2-3-2014

Improving the Estimation of Word Importance for News Multi-Document Summarization - Extended Technical Report

Kai Hong
hongkai1@seas.upenn.edu

Ani Nenkova
nenkova@seas.upenn.edu

Recommended Citation

Kai Hong and Ani Nenkova, "Improving the Estimation of Word Importance for News Multi-Document Summarization - Extended Technical Report", . February 2014.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-14-02.

Improving the Estimation of Word Importance for News Multi-Document Summarization - Extended Technical Report

Abstract

In this paper, we propose a supervised model for ranking word importance that incorporates a rich set of features. Our model is superior to prior approaches for identifying words used in human summaries. Moreover we show that an extractive summarizer which includes our estimation of word importance results in summaries comparable with the state-of-the-art by automatic evaluation.

Disciplines

Computer Engineering | Computer Sciences

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-14-02.

Improving the Estimation of Word Importance for News Multi-Document Summarization - Extended Technical Report

Kai Hong
University of Pennsylvania
Philadelphia, PA, 19104
hongkai1@seas.upenn.edu

Ani Nenkova
University of Pennsylvania
Philadelphia, PA, 19104
nenkova@seas.upenn.edu

Abstract

We introduce a supervised model for predicting word importance that incorporates a rich set of features. Our model is superior to prior approaches for identifying words used in human summaries. Moreover we show that an extractive summarizer using these estimates of word importance is comparable in automatic evaluation with the state-of-the-art.

1 Introduction

In automatic extractive summarization, sentence importance is calculated by taking into account, among possibly other features, the importance of words that appear in the sentence. In this paper, we describe experiments on identifying words from the input that are also included in human summaries; we call such words **summary keywords**. We review several unsupervised approaches for summary keyword identification and further combine these, along with features including position, part-of-speech, subjectivity, topic categories, context and intrinsic importance, in a superior supervised model for predicting word importance.

One of the novel features we develop aims to determine the intrinsic importance of words. To this end, we analyze abstract-article pairs in the New York Times corpus (Sandhaus, 2008) to identify words that tend to be preserved in the abstracts. We demonstrate that judging word importance just based on this criterion leads to significantly higher performance than selecting sentences at random. Identifying intrinsically important words allows us to generate summaries without doing any feature computation on the input, equivalent in quality to the standard baseline of extracting the first 100 words from the latest article in the input. Finally, we integrate the schemes for assignment of word importance into a summarizer which greedily optimizes for the presence of important words. We show that our better estimation of word importance leads to better extractive summaries.

2 Prior work

The idea of identifying words that are descriptive of the input can be dated back to Luhn's earliest work in automatic summarization (Luhn, 1958). There keywords were identified based on the number of times they appeared in the input, and words that appeared most and least often were excluded. Then the sentences in which keywords appeared near each other, presumably better conveying the relationship between the keywords, were selected to form a summary.

Many successful recent systems also estimate word importance. The simplest but competitive way to do this task is to estimate the word probability from the input (Nenkova and Vanderwende, 2005). Another powerful method is log-likelihood ratio test (Lin and Hovy, 2000), which identifies the set of words that appear in the input more often than in a background corpus (Conroy et al., 2006; Harabagiu and Lacatusu, 2005).

In contrast to selecting a set of keywords, weights are assigned to all words in the input in the majority of summarization methods. Approaches based on (approximately) optimizing the coverage of these words have become widely popular. Earliest such work relied on TF*IDF weights (Filatova and Hatzivassiloglou, 2004), later approaches included heuristics to identify summary-worthy bigrams (Riedhammer et al., 2010). Most optimization approaches, however, use TF*IDF or word probability in the input as word weights (McDonald, 2007; Shen and Li, 2010; Berg-Kirkpatrick et al., 2011).

Word weights have also been estimated by supervised approaches, with word probability and location of occurrence as typical features (Yih et al., 2007; Takamura and Okumura, 2009; Sipos et al., 2012).

A handful of investigations have productively explored the mutually reinforcing relationship between word and sentence importance, iteratively re-estimating each in either supervised or unsupervised framework (Zha, 2002; Wan et al., 2007; Wei et al., 2008; Liu et al., 2011). Most existing work directly focuses on predicting sentence importance, with emphasis on the formalization of the problem (Kupiec et al., 1995; Celikyilmaz and Hakkani-Tur, 2010; Litvak et al., 2010). There has been little work directly focused on predicting keywords from the input that will appear in human summaries. Also there has been only a few investigations of suitable features for estimating word importance and identifying keywords in summaries; we address this issue by exploring a range of possible indicators of word importance in our model.

3 Data and Planned Experiments

We carry out our experiments on two datasets from the Document Understanding Conference (DUC) (Over et al., 2007). DUC 2003 is used for training and development, DUC 2004 is used for testing. These are the last two years in which generic summarization was evaluated at DUC workshops.

There are 30 multi-document clusters in DUC 2003 and 50 in DUC 2004, each with about 10 news articles on a related topic. The task is to produce a 100-word generic summary. Four human abstractive summaries are available for each cluster.

We compare different keyword extraction methods by the F-measure[1] they achieve against the gold-standard summary keywords. We do not use stemming when calculating these scores.

