

# A Sentence Compression Based Framework to Query-Focused Multi-Document Summarization

Lu Wang<sup>1</sup> Hema Raghavan<sup>2</sup> Vittorio Castelli<sup>2</sup> Radu Florian<sup>2</sup> Claire Cardie<sup>1</sup>

<sup>1</sup>Department of Computer Science, Cornell University, Ithaca, NY 14853, USA

{luwang, cardie}@cs.cornell.edu

<sup>2</sup>IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

{hraghav, vittorio, raduf}@us.ibm.com

## Abstract

We consider the problem of using sentence compression techniques to facilitate query-focused multi-document summarization. We present a sentence-compression-based framework for the task, and design a series of learning-based compression models built on parse trees. An innovative beam search decoder is proposed to efficiently find highly probable compressions. Under this framework, we show how to integrate various indicative metrics such as linguistic motivation and query relevance into the compression process by deriving a novel formulation of a compression scoring function. Our best model achieves statistically significant improvement over the state-of-the-art systems on several metrics (e.g. 8.0% and 5.4% improvements in ROUGE-2 respectively) for the DUC 2006 and 2007 summarization task.

## 1 Introduction

The explosion of the Internet clearly warrants the development of techniques for organizing and presenting information to users in an effective way. Query-focused multi-document summarization (MDS) methods have been proposed as one such technique and have attracted significant attention in recent years. The goal of query-focused MDS is to synthesize a brief (often fixed-length) and well-organized summary from a set of topic-related documents that answer a complex question or address a topic statement. The resulting summaries, in turn, can support a number of information analysis applications including open-ended question answering, recommender systems, and summarization of search engine results. As further evidence of its importance, the Document Understanding Conference (DUC) has used query-focused MDS as its main task since 2004 to foster

new research on automatic summarization in the context of users' needs.

To date, most top-performing systems for multi-document summarization—whether query-specific or not—remain largely *extractive*: their summaries are comprised exclusively of sentences selected directly from the documents to be summarized (Erkan and Radev, 2004; Haghighi and Vanderwende, 2009; Celikyilmaz and Hakkani-Tür, 2011). Despite their simplicity, extractive approaches have some disadvantages. First, lengthy sentences that are partly relevant are either excluded from the summary or (if selected) can block the selection of other important sentences, due to summary length constraints. In addition, when people write summaries, they tend to abstract the content and seldom use entire sentences taken verbatim from the original documents. In news articles, for example, most sentences are lengthy and contain both potentially useful information for a summary as well as unnecessary details that are better omitted. Consider the following DUC query as input for a MDS system:<sup>1</sup> “*In what ways have stolen artworks been recovered? How often are suspects arrested or prosecuted for the thefts?*” One manually generated summary includes the following sentence but removes the bracketed words in gray:

A man suspected of stealing a million-dollar collection of [hundreds of ancient] Nepalese and Tibetan art objects in New York [11 years ago] was arrested [Thursday at his South Los Angeles home, where he had been hiding the antiquities, police said].

In this example, the compressed sentence is rela-

<sup>1</sup>From DUC 2005, query for topic d422g.

tively more succinct and readable than the original (e.g. in terms of Flesch-Kincaid Reading Ease Score (Kincaid et al., 1975)). Likewise, removing information irrelevant to the query (e.g. “11 years ago”, “police said”) is crucial for query-focused MDS.

Sentence compression techniques (Knight and Marcu, 2000; Clarke and Lapata, 2008) are the standard for producing a compact and grammatical version of a sentence while preserving relevance, and prior research (e.g. Lin (2003)) has demonstrated their potential usefulness for generic document summarization. Similarly, strides have been made to incorporate sentence compression into query-focused MDS systems (Zajic et al., 2006). Most attempts, however, fail to produce better results than those of the best systems built on pure extraction-based approaches that use no sentence compression.

In this paper we investigate the role of sentence compression techniques for query-focused MDS. We extend existing work in the area first by investigating the role of *learning-based* sentence compression techniques. In addition, we design three types of approaches to sentence-compression—*rule-based*, *sequence-based* and *tree-based*—and examine them within our compression-based framework for query-specific MDS. Our top-performing sentence compression algorithm incorporates measures of query relevance, content importance, redundancy and language quality, among others. Our tree-based methods rely on a scoring function that allows for easy and flexible tailoring of sentence compression to the summarization task, ultimately resulting in significant improvements for MDS, while at the same time remaining competitive with existing methods in terms of sentence compression, as discussed next.

