannot handle navigation-type news queries, an online report published by Wordtracker[16] on February 2, 2015 shows that out of the top 500 most popular query keywords created by web search engine users, only 51 of them include keywords explicitly specify a website, such as bbc.com, espn.com, and abcnews.go.com. The report illustrates that the percentage of navigation-typed web queries is not a dominating type of commonly-used web queries.

G. Query Processing Time of Our Summarization Approach

We have measured the *processing time* of creating a summary using our news article summarization approach based on the 108 queries from the DUC datasets. The processing time required to generate a summary is less than *2 seconds* on an average. While a user of our summarization system is viewing a summary generated for the news articles in response to a news query, access to individual news articles of the corresponding sentences in the summary are prepared in sequence behind the screen, which is a time-saving process.

Our summarization system is implemented on a HP workstation, running under Windows 10 with Intel Core i7-3770 3.4 GHz processors, 64 GB RAM, and a hard disk of 931 GB.

V. Conclusion

Online news websites play a vital role in educating and informing users with latest updates and current happenings around the globe. However, since people don't have time to read the entire posted news articles to find the ones that meet their personal information need, we have developed a new, efficient, and straightforward approach to perform multi-news-article summarization given a user query. We first adopt a simple multinomial classifier to categorize news articles based on their unique topic. Hereafter, we filter news articles that match the topic specified in a user query and prioritize the most informative sentences in top-ranked articles to generate a concise summary without requiring the use of cumbersome machine learning algorithms nor complex heuristic approaches. Experimental results have verified that summaries generated by using our summarization approach are high in quality and relevant to user-information needs as specified in user queries. Furthermore, our summarizer is efficient and outperforms well-known DUC summarizers.

References

[1] R. Alguliev and R. Alyguliev. Automatic Text Documents Summarization through Sentences Clustering. *J. Autom. Inf. Sci.*, 40:53–63, 2008.

[2] C. Bouras and T. Vassilis. Noun Retrieval Effect on Text Summarization and Delivery of Personalized News Articles to the User's Desktop. *Data & Knowledge Engineering*, 69(7):664–677, 2010.

[3] W. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison Wesley, 2010.

[4] D. Dunlavy, D. O'Leary, J. Conroy, and J. Schlesinger. QCS: A System for Querying, Clustering, and Summarizing Documents. *IPM*, 43:1588–1605, 2007.

[5] J. Hyoungil, K. Youngjoong, and S. Jungyun. Efficient Keyword Extraction and Text Summarization for Reading Articles on Smart Phone. *Computing & Informatics*, 34:779–794., 2015.

[6] B. Jansen, A. Spink, and T. Saracevic. Real Life, Real Users, and Real Needs: a Study and Analysis of User Queries on the Web. *IPM*, 36(2):207–227, 2000.

[7] B. Jones and M. Kenward. *Design and Analysis of Cross-Over Trials, 2nd Ed.* Chapman and Hall, 2003.

[8] P. Judea. *Probabilistic Reasoning in the Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

[9] L. Kazmier. *Schaum's Outline of Business Statistics*. McGraw-Hill, 2003.

[10] C. Lin and E. Hovy. From Single to Multi-Document Summarization: A Prototype System and its Evaluation. In *ACL*, pages 457–464, 2002.

[11] S. Malhotra and A. Dixit. An Effective Approach for News Article Summarization. *Computer Applications*, 76:5–10, 2013.

[12] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. In *CoNLL*, pages 280–290, 2016.

[13] S. Osinski. Improving Quality of Search Results Clustering with Approximate Matrix Factorisations. In *ECIR*, pages 167–178, 2006.

[14] S. Ou, C. Khoo, and D. Goh. Multi-Document Summarization of News Articles Using an Event-Based Framework. *Aslib*, 58(3):197–217, 2006.

[15] L. Rozakis. *Test Taking Strategies and Study Skills for the Utterly Confused*. McGraw Hill, 2002.

[16] B. Schiffman, A. Nenkova, and K. McKeown. Experiments in Multi-document Summarization. In *HLT*, pages 52–58, 2002.

[17] J. Schlesinger, D. Leary, and J. Conroy. Arabic/English Multi-document Summarization with CLASSY - The Past and the Future. In *CICLing*, pages 568–581, 2008.

[18] P. Singh, P. Chhikara, and J. Singh. An Ensemble Approach for Extractive Text Summarization. In *ic-ETITE*, pages 1–7, 2020.

[19] R. Singh, S. Khetarpaul, R. Gorantla, and S. Allada. SHEG: Summarization and Headline Generation of News Articles Using Deep Learning. *Neural Computing & Applications*, 33:3251–3265, 2021.

[20] P. Yu, X. Li, and B. Liu. Adding the Temporal Dimension to Search - A Case Study in Publication Search. In *WI*, pages 543–549, 2005.

[16] www.top-keywords.com/longterm.html