In our work, keywords for an input are defined as those words that appear in *at least* i of the human abstracts, yielding four gold-standard sets of keywords, denoted by G_i. $|G_i|$ is thus the cardinality of the set for the input. We only consider the words in the summary that also appear in the original input[2], with stopwords excluded[3]. Table 1 shows the average number of unique content words for the respective keyword gold-standard.

i	1	2	3	4
Mean $\|G_i\|$	102	32	15	6

Table 1: Average number of words in G_i

To better compare the performances of different keyword identification systems, we here employ a weighted approach similar to the pyramid method (Nenkova et al., 2007). We call our approach *keyword pyramid method*. This approach assigns different weights for the words which appear in different number of human models, while generating a single score as its output.

For the summarization task, we compare results using ROUGE (Lin, 2004). We report ROUGE-1, -2, -4 recall, with stemming and without removing stopwords. We consider ROUGE-2 recall as the main metric for this comparison due to its effectiveness in comparing machine summaries (Owczarzak et al., 2012). All of the summaries were truncated to the first 100 words by ROUGE[4].

We use two-tailed Wilcoxon signed-rank test to examine the statistical significance as advocated by Rankel et al. (2011) for both tasks, and consider differences to be significant if the p-value is less than 0.05.

4 Unsupervised Word Weighting

In this section we describe three unsupervised approaches of assigning importance weights to words. The first two are probability and log-likelihood ratio, which have been extensively used in prior work. We also apply a markov random walk model for keyword ranking, similar to (2004). We further discuss the correlations and differences between the words ranked by these three approaches. In the next section we describe a summarizer that uses these weights to form a summary and then describe our regression approach to combine these and other predictors in order to achieve more accurate predictions for the word importance in Section 7.

The task is to assign a score to each word in the input. The keywords extracted are thus the content

[1] 2*precision*recall/(precision+recall)

[2] On average 26.3% (15.0% with stemming) of the words in the four abstracts never appear in the input.

[3] We use the stopword list from the SMART system (Salton, 1971), augmented with punctuation and symbols.

[4] ROUGE version 1.5.5 with parameters: -c 95 -r 1000 -n 4 -m -a -l 100 -x

words with highest scores.

4.1 Word Probability (Prob)

The frequency with which a word occurs in the input is often considered as an indicator of its importance. The weight for a word is computed as $p(w) = \frac{c(w)}{N}$, where $c(w)$ is the number of times word w appears in the input and N is the total number of word tokens in the input.

4.2 Log-likelihood Ratio (LLR)

The log-likelihood ratio test (Lin and Hovy, 2000) compares the distribution of a word in the input with that in a large background corpus to identify topic words, where Gigaword corpus is used here (Graff et al., 2007). The test statistic has a χ^2 distribution, so a desired confidence level can be chosen to find a small set of topic words.

4.3 Markov Random Walk Model (MRW)

Graph methods have been successfully applied to weighting sentences for generic (Wan and Yang, 2008; Mihalcea and Tarau, 2004; Erkan and Radev, 2004) and query-focused summarization (Otterbacher et al., 2009).

Here instead of constructing a graph with sentences as nodes and edges weighted by sentence similarity, we treat the words as vertices, similar to Mihalcea and Tarau (2004). The difference in our approach is that the edges between the words are defined by syntactic dependencies rather than depending on the co-occurrence of words within a window of k. We use the Stanford dependency parser (Marneffe et al., 2006). In our approach, we consider a word w more likely to be included in a human summary when it is syntactically related to other (important) words, even if w itself is not mentioned often. The edge weight between two vertices is equal to the number of syntactic dependencies of any type between two words within the same sentence in the input. The weights are then normalized by summing up the weights of edges linked to one node.

We apply the Pagerank algorithm (Lawrence et al., 1998) on the resulting graph. We set the probability of performing random jump between nodes λ=0.15. The algorithm terminates when the change of node weight between iterations is smaller than 10^{-4} for all nodes. Word importance is equal to the final weight of its corresponding node in the graph.

4.4 Comparison of the Three Approaches

Here we show the correlations and differences between the three unsupervised weighting approaches: PROB, LLR and MRW. For each input on the DUC 2004 dataset, we compute the Spearman correlation between the weights assigned to words between any two unsupervised methods. This results in 50 scores for each pair of comparison. In Table 2, we show the median, maximum and minimum of Spearman correlation between these methods. Among the three comparisons, PROB and MRW are the most similar (Median = 0.6745), followed by PROB and LLR (Median = 0.4859), while LLR and MRW are the least similar (Median = 0.3298). All correlations are positive and significant, however they clearly show that the ranks assigned to words differ considerably from each other. Even the maximum correlation between the two most similar lists (PROB and MRW) is only 0.751.

	Median	Max	Min
PROB vs MRW	0.6745	0.7505	0.5883
PROB vs LLR	0.4859	0.6867	0.3019
LLR vs MRW	0.3298	0.5300	0.2062

Table 2: Spearman correlation between word weights assigned by the three unsupervised approaches on DUC 2004

We provide an example of the top 20 keywords identified by those three approaches for two of the inputs, *d30001t* and *d30049t* in Table 3. This table gives us an intuition about the words which tend to be ranked high using each approach. We also discuss the change of ranking for the same word from the output of these three approaches. Compared with word probability, we observe a drop in rank for the words which appear more often in the background corpus using log-likelihood ratio. For example, the rank of the words *party* and *government* dropped in *d30001t*; the rank of the words *clinton* and *united* dropped in *d30049t*. Meanwhile, the ranks of the relatively rare words improved while using log-likelihood ratio. For instance, the rank of *CPP* and *Funcinpec* improved in *d30001t* and the rank of *underground* improved in *d30049t*. When ranked by MRW, the words which are syntactically related with other high frequency terms have been rewarded, compared with PROB. Such examples include