We evaluate the summarization models on the standard Document Understanding Conference (DUC) 2006 and 2007 corpora<sup>2</sup> for query-focused MDS and find that all of our compression-based summarization models achieve statistically significantly better performance than the best DUC 2006 systems. Our best-performing system yields an 11.02 ROUGE-2 score (Lin and Hovy, 2003), a 8.0% improvement over the best reported score (10.2 (Davis et al., 2012)) on the

<sup>2</sup>We believe that we can easily adapt our system for tasks (e.g. TAC-08’s opinion summarization or TAC-09’s update summarization) or domains (e.g. web pages or wikipedia pages). We reserve that for future work.

DUC 2006 dataset, and an 13.49 ROUGE-2, a 5.4% improvement over the best score in DUC 2007 (12.8 (Davis et al., 2012)). We also observe substantial improvements over previous systems w.r.t. the manual Pyramid (Nenkova and Passonneau, 2004) evaluation measure (26.4 vs. 22.9 (Jagarlamudi et al., 2006)); human annotators furthermore rate our system-generated summaries as having less redundancy and comparable quality w.r.t. other linguistic quality metrics. With these results we believe we are the first to successfully show that sentence compression can provide statistically significant improvements over pure extraction-based approaches for query-focused MDS.

## 2 Related Work

Existing research on query-focused multi-document summarization (MDS) largely relies on extractive approaches, where systems usually take as input a set of documents and select the top relevant sentences for inclusion in the final summary. A wide range of methods have been employed for this task. For unsupervised methods, sentence importance can be estimated by calculating topic signature words (Lin and Hovy, 2000; Conroy et al., 2006), combining query similarity and document centrality within a graph-based model (Otterbacher et al., 2005), or using a Bayesian model with sophisticated inference (Daumé and Marcu, 2006). Davis et al. (2012) first learn the term weights by Latent Semantic Analysis, and then greedily select sentences that cover the maximum combined weights. Supervised approaches have mainly focused on applying discriminative learning for ranking sentences (Fuentes et al., 2007). Lin and Bilmes (2011) use a class of carefully designed submodular functions to reward the diversity of the summaries and select sentences greedily.

Our work is more related to the less studied area of sentence compression as applied to (single) document summarization. Zajic et al. (2006) tackle the query-focused MDS problem using a compress-first strategy: they develop heuristics to generate multiple alternative compressions of all sentences in the original document; these then become the candidates for extraction. This approach, however, does not outperform some extraction-based approaches. A similar idea has been studied for MDS (Lin, 2003; Gillick and Favre, 2009),

but limited improvement is observed over extractive baselines with simple compression rules. Finally, although learning-based compression methods are promising (Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011), it is unclear how well they handle issues of redundancy.

Our research is also inspired by probabilistic sentence-compression approaches, such as the noisy-channel model (Knight and Marcu, 2000; Turner and Charniak, 2005), and its extension via synchronous context-free grammars (SCFG) (Aho and Ullman, 1969; Lewis and Stearns, 1968) for robust probability estimation (Galley and McKeown, 2007). Rather than attempt to derive a new parse tree like Knight and Marcu (2000) and Galley and McKeown (2007), we learn to safely remove a set of constituents in our parse tree-based compression model while preserving grammatical structure and essential content. Sentence-level compression has also been examined via a discriminative model McDonald (2006), and Clarke and Lapata (2008) also incorporate discourse information by using integer linear programming.

### 3 The Framework

We now present our query-focused MDS framework consisting of three steps: Sentence Ranking, Sentence Compression and Post-processing. First, sentence ranking determines the importance of each sentence given the query. Then, a sentence compressor iteratively generates the most likely succinct versions of the ranked sentences, which are cumulatively added to the summary, until a length limit is reached. Finally, the post-processing stage applies coreference resolution and sentence reordering to build the summary.