Metric	d30001t	d30049t
PROB	hun, sen, party, ranariddh, opposition	north, korea, nuclear, clinton, korean, kim
	government, rainsy, sam, combodia, assembly	site, south, united, underground, states
	combodian, sihanouk, parliament, king, prince	weapons, koreans, president, officials, complex
	norodom, parties, cpp, national, form	construction, government, program, american
TS	hun, sen, ranariddh, rainsy, party, opposition	north, korea, nuclear, korean, underground
	combodia, sam, combodian, cpp, sihanouk	clinton, site, weapons, kim, koreans
	norodom, funcipec, assembly, government	south, full, made, give, kartman
	parliament, prince, parties, top, election	complex, construction, iraq, yongbyon, missile
MRW	sen, party, government, ranriddh, rainsy	korea, north, site, clinton, states
	hun, form, opposition, assembly, parties	called, korean, nuclear, koreans, program
	return, make, agreed, members, sihanouk	complex, u.s., kim, officials, give
	won, cpp, country, election, called	weapons, pay, south, government, facility

Table 3: Top 20 words from the input d30001t and d30049t by the three unsupervised approaches

verbs like *form, make, return, agreed* and nouns like *government, party* in the input *d30001*; as well as verbs such as *called, give* and nouns such as *site, program* in the input *d30049t*.

To better exploit the information provided by each approach, features from these three ranking methods are included in our supervised learning framework. We also show the performance of these three unsupervised approaches for both keyword identification and summarization in Section 8.

5 Summary Generation Process

In this section, we outline how summaries are generated by a greedy optimization system which selects the sentence with highest weight iteratively. This is the main process we use in all our summarization systems. For comparison we also use a summarization algorithm based on KL divergence.

5.1 Greedy Optimization Approach

Our algorithm extracts sentences by weighting them based on word importance. The approach is similar to the standard word probability baseline (Nenkova et al., 2006) but we explore a range of possibilities for assigning weights to individual words. Pseudo code of the procedure is shown in Algorithm 1. We denote the set of input sentences as D_k, the weighting function of words as $f_s(w)$. For each sentence s_i in the input set D_k, we compute the sentence weight $Score(s_i)$ by summing up the weights of all words, normalized by the number of words in the sentence. Then the sentences are sorted in descending order according to their scores and put into a queue Q. To create a summary, we iteratively dequeue one sentence, and append it to the current summary if the sentence is valid. Two conditions should be met for one sentence to be judged as valid, (1) the sentence should be at least 9 words in length (as in Erkan and Radev (2004)); (2) the sentence should not be redundant. A sentence is considered non-redundant if it is not similar to any sentences already in the summary, measured by cosine similarity on binary vector representations with stopwords excluded. We use the cut-off of 0.5 for cosine similarity. This value was tuned on the DUC 2003 dataset, by testing the impact of the cut-off value on the ROUGE scores for the final summary. Possible values ranged from 0.1 to 0.9

Algorithm 1 Greedy Optimization Summarizer

procedure GREEDYSUM(D_k, f_s)
  for each sentence s_i in D_k **do**
    $Score(s_i) \leftarrow \frac{\sum_{j=1}^{len(s_i)} f_s(w_j)}{len(s_i)}$
  end for
  Sort all Sentences according to $Score(s_i)$ in descending order, then storing into Q.
  $Sum_i \leftarrow \emptyset$. ▷ Summary set
  while $Len(Sum_i) \leq 100$ **do**
    $s_{cur} \leftarrow Q.dequeue()$
    if Valid(s_{cur}) **then**
      $Sum_i \leftarrow Sum_i \bigcup \{s_{cur}\}$
    end if
  end while
  return Sum_i
end procedure

with a step of 0.1.

5.2 KL Divergence Summarizer

The KLSUM summarizer (Haghighi and Vanderwende, 2009) aims at minimizing the KL divergence between the probability distribution over words estimated from the summary and the input respectively. This summarizer is a component of the popular topic model approaches (Daumé and Marcu, 2006; Celikyilmaz and Hakkani-Tür, 2011; Mason and Charniak, 2011) and achieves competitive performance with minimal differences compared to a full-blown topic model system. Denote Q as the unigram distribution of the input, P as the unigram distribution of the summary. The objective is to minimize the following:[5]

$$KL(P \parallel Q) = \sum_{\text{w in summary}} P(w) \cdot \ln \frac{P(w)}{Q(w)}$$

Since optimizing $KL(P \parallel Q)$ is exponential to the number of sentences in the input, we apply the Greedy-KL algorithm (Haghighi and Vanderwende, 2009). Denote D_k as the input, for each iteration we pick the sentence $s_i \in D_k$ which would minimize the KL divergence between the current version of the summary and the probability over the input. While selecting the sentence, again we check the validity, similar to Section 5.1. Pseudo code of the summarizer is shown in Algorithm 2.

Algorithm 2 KLSUM Summarizer

procedure KLSUM(D_k)
  Compute the unigram distribution Q for D_k.
  while $Len(Sum_k) \leq 100$ **do**
    j = $argmin_i\ KL(P_i \parallel Q)$ ▷ P_i is the unigram distribution for $Sum_k \bigcup s_i$, among all valid $s_i \in S$
    $Sum_k \leftarrow Sum_k \bigcup \{s_j\}$
  end while
  return Sum_k
end procedure

[5] This is different from the formula in Haghighi and Vanderwende (2009), as we minimize $KL(P \parallel Q)$ instead of $KL(Q \parallel P)$. The summarizer we present here is the one with better performance.

6 Global Indicators from NYT

Some words evoke topics that are of intrinsic interest to people. Here we search for global indicators of word importance regardless of particular input.

6.1 Global Indicators of Word Importance

We analyze a large corpus of original documents and corresponding summaries in order to identify words that consistently get included in or excluded from the summary. In the 2004-2007 NYT corpus, many news articles have abstracts along with the original article, which makes it an appropriate resource to do such analysis. We identified $160,001$ abstract-original pairs in the corpus. From these, we generate two language models, one estimated from the text of all abstracts (LM_A), the other estimated from the corpus of original articles (LM_G). We use *SRILM* (Stolcke, 2002) with Ney smoothing.

We denote the probability of word w in LM_A as $Pr_A(w)$, the probability in LM_G as $Pr_G(w)$, and calculate the difference $Pr_A(w) - Pr_G(w)$ and the ratio $Pr_A(w)/Pr_G(w)$ to capture the change of probability. In addition, we calculate KL-like weighted scores for words which reflect both the change of probabilities between the two samples and the overall frequency of the word. Here we calculate both $KL(A \parallel G)$ and $KL(G \parallel A)$. The formulas are shown below. Words with high values for the former score are favored in the summaries because they have higher probability in the abstracts than in the originals and have relatively high probability in the abstracts. The later score is high for words that are often not included in summaries.

$$KL(A \parallel G)(w) = Pr_A(w) \cdot \ln \frac{Pr_A(w)}{Pr_G(w)}$$

$$KL(G \parallel A)(w) = Pr_G(w) \cdot \ln \frac{Pr_G(w)}{Pr_A(w)}$$

Table 4 shows examples of the global information captured from the three types of scores—$KL(A \parallel G)$, $KL(G \parallel A)$ and $Pr_A(w)$—listing the 30 content words with higher scores for each type. Here the stopwords are excluded. Words that tend to be used in the summaries, characterized by high $KL(A \parallel G)$ scores, include locations *(York, NJ, Iraq)*, people's names and titles *(Bush, Sen, John)*, some abbreviations *(pres, corp, dept)* and verbs of

Metric	Top-30 words
$KL(A \parallel G)(w)$	photo, photos, pres, article, column, reviews, letter, York, Sen, NY, discusses drawing, op-ed, holds, Bush, correction, editorial, dept, city, NJ map, corp, graph, contends, Iraq, John, dies, sec, state, comments
$KL(G \parallel A)(w)$	Mr, Ms, p.m., lot, Tuesday, CA, Wednesday, Friday, told, Monday, time, a.m. added, thing, Sunday, hings, asked, good, night, Saturday, nyt, back senator, wanted, kind, Jr., Mrs, bit, looked, wrote
$Pr_A(w)$	photo, photos, article, York, column, letter, Bush, state, reviews, million, American pres, percent, Iraq, year, people, government, John, years, company, correction national, federal, officials, city, drawing, billion, public, world, administration

Table 4: Top 30 words derived by three metrics from the NYT corpus

conflict *(contends, dies)*. On the other hand, from $KL(G \parallel A)$, we can see that it is unlikely for writers to include courtesy titles *(Mr, Ms, Jr.)* and relative time reference in summaries. The words with high $Pr_A(w)$ scores overlaps with those ranked highly by $KL(A \parallel G)$ to some extent, but also includes a number of generally frequent words which appeared often both in the abstracts and original texts, such as *million* and *percent*.

6.2 Blind Sentence Extraction

In later sections we include the measures of global word importance as a feature of our regression model for predicting word weights for summarization. Before turning to that, however, we report the results of an experiment aimed to confirm the usefulness of these features. We present a system, BLIND, which uses only weights assigned to words by $KL(A \parallel G)$ from NYT, without doing any analysis of the original input. We rank all non-stopword words from the input according to this score. The top k words are given weight 1, while the others are given weight 0. The summaries are produced following the greedy procedure described in Section 5.1.

Systems	R-1	R-2	R-4
RANDOM	30.32	4.42	0.36
BLIND (80 keywords)	30.77	5.18	0.53
BLIND (300 keywords)	32.91	5.94	0.61
LASTESTLEAD	31.39	6.11	0.63
FIRST-SENTENCE	34.26	7.22	1.21

Table 5: Blind sentence extraction system, compared with three baseline systems (%)

Table 5 shows that the BLIND system has R-2 recall of 0.0594 using the top 300 keywords, significantly better than picking sentences from the input randomly. It also achieves comparable performance with the baseline in DUC 2004, formed by selecting the first 100 words from the latest article in the input (LASTESTLEAD). However it is significantly worse than another baseline of selecting the first sentences from the input. Table 6 gives sample summaries generated by these three approaches. These results confirm that the information gleaned from the analysis of NYT abstract-original pairs encodes highly relevant information about important content independent of the actual text of the input.

7 Regression-Based Keyword Extraction

Here we introduce a logistic regression model for assigning importance weights to words in the input. Crucially, this model combines evidence from multiple indicators of importance. We have at our disposal abundant data for learning because each content word in the input can be treated as a labeled instance. There are in total $32,052$ samples from the 30 inputs of DUC 2003 for training, $54,591$ samples from the 50 inputs of DUC 2004 for testing. For a word in the input, we assign label 1 if the word appears in at least one of the four human summaries for this input. Otherwise we assign label 0.