**Sentence Ranking.** This stage aims to rank sentences in order of relevance to the query. Unsurprisingly, ranking algorithms have been successfully applied to this task. We experimented with two of them – Support Vector Regression (SVR) (Mozer et al., 1997) and LambdaMART (Burges et al., 2007). The former has been used previously for MDS (Ouyang et al., 2011). LambdaMart on the other hand has shown considerable success in information retrieval tasks (Burges, 2010); we are the first to apply it to summarization. For training, we use 40 topics (i.e. queries) from the DUC 2005 corpus (Dang, 2005) along with their manually generated abstracts. As in previous work (Shen and Li,

Basic Features
relative/absolute position
is among the first 1/3/5 sentences?
number of words (with/without stopwords)
number of words more than 5/10 (with/without stopwords)
Query-Relevant Features
unigram/bigram/skip bigram (at most four words apart) overlap
unigram/bigram TF/TF-IDF similarity
mention overlap
subject/object/indirect object overlap
semantic role overlap
relation overlap
Query-Independent Features
average/total unigram/bigram IDF/TF-IDF
unigram/bigram TF/TF-IDF similarity with the centroid of the cluster
average/sum of sumBasic/SumFocus (Toutanova et al., 2007)
average/sum of mutual information
average/sum of number of topic signature words (Lin and Hovy, 2000)
basic/improved sentence scorers from Conroy et al. (2006)
Content Features
contains verb/web link/phone number?
contains/portion of words between parentheses

Table 1: Sentence-level features for sentence ranking.

2011; Ouyang et al., 2011), we use the ROUGE-2 score, which measures bigram overlap between a sentence and the abstracts, as the objective for regression.

While space limitations preclude a longer discussion of the full feature set (ref. Table 1), we describe next the query-relevant features used for sentence ranking as these are the most important for our summarization setting. The goal of this feature subset is to determine the similarity between the query and each candidate sentence. When computing similarity, we remove stopwords as well as the words “discuss, describe, specify, explain, identify, include, involve, note” that are adopted and extended from Conroy et al. (2006). Then we conduct simple query expansion based on the title of the topic and cross-document coreference resolution. Specifically, we first add the words from the topic title to the query. And for each mention in the query, we add other mentions within the set of documents that corefer with this mention. Finally, we compute two versions of the features—one based on the original query and another on the expanded one. We also derive the semantic role overlap and relation instance overlap between the query and each sentence. Cross-document coreference resolution, semantic role labeling and relation extraction are accomplished via the methods described in Section 5.

**Sentence Compression.** As the main focus of this paper, we propose three types of compression methods, described in detail in Section 4 below.

**Post-processing.** Post-processing performs *coreference resolution* and *sentence ordering*.

Basic Features	Syntactic Tree Features
first 1/3/5 tokens (toks)? last 1/3/5 toks? first letter/all letters capitalized? is negation? is stopword? <b>Dependency Tree Features</b> dependency relation (dep rel) parent/grandparent dep rel is the root? has a depth larger than 3/5?	POS tag parent/grandparent label leftmost child of parent? second leftmost child of parent? is headword? in NP/VP/ADVP/ADJP chunk? <b>Semantic Features</b> is a predicate? semantic role label
Rule-Based Features	
For each rule in Table 2, we construct a corresponding feature to indicate whether the token is identified by the rule.	

Table 3: Token-level features for sequence-based compression.

We replace each pronoun with its referent unless they appear in the same sentence. For sentence ordering, each compressed sentence is assigned to the most similar (tf-idf) query sentence. Then a Chronological Ordering algorithm (Barzilay et al., 2002) sorts the sentences for each query based first on the time stamp, and then the position in the source document.

## 4 Sentence Compression

Sentence compression is typically formulated as the problem of removing secondary information from a sentence while maintaining its grammaticality and semantic structure (Knight and Marcu, 2000; McDonald, 2006; Galley and McKeown, 2007; Clarke and Lapata, 2008). We leave other rewrite operations, such as paraphrasing and re-ordering, for future work. Below we describe the sentence compression approaches developed in this research: RULE-BASED COMPRESSION, SEQUENCE-BASED COMPRESSION, and TREE-BASED COMPRESSION.

### 4.1 Rule-based Compression

Turner and Charniak (2005) have shown that applying hand-crafted rules for trimming sentences can improve both content and linguistic quality. Our rule-based approach extends existing work (Conroy et al., 2006; Toutanova et al., 2007) to create the linguistically-motivated compression rules of Table 2. To avoid ill-formed output, we disallow compressions of more than 10 words by each rule.