In the rest of this section, we describe the rich variety of features included in our system. We also analyze and discuss the predictive power of those features by performing two-tailed Wilcoxon signed-rank test on the DUC 2003 dataset. There are in total $9,261$ features used, among them $1,625$ are significant (p-value < 0.05). We rank these features in increasing p-values derived from Wilcoxon test. Apart from the widely used

Random Summary It was sunny and about 14 degrees C(57 degrees F) in Tashkent on Sunday. The president is a strong person, and he has been through far more difficult political situations, Mityukov said, according to Interfax. But Yeltsin's aides say his first term, from 1991 to 1996, does not count because it began six months before the Soviet Union collapsed and before the current constitution took effect. He must stay in bed like any other person, Yakushkin said. The issue was controversial earlier this year when Yeltsin refused to spell out his intentions and his aides insisted he had the legal right to seek re-election.
NYT Summary from global keyword selection, $KL(A \parallel G)$, k = 300 Russia's constitutional court opened hearings Thursday on whether Boris Yeltsin can seek a third term. Yeltsin's growing health problems would also seem to rule out another election campaign. The Russian constitution has a two-term limit for presidents. Russian president Boris Yeltsin cut short a trip to Central Asia on Monday due to a respiratory infection that revived questions about his overall health and ability to lead Russia through a sustained economic crisis. The upper house of parliament was busy voting on a motion saying he should resign. The start of the meeting was shown on Russian television.
First Sentence Generated Summary President Boris Yeltsin has suffered minor burns on his right hand, his press office said Thursday. President Boris Yeltsin's doctors have pronounced his health more or less normal, his wife Naina said in an interview published Wednesday. President Boris Yeltsin, on his first trip out of Russia since this spring, canceled a welcoming ceremony in Uzbekistan on Sunday because he wasn't feeling well, his spokesman said. Doctors ordered Russian President Boris Yeltsin to cut short his Central Asian trip because of a respiratory infection and he agreed to return home Monday, a day earlier than planned, officials said.

Table 6: Summaries generated from the Random, Blind Extraction and First Sentence systems

features of word frequency and positions, some other less explored features are highly significant.

7.1 Frequency Features

We use the probability, LLR chi-square statistic value and MRW scores as features. Since prior work has demonstrated that for LLR weights in particular, it is useful to identify a small set of important words and ignore all other words in summary selection (Gupta et al., 2007), we use a number of keyword indicators as features. For these indicators, the value of feature is 1 if one word is ranked within top k_i, 0 otherwise. Here k_i are preset cutoffs[6]. These cutoffs capture different possibilities for defining the keywords in the input. We also add the number of input documents that contain the word as a feature. There are a total of 100 features in this group, all of which are highly significant, ranked among the top 200.

7.2 Standard features

We now describe some standard features which have been applied in prior work on summarization.

Word Locations: Especially in news articles, sentences that occur at the beginning are often the most important ones. In line with this observation, we calculate several features related to the position in which a word appears. We first compute the relative positions for word tokens, where the tokens are numbered sequentially in order of appearance in each document in the input. The relative position for one word token is therefore its corresponding number divided by total number of tokens minus one in the document, e.g., 0 for the first token, 1 for the last token. For each word, we calculate its *earliest first location*, *latest last location*, *average location* and *average first location* for tokens of this word across all documents in the input. In addition we have a binary feature indicating if the word appears in the first sentence and the number of times it appears in a first sentence among documents in one input. There are 6 features in this group. All of them are very significant, ranked within the top 100.

Word type: These features include Part of Speech (POS) tags, Name Entity (NE) labels and capitalization information. We use the Stanford POS-Tagger (Toutanova et al., 2003) and Name Entity Recognizer (Finkel et al., 2005). We have one feature corresponding to each possible POS and NE tag. The value of this feature is the proportion of occurrences of the word with this tag; in most cases only one feature gets a non-zero value. We have two features which indicate if one word has been capitalized and the ratio of its capitalized occurrences.

Most of the NE features (6 out of 8) are significant: there are more *Organizations* and *Locations* but fewer *Time* and *Date* words in the human summaries. Of the POS tags, 11 out of 41 are significant: there are more nouns (*NN, NNS, NNPS*); fewer verbs (*VBG, VBP, VB*) and fewer cardinal numbers in the abstracts compared to the input. Capitalized words also tend to be included in human summaries.

KL: Prior work has shown that having estimates of sentence importance can also help in estimating

[6] 10, 15, 20, 30, 40, $\cdots$, 190, 200, 220, 240, 260, 280, 300, 350, 400, 450, 500, 600, 700 (in total 33 values)

word importance (Wan et al., 2007; Liu et al., 2011; Wei et al., 2008). The summarizer based on KL-divergence assigns importance to sentences directly, in a complex function according to the word distribution in the sentence. Therefore, we use these summaries as potential indicators of word importance. We include two features here, the first one indicates if the word appears in a KLSUM summary of the input, as well as a feature corresponding to the number of times the word appeared in that summary. Both of the features are highly significant, ranked within the top 200.

7.3 NYT-weights as Features

We include features from the relative rank of a word according to $KL(A \parallel G)$, $KL(G \parallel A)$, $Pr_A(w) - Pr_G(w)$, $Pr_A(w)/Pr_G(w)$ and $Pr_A(w)$, derived from the NYT as described in Section 6. If the rank of a word is within top-k or bottom-k by one metric, we would label it as 1 (one feature dimension for top-k, one feature dimension for bottom-k), where k is selected from a set of pre-defined values[7]. We have in total 70 features in this category, of which 56 are significant, 47 having a p-value less than 10^{-7}. The predictive power of those global indicators are only behind the features which indicates frequency and word positions.