### 4.2 Sequence-based Compression

As in McDonald (2006) and Clarke and Lapata (2008), our sequence-based compression model makes a binary “keep-or-delete” decision for each word in the sentence. In contrast, however, we

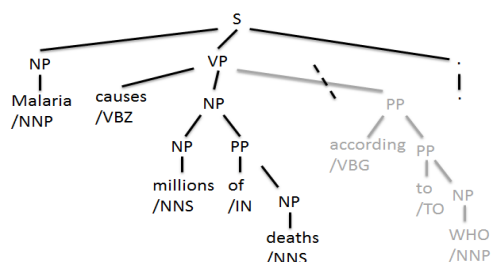


Figure 1: Diagram of tree-based compression. The nodes to be dropped are grayed out. In this example, the root of the gray subtree (a “PP”) would be labeled REMOVE. Its siblings and parent are labeled RETAIN and PARTIAL, respectively. The trimmed tree is realized as “Malaria causes millions of deaths.”

view compression as a sequential tagging problem and make use of linear-chain Conditional Random Fields (CRFs) (Lafferty et al., 2001) to select the most likely compression. We represent each sentence as a sequence of tokens,  $X = x_0x_1 \dots x_n$ , and generate a sequence of labels,  $Y = y_0y_1 \dots y_n$ , that encode which tokens are kept, using a BIO label format: {B-RETAIN denotes the beginning of a retained sequence, I-RETAIN indicates tokens “inside” the retained sequence, O marks tokens to be removed}.

The CRF model is built using the features shown in Table 3. “Dependency Tree Features” encode the grammatical relations in which each word is involved as a dependent. For the “Syntactic Tree”, “Dependency Tree” and “Rule-Based” features, we also include features for the two words that precede and the two that follow the current word. Detailed descriptions of the training data and experimental setup are in Section 5.

During inference, we find the maximally likely sequence  $Y$  according to a CRF with parameter  $\theta$  ( $Y = \arg \max_{Y'} P(Y'|X; \theta)$ ), while simultaneously enforcing the rules of Table 2 to reduce the hypothesis space and encourage grammatical compression. To do this, we encode these rules as features for each token, and whenever these feature functions fire, we restrict the possible label for that token to “O”.

### 4.3 Tree-based Compression

Our tree-based compression methods are in line with syntax-driven approaches (Galley and McKeown, 2007), where operations are carried out on parse tree constituents. Unlike previous work (Knight and Marcu, 2000; Galley and McKeown, 2007), we do not produce a new parse tree,

Rule	Example
Header	[MOSCOW , October 19 ( Xinhua ) –] Russian federal troops Tuesday continued...
Relative dates	...Centers for Disease Control confirmed [Tuesday] that there was...
Intra-sentential attribution	...fueling the La Nina weather phenomenon, [the U.N. weather agency said].
Lead adverbials	[Interestingly], while the Democrats tend to talk about...
Noun appositives	Wayne County Prosecutor [John O’Hara] wanted to send a message...
Nonrestrictive relative clause	Putin, [who was born on October 7, 1952 in Leningrad], was elected in the presidential election...
Adverbial clausal modifiers (Lead sentence)	[Starting in 1998], California will require 2 per cent of a manufacturer... [Given the short time], car makers see electric vehicles as...
Within Parentheses	...to Christian home schoolers in the early 1990s [(www.homecomputermarket.com)].

Table 2: Linguistically-motivated rules for sentence compression. The grayed-out words in brackets are removed.

but focus on learning to identify the proper set of constituents to be removed. In particular, when a node is dropped from the tree, all words it subsumes will be deleted from the sentence.

Formally, given a parse tree  $T$  of the sentence to be compressed and a tree traversal algorithm,  $T$  can be presented as a list of ordered constituent nodes,  $T = t_0 t_1 \dots t_m$ . Our objective is to find a set of labels,  $L = l_0 l_1 \dots l_m$ , where  $l_i \in \{\text{RETAIN, REMOVE, PARTIAL}\}$ . RETAIN (RET) and REMOVE (REM) denote whether the node  $t_i$  is retained or removed. PARTIAL (PAR) means  $t_i$  is partly removed, i.e. at least one child subtree of  $t_i$  is dropped.

Labels are identified, in order, according to the tree traversal algorithm. Every node label needs to be *compatible* with the labeling history: given a node  $t_i$ , and a set of labels  $l_0 \dots l_{i-1}$  predicted for nodes  $t_0 \dots t_{i-1}$ ,  $l_i = \text{RET}$  or  $l_i = \text{REM}$  is *compatible* with the history when all children of  $t_i$  are labeled as RET or REM, respectively;  $l_i = \text{PAR}$  is *compatible* when  $t_i$  has at least two descendents  $t_j$  and  $t_k$  ( $j < i$  and  $k < i$ ), one of which is RETained and the other, REMoved. As such, the root of the gray subtree in Figure 1 is labeled as REM; its left siblings as RET; its parent as PAR.