7.4 Unigrams

This is a binary feature corresponding to each of the words that appeared at least twice in the training data. The idea is to learn which words from the input tend to be mentioned in the human summaries. There are in total $8,691$ unigrams, among which $1,290$ are significant. Despite the high number of significant unigram features, most of them are not as significant as the more general ones we described so far. It is interesting to compare the significant unigrams identified in the DUC abstract/input data with those derived from the NYT corpus. Unigrams that tend to appear in DUC summaries include *president*, *government*, *political*. We also find the same unigrams among the top words from NYT corpus according to $KL(A \parallel G)$. As for words unlikely to appear in summaries, we see *Wednesday*, *added*, *thing*, etc, which again rank high according to $KL(G \parallel A)$.

[7] 100, 200, 500, 1000, 2000, 5000, 10000 in this case.

7.5 Dictionary Features: MPQA and LIWC

Unigram features are notoriously sparse. To mitigate the sparsity problem, we resort to more general groupings to words according to salient semantic and functional categories. We here employ two hand-crafted dictionaries, MPQA for subjectivity analysis and LIWC for topic analysis.

The MPQA dictionary (Wiebe and Cardie, 2005) contains words with different polarities (positive, neutral, negative) and intensities (strong, weak). The combinations correspond to six features. It turns out that words with strong polarity, either positive or negative, are seldomly included in the summaries. Most strikingly, the p-value from significance test for the strong negative words is less than 10^{-4}—these words are rarely included in summaries. There is no significant difference on weak polarity categories.

Another dictionary we use is LIWC (Tausczik and Pennebaker, 2007), which contains manually constructed dictionaries for multiple categories of words. The value of the feature is 1 for one word if the word appears in the particular dictionary for the category. 34 out of 64 LIWC features are significant. Interesting categories which appear at higher rate in summaries include events about death, anger, achievements, money and negative emotions. Those that appear at lower rate in the summaries include auxiliary verbs, hear, pronouns, negation, function words, social words, swear, adverbs, words related to families, etc.

7.6 Context Features

We use context features here, based on the assumption that context importance around a word affects the importance of this word. For context we consider the words before and after the target word. We extend our feature space by calculating the weighted average of the feature values of the context words. For word w, we denote L_w as the set of words before w, R_w as the set of words after w. We denote the feature for one word as $w.f_i$, the way of calculating the newly extended word-before feature $w.l_{f_i}$ could be written as:

$$w.l_{f_i} = \sum_i p(w_l) \cdot w_l.f_i, \; \forall w_l \in L_w$$

Here $p(w_l)$ is the probability word w_l appears before w among all words in L_w.

For context features, we calculate the weighted average of the most widely used basic features, including frequency, location and capitalization for surrounding contexts. There are in total 220 features of this kind, among which 117 are significant, 74 having a p-value less than 10^{-4}.

8 Experiments

The performance of our logistic regression model is evaluated on two tasks: keyword identification and extractive summarization. We name our system REGSUM.

8.1 Regression for Keyword Identification

For each input, we define the set of keywords as the top k words according to the scores generated from different models. We compare our regression system with three unsupervised systems: PROB, LLR, MRW. To show the effectiveness of new features, we compare our results with a regression system trained only on word frequency and location related features described in Section 7. Those features are the ones standardly used for ranking the importance of words in recent summarization works (Yih et al., 2007; Takamura and Okumura, 2009; Sipos et al., 2012), and we name this system REGBASIC.

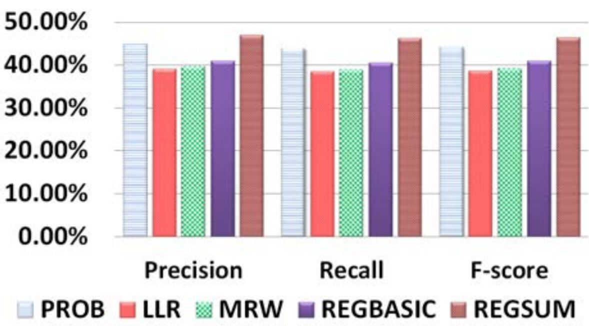

Figure 1: Precision, Recall and F-score of keyword identification, 100 words selected, G_1 as gold-standard

Figure 1 shows the performance of systems when selecting the 100 words with highest weights as keywords. Each word from the input that appeared in any of the four human summaries is considered as a gold-standard keyword. Among the unsupervised approaches, word probability identifies keywords better than LLR and MRW by at least 4% on F-score. REGBASIC does not give better performance at keyword identification compared with PROB, even though it includes location information. Our system gets 2.2% F-score improvement over PROB, 5.2% over REGBASIC, and more improvement over the other approaches. All of these improvements are statistically significant by Wilcoxon test.

Table 7 shows the performance of keyword identification for different G_i and different number of keywords selected. The regression system has no advantage over PROB when identifying keywords that appeared in all of the four human summaries. However our system achieves significant improvement for predicting words that appeared in at least one (G_1) or at least two (G_2) human summaries.