As the space of possible compressions is exponential in the number of leaves in the parse tree, instead of looking for the globally optimal solution, we use beam search to find a set of highly likely compressions and employ a language model trained on a large corpus for evaluation.

**A Beam Search Decoder.** The beam search decoder (see Algorithm 1) takes as input the sentence’s parse tree  $T = t_0 t_1 \dots t_m$ , an ordering  $O$  for traversing  $T$  (e.g. postorder) as a sequence of nodes in  $T$ , the set  $L$  of possible node labels, a scoring function  $S$  for evaluating each sentence compression hypothesis, and a beam size  $N$ . Specifically,  $O$  is a permutation on the set  $\{0, 1, \dots, m\}$ —each element an

index onto  $T$ . Following  $O$ ,  $T$  is re-ordered as  $t_{O_0} t_{O_1} \dots t_{O_m}$ , and the decoder considers each ordered constituent  $t_{O_i}$  in turn. In iteration  $i$ , all existing sentence compression hypotheses are expanded by one node,  $t_{O_i}$ , labeling it with **all compatible** labels. The new hypotheses (usually sub-sentences) are ranked by the scorer  $S$  and the top  $N$  are preserved to be extended in the next iteration. See Figure 2 for an example.

```

Input : parse tree  $T$ , ordering  $O = O_0 O_1 \dots O_m$ ,
          $L = \{\text{RET, REM, PAR}\}$ , hypothesis scorer  $S$ ,
         beam size  $N$ 
Output:  $N$  best compressions

stack  $\leftarrow \Phi$  (empty set);
foreach node  $t_{O_i}$  in  $T = t_{O_0} \dots t_{O_m}$  do
  if  $i == 0$  (first node visited) then
    foreach label  $l_{O_0}$  in  $L$  do
      newHypothesis  $h' \leftarrow [l_{O_0}]$ ;
      put  $h'$  into Stack;
    end
  else
    newStack  $\leftarrow \Phi$  (empty set);
    foreach hypothesis  $h$  in stack do
      foreach label  $l_{O_i}$  in  $L$  do
        if  $l_{O_i}$  is compatible then
          newHypothesis  $h' \leftarrow h + [l_{O_i}]$ ;
          put  $h'$  into newStack;
        end
      end
    end
    stack  $\leftarrow$  newStack;
  end
Apply  $S$  to sort hypotheses in stack in descending order;
Keep the  $N$  best hypotheses in stack;
end

```

Algorithm 1: Beam search decoder.

Our **BASIC Tree-based Compression** instantiates the beam search decoder with postorder traversal and a hypothesis scorer that takes a possible sentence compression—a sequence of nodes (e.g.  $t_{O_0} \dots t_{O_k}$ ) and their labels (e.g.  $l_{O_0} \dots l_{O_k}$ )—and returns  $\sum_{j=1}^k \log P(l_{O_j} | t_{O_j})$  (denoted later as  $Score_{Basic}$ ). The probability is estimated by a Maximum Entropy classifier (Berger et al.,

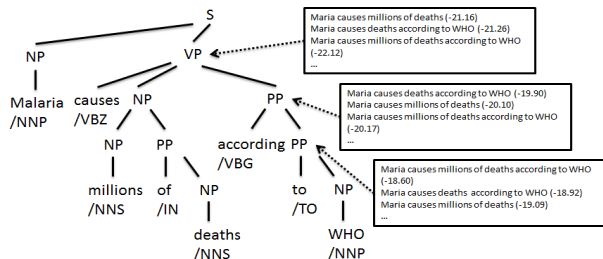


Figure 2: Example of beam search decoding. For postorder traversal, the three nodes are visited in a bottom-up order. The associated compression hypotheses (boxed) are ranked based on the scores in parentheses. Beam scores for other nodes are omitted.