8.2 Keyword Pyramid Method

In Section 8.1, we compare the keywords extracted using different approaches with the gold standard keywords that appear in at least i human abstracts (G_i). Doing evaluation in this way does not differentiate between the keywords which appear in different number of human models. Inspired by the pyramid method (Nenkova et al., 2007), we employ a weighted approach for evaluating the identification of summary keywords.[8] This method takes into account the fact that words appear in more human summaries are of greater importance. Moreover, it generates a single score as its output, which makes the evaluation simpler. We name our approach as *keyword pyramid method*.

Consider an input with n manually generated summaries. For each word w, we assign a weight $t(w) = i$, representing the number of summaries this word appears in. All of the summary keywords can be partitioned into tiers based on $t(w)$ in this way. We here denote the tiers as T_i ($1 \leq i \leq n$), where T_i includes all words that appear in exactly i human models. We name the tiers of keywords for the input here as our keyword pyramid.

Consider we are given a word list L returned by a word weighting approach. Denote w_i as the word ranked the i-th in the list. For the words which have appeared in the keyword pyramid, they are assigned weight $t(w_i)$ according to the number of human models they appear in. The words which have never appeared in human models are assigned weight zero. Denote L_k as the set of top k words in the list L, the coverage weight of these

[8]We would like to thank an anonymous reviewer for suggesting us doing this evaluation.

G_i	#words	PROB	LLR	MRW	REGBASIC	**REGSUM**
G_1	80	43.6	37.9	38.9	39.9	45.7
G_1	100	44.3	38.7	39.2	41.0	46.5
G_1	120	44.6	38.5	39.2	40.9	46.4
G_2	30	47.8	44.0	42.4	47.4	50.2
G_2	35	47.1	43.3	42.1	47.0	49.5
G_2	40	46.5	42.4	41.8	46.4	49.2
G_3	10	51.2	46.2	43.8	46.9	50.2
G_3	15	51.4	47.5	43.7	49.8	52.9
G_3	20	49.7	47.6	42.5	49.3	51.5
G_4	5	50.0	48.8	44.9	43.6	45.1
G_4	6	51.4	46.9	43.7	45.2	47.6
G_4	7	50.9	48.2	43.7	45.8	47.8

Table 7: Keyword identification F-score (%) for different G_i and different number of words selected.

words can be computed as:

$$Coverage(L_k) = \sum_{i=1}^{k} t(w_i)$$

Similar to the pyramid method, we normalize $Coverage(L_k)$ by the optimal coverage weight that could be achieved with k words selected. The optimal score is computed from a list where all of the words from T_n are ranked first, followed by all of the words from T_{n-1}, etc. The formula of computing the optimal coverage weight is:

$$Opt_k = \sum_{i=s+1}^{n} i \cdot |T_i| + s \cdot (k - \sum_{i=s+1}^{n} |T_i|)$$

$$\text{Here } s = max_i(\sum_{i=1}^{n} |T_i| \geq k)$$

The normalized keyword pyramid score is simply $\frac{Coverage(L_k)}{Opt_k}$, ranges between 0 and 1.

We now compare the performance of the five systems using our *keyword pyramid method*. Figure 2 illustrates the normalized keyword pyramid scores changing with the number of keywords extracted. When k is equal to ten, our regression system performs similar to using word probability. However, our system achieves significantly better performance than all of the others when $k \geq 20$. The improvement ranges between 2.1% to 3.2% for all of the k we test on between 20 and 100. Word probability gives the best performance among the three unsupervised approaches. This result is consistent with the ones in Figure 1 and Table 7.

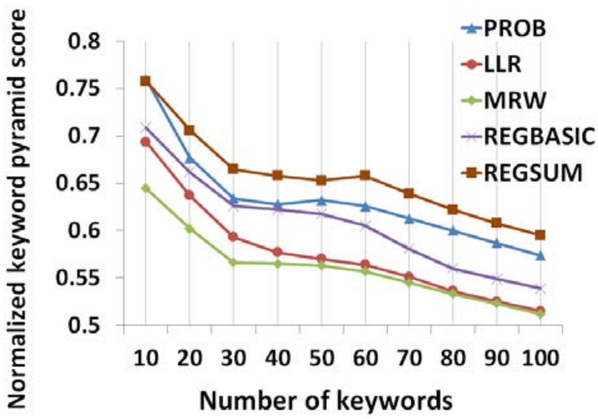

Figure 2: Normalized keyword pyramid scores

8.3 Regression for Summarization

We now show that the performance of extractive summarization can be improved by better estimation of word weights. We compare our regression system with the four models introduced in Section 8.1. We also include PEER-65, the best system in DUC-2004, as well as KLSUM for comparison. Apart from these, we compare our model with two state-of-the-art systems, including the submodular approach (SUBMOD) (Lin and Bilmes, 2012) and the determinantal point process (DPP) summarizer (Kulesza and Taskar, 2012). The summaries were kindly provided by the authors of these systems (Hong et al., 2014).

As can been seen in Table 8, our system outperforms PROB, LLR, MRW, PEER-65, KLSUM and REGBASIC. These improvements are significant on ROUGE-2 recall. Interestingly, although the supervised system REGBASIC which uses only frequency and positions achieve low performance in keyword identification, the summaries it generates are of high quality. The inclusion of position features negatively affects the performance in summary keyword identification but boosts the weights for the words which appear close to the beginning of the documents, which is helpful for identifying informative sentences. By including other features we greatly improve over REGBASIC in keyword identification. Similarly here the richer set of features results in better quality summaries.

We also examined the ROUGE-1, -2, -4 recall compared with the SUBMOD and DPP summarizers[9]. There is no significant difference on R-2 and R-4 recall compared with these two state-of-the-art systems. DPP performed significantly better than our system on R-1 recall, but that system is optimizing on R-1 F-score in training. Overall, our conceptually simple system is on par with the state of the art summarizers and points to the need for better models for estimating word importance.

System	R-1	R-2	R-4
PROB	35.14	8.17	1.06
LLR	34.60	7.56	0.83
MRW	35.78	8.15	0.99
REGBASIC	37.56	9.28	1.49
KL	37.97	8.53	1.26
PEER-65	37.62	8.96	1.51
SUBMOD	39.18	9.35	1.39
DPP	39.79	9.62	1.57
REGSUM	**38.57**	**9.75**	**1.60**

Table 8: System performance comparison (%)

9 Conclusion

We presented a series of experiments which show that keyword identification can be improved in a supervised framework which incorporates a rich set of indicators of importance. We also show that the better estimation of word importance leads to better extractive summaries. Our analysis of features related to global importance, sentiment and topical categories reveals rather unexpected results and confirms that word importance estimation is a worthy research direction. Success in the task is likely to improve sophisticated summarization approaches too, as well as sentence compression systems which use only crude frequency related measures to decide which words should be deleted from a sentence.[10]

[9]The results are slightly different from the ones reported in the original papers due to the fact that we truncated to 100 words, while they truncated to 665 bytes.

References

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of ACL-HLT*, pages 481–490.

Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A hybrid hierarchical model for multi-document summarization. In *Proceedings of ACL*, pages 815–824.

Asli Celikyilmaz and Dilek Hakkani-Tür. 2011. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of ACL-HLT*, pages 491–499.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of COLING/ACL*, pages 152–159.

Hal Daumé, III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of ACL*, pages 305–312.

Gunes Erkan and Dragomir R. Radev. 2004. Lexrank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. In *Proceedings of COLING*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL*, pages 363–370.

D. Graff, J. Kong, K. Chen, and K. Maeda. 2007. English gigaword third edition. *Linguistic Data Consortium, Philadelphia, PA*.

Surabhi Gupta, Ani Nenkova, and Dan Jurafsky. 2007. Measuring importance and query relevance in topic-focused multi-document summarization. In *Proceedings of ACL*, pages 193–196.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of HLT-NAACL*, pages 362–370.

[10]The work is partially funded by NSF CAREER award IIS 0953445.

Sanda Harabagiu and Finley Lacatusu. 2005. Topic themes for multi-document summarization. In *Proceedings of SIGIR 2005*, pages 202–209.

Kai Hong, John M. Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repositary of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of LREC*, May.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3).

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of SIGIR*, pages 68–73.

Page Lawrence, Brin Sergey, Rajeev Motwani, and Terry Winograd. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University.

Hui Lin and Jeff Bilmes. 2012. Learning mixtures of submodular shells with application to document summarization. In *UAI*, pages 479–490.

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedgins of COLING*, pages 495–501.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.

Marina Litvak, Mark Last, and Menahem Friedman. 2010. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of ACL*, pages 927–936.

Fei Liu, Feifan Liu, and Yang Liu. 2011. A supervised framework for keyword extraction from meeting transcripts. *Transactions on Audio Speech and Language Processing*, 19(3):538–548.

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

M. Marneffe, B. Maccartney, and C. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC-06*, pages 449–454.

Rebecca Mason and Eugene Charniak. 2011. Extractive multi-document summaries should explicitly not contain document-specific content. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pages 49–54.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of ECIR*, pages 557–564.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of EMNLP*, pages 404–411.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. Technical report, Microsoft Research.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of SIGIR*, pages 573–580.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The Pyramid Method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2), May.

Jahna Otterbacher, Günes Erkan, and Dragomir R. Radev. 2009. Biased lexrank: Passage retrieval using random walks with question-based priors. *Information Processing and Management*, 45(1):42–54.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Inf. Process. Manage.*, 43(6):1506–1520.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *NAACL-HLT 2012: Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9.

Peter Rankel, John Conroy, Eric Slud, and Dianne O'Leary. 2011. Ranking human and machine summarization systems. In *Proceedings of EMNLP*, pages 467–473.

Korbinian Riedhammer, Benoît Favre, and Dilek Hakkani-Tür. 2010. Long story short - global unsupervised models for keyphrase based meeting summarization. *Speech Communication*, 52(10):801–815.

G. Salton. 1971. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia, PA*.

Chao Shen and Tao Li. 2010. Multi-document summarization via the minimum dominating set. In *Proceedings of Coling*, pages 984–992.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Large-margin learning of submodular summarization models. In *Proceedings of EACL*, pages 224–233.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904.

Hiroya Takamura and Manabu Okumura. 2009. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of EACL*, pages 781–789.

Yla R Tausczik and James W Pennebaker. 2007. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29:24–54.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the NAACL-HLT*, pages 173–180.

Xiaojun Wan and Jianwu Yang. 2008. Multi-document summarization using cluster-based link analysis. In *Proceedings of SIGIR*, pages 299–306.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of ACL*, pages 552–559.

Furu Wei, Wenjie Li, Qin Lu, and Yanxiang He. 2008. Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In *Proceedings of SIGIR*, pages 283–290.

Janyce Wiebe and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. language resources and evaluation. In *Language Resources and Evaluation (formerly Computers and the Humanities)*, page 1(2).

Wen-tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. 2007. Multi-document summarization by maximizing informative content-words. In *Proceedings of IJCAI*, pages 1776–1782.

Hongyuan Zha. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of SIGIR*, pages 113–120.