Basic Features	Syntactic Tree Features
projection falls w/in first 1/3/5 toks?*	constituent label
projection falls w/in last 1/3/5 toks?*	parent left/right sibling label
subsumes first 1/3/5 toks?*	grandparent left/right sibling label
subsumes last 1/3/5 toks?*	is leftmost child of parent?
number of words larger than 5/10?*	is second leftmost child of parent?
is leaf node?*	is head node of parent?
is root of parsing tree?*	label of its head node
has word with first letter capitalized?	has a depth greater than 3/5/10?
has word with all letters capitalized?	<b>Dependency Tree Features</b>
has negation?	dep rel of head node†
has stopwords?	dep rel of parent's head node†
<b>Semantic Features</b>	dep rel of grandparent's head node†
the head node has predicate?	contain root of dep tree?†
semantic roles of head node	has a depth larger than 3/5?†
<b>Rule-Based Features</b>	
For each rule in Table 2, we construct a corresponding feature to indicate whether the token is identified by the rule.	

Table 4: Constituent-level features for tree-based compression. \* or † denote features that are concatenated with every Syntactic Tree feature to compose a new one.

1996) trained at the constituent level using the features in Table 4. We also apply the rules of Table 2 during the decoding process. Concretely, if the words subsumed by a node are identified by any rule, we only consider REM as the node's label.

Given the  $N$ -best compressions from the decoder, we evaluate the yield of the trimmed trees using a language model trained on the Gigaword (Graff, 2003) corpus and return the compression with the highest probability. Thus, the decoder is quite flexible — its learned scoring function allows us to incorporate features salient for sentence compression while its language model guarantees the linguistic quality of the compressed string. In the sections below we consider additional improvements.

### 4.3.1 Improving Beam Search

**CONTEXT-aware search** is based on the intuition that predictions on preceding context can be leveraged to facilitate the prediction of the current node. For example, parent nodes with

children that have all been removed (retained) should have a label of REM (RET). In light of this, we encode these contextual predictions as additional features of  $S$ , that is, ALL-CHILDREN-REMOVED/RETAINED, ANY-LEFTSIBLING-REMOVED/RETAINED/PARTLY\_REMOVED, LABEL-OF-LEFT-SIBLING/HEAD-NODE.

**HEAD-driven search** modifies the BASIC postorder tree traversal by visiting the head node first at each level, leaving other orders unchanged. In a nutshell, if the head node is dropped, then its modifiers need not be preserved. We adopt the same features as CONTEXT-aware search, but remove those involving left siblings. We also add one more feature: LABEL-OF-THE-HEAD-NODE-IT-MODIFIES.

### 4.3.2 Task-Specific Sentence Compression

The current scorer  $Score_{Basic}$  is still fairly naive in that it focuses only on features of the sentence to be compressed. *However extra-sentential knowledge can also be important for query-focused MDS.* For example, information regarding relevance to the query might lead the decoder to produce compressions better suited for the summary. Towards this goal, we construct a compression scoring function—the *multi-scorer* (MULTI)—that allows the incorporation of multiple task-specific scorers. Given a hypothesis at any stage of decoding, which yields a sequence of words  $W = w_0w_1\dots w_j$ , we propose the following component scorers.

**Query Relevance.** Query information ought to guide the compressor to identify the relevant content. The query  $Q$  is expanded as described in Section 3. Let  $|W \cap Q|$  denote the number of unique overlapping words between  $W$  and  $Q$ , then  $score_q = |W \cap Q|/|W|$ .

**Importance.** A query-independent importance score is defined as the average Sum-Basic (Toutanova et al., 2007) value in  $W$ , i.e.  $score_{im} = \sum_{i=1}^j SumBasic(w_i)/|W|$ .

**Language Model.** We let  $score_{lm}$  be the probability of  $W$  computed by a language model.

**Cross-Sentence Redundancy.** To encourage diversified content, we define a redundancy score to discount replicated content:  $score_{red} = 1 - |W \cap C|/|W|$ , where  $C$  is the words already selected for the summary.

The *multi-scorer* is defined as a linear combination of the component scorers: Let  $\vec{\alpha} = (\alpha_0, \dots, \alpha_4)$ ,  $0 \leq \alpha_i \leq 1$ ,  $\overrightarrow{score} = (score_{Basic}, score_q, score_{im}, score_{lm}, score_{red})$ ,

$$S = score_{multi} = \vec{\alpha} \cdot \overrightarrow{score} \quad (1)$$

The parameters  $\vec{\alpha}$  are tuned on a held-out tuning set by grid search. We linearly normalize the score of each metric, where the minimum and maximum values are estimated from the tuning data.

## 5 Experimental Setup

We evaluate our methods on the DUC 2005, 2006 and 2007 datasets (Dang, 2005; Dang, 2006; Dang, 2007), each of which is a collection of newswire articles. 50 complex queries (topics) are provided for DUC 2005 and 2006, 35 are collected for DUC 2007 main task. Relevant documents for each query are provided along with 4 to 9 human MDS abstracts. The task is to generate a summary within 250 words to address the query. We split DUC 2005 into two parts: 40 topics to train the sentence ranking models, and 10 for ranking algorithm selection and parameter tuning for the multi-scorer. DUC 2006 and DUC 2007 are reserved as held out test sets.

*Sentence Compression.* The dataset from Clarke and Lapata (2008) is used to train the CRF and MaxEnt classifiers (Section 4). It includes 82 newswire articles with one manually produced compression aligned to each sentence.

*Preprocessing.* Documents are processed by a full NLP pipeline, including token and sentence segmentation, parsing, semantic role labeling, and an information extraction pipeline consisting of mention detection, NP coreference, cross-document resolution, and relation detection (Florin et al., 2004; Luo et al., 2004; Luo and Zitouni, 2005).

*Learning for Sentence Ranking and Compression.* We use Weka (Hall et al., 2009) to train a support vector regressor and experiment with various rankers in RankLib (Dang, 2011)<sup>3</sup>. As LambdaMART has an edge over other rankers on the held-out dataset, we selected it to produce ranked sentences for further processing. For sequence-based compression using CRFs, we employ Mallet (McCallum, 2002) and integrate the Table 2 rules during inference. NLTK (Bird et al., 2009)

<sup>3</sup>Default parameters are used. If an algorithm needs a validation set, we use 10 out of 40 topics.

MaxEnt classifiers are used for tree-based compression. Beam size is fixed at 2000.<sup>4</sup> Sentence compressions are evaluated by a 5-gram language model trained on Gigaword (Graff, 2003) by SRILM (Stolcke, 2002).

## 6 Results

The results in Table 5 use the official ROUGE software with standard options<sup>5</sup> and report ROUGE-2 (R-2) (measures bigram overlap) and ROUGE-SU4 (R-SU4) (measures unigram and skip-bigram separated by up to four words). We compare our sentence-compression-based methods to the best performing systems based on ROUGE in DUC 2006 and 2007 (Jagarlamudi et al., 2006; Pingali et al., 2007), system by Davis et al. (2012) that report the best R-2 score on DUC 2006 and 2007 thus far, and to the purely extractive methods of SVR and LambdaMART.

Our sentence-compression-based systems (marked with †) show statistically significant improvements over pure extractive summarization for both R-2 and R-SU4 (paired *t*-test,  $p < 0.01$ ). This means our systems can effectively remove redundancy within the summary through compression. Furthermore, our HEAD-driven beam search method with MULTI-scorer beats all systems on DUC 2006<sup>6</sup> and all systems on DUC 2007 except the best system in terms of R-2 ( $p < 0.01$ ). Its R-SU4 score is also significantly ( $p < 0.01$ ) better than extractive methods, rule-based and sequence-based compression methods on both DUC 2006 and 2007. Moreover, our systems with learning-based compression have considerable compression rates, indicating their capability to remove superfluous words as well as improve summary quality.

**Human Evaluation.** The Pyramid (Nenkova and Passonneau, 2004) evaluation was developed to manually assess how many relevant facts or Summarization Content Units (SCUs) are captured by system summaries. We ask a professional annotator (who is not one of the authors, is highly experienced in annotating for various NLP tasks, and is fluent in English) to carry out a Pyramid evaluation on 10 randomly selected topics from

<sup>4</sup>We looked at various beam sizes on the heldout data, and observed that the performance peaks around this value.

<sup>5</sup>ROUGE-1.5.5.pl -n 4 -w 1.2 -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -a -d

<sup>6</sup>The system output from Davis et al. (2012) is not available, so significance tests are not conducted on it.

System	DUC 2006			DUC 2007		
	C Rate	R-2	R-SU4	C Rate	R-2	R-SU4
Best DUC system	–	9.56	15.53	–	12.62	17.90
Davis et al. (2012)	–	10.2	15.2	–	12.8	17.5
SVR	100%	7.78	13.02	100%	9.53	14.69
LambdaMART	100%	9.84	14.63	100%	12.34	15.62
Rule-based	78.99%	10.62 *†	15.73 †	78.11%	13.18†	18.15†
Sequence	76.34%	10.49 †	15.60 †	77.20%	13.25†	18.23†
Tree (BASIC + <i>Score<sub>Basic</sub></i> )	70.48%	10.49 †	15.86 †	69.27%	13.00†	18.29†
Tree (CONTEXT + <i>Score<sub>Basic</sub></i> )	65.21%	10.55 *†	16.10 †	63.44%	12.75	18.07†
Tree (HEAD + <i>Score<sub>Basic</sub></i> )	66.70%	10.66 *†	16.18 †	65.05%	12.93	18.15†
Tree (HEAD + MULTI)	70.20%	<b>11.02</b> *†	<b>16.25</b> †	73.40%	<b>13.49</b> †	<b>18.46</b> †

Table 5: Query-focused MDS performance comparison: C Rate or *compression rate* is the proportion of words preserved. R-2 (ROUGE-2) and R-SU4 (ROUGE-SU4) scores are multiplied by 100. “–” indicates that data is unavailable. BASIC, CONTEXT and HEAD represent the basic beam search decoder, context-aware and head-driven search extensions respectively. *Score<sub>Basic</sub>* and MULTI refer to the type of scorer used. Statistically significant improvements ( $p < 0.01$ ) over the best system in DUC 06 and 07 are marked with \*. † indicates statistical significance ( $p < 0.01$ ) over extractive approaches (SVR or LambdaMART). HEAD + MULTI outperforms all the other extract- and compression-based systems in R-2.

System	Pyr	Gra	Non-Red	Ref	Foc	Coh
Best DUC system (ROUGE)	22.9±8.2	3.5±0.9	3.5±1.0	3.5±1.1	3.6±1.0	2.9±1.1
Best DUC system (LQ)	–	4.0±0.8	4.2±0.7	3.8±0.7	3.6±0.9	3.4±0.9
Our System	<b>26.4±10.3</b>	3.0±0.9	4.0±1.1	3.6±1.0	3.4±0.9	2.8±1.0

Table 6: Human evaluation on our multi-scorer based system, Jagarlamudi et al. (2006) (Best DUC system (ROUGE)), and Lacatusu et al. (2006) (Best DUC system (LQ)). Our system can synthesize more relevant content according to Pyramid ( $\times 100$ ). We also examine linguistic quality (LQ) in Grammaticality (Gra), Non-redundancy (Non-Red), Referential clarity (Ref), Focus (Foc), and Structure and Coherence (Coh) like Dang (2006), each rated from 1 (very poor) to 5 (very good). Our system has better non-redundancy than Jagarlamudi et al. (2006) and is comparable to Jagarlamudi et al. (2006) and Lacatusu et al. (2006) in other metrics except grammaticality.

the DUC 2006 task with gold-standard SCU annotation in abstracts. The Pyramid score (see Table 6) is re-calculated for the system with best ROUGE scores in DUC 2006 (Jagarlamudi et al., 2006) along with our system by the same annotator to make a meaningful comparison.

We further evaluate the linguistic quality (LQ) of the summaries for the same 10 topics in accordance with the measurement in Dang (2006). Four native speakers who are undergraduate students in computer science (none are authors) performed the task. We compare our system based on HEAD-driven beam search with MULTI-scorer to the best systems in DUC 2006 achieving top ROUGE scores (Jagarlamudi et al., 2006) (Best DUC system (ROUGE)) and top linguistic quality scores (Lacatusu et al., 2006) (Best DUC system (LQ))<sup>7</sup>. The average score and standard deviation for each metric is displayed in Table 6. Our system achieves a higher Pyramid score, an indication that it captures more of the salient facts. We also

attain better non-redundancy than Jagarlamudi et al. (2006), meaning that human raters perceive less replicative content in our summaries. Scores for other metrics are comparable to Jagarlamudi et al. (2006) and Lacatusu et al. (2006), which either uses minimal non-learning-based compression rules or is a pure extractive system. However, our compression system sometimes generates less grammatical sentences, and those are mostly due to parsing errors. For example, parsing a clause starting with a past tense verb as an adverbial clausal modifier can lead to an ill-formed compression. Those issues can be addressed by analyzing  $k$ -best parse trees and we leave it in the future work. A sample summary from our multi-scorer based system is in Figure 3.

**Sentence Compression Evaluation.** We also evaluate sentence compression separately on (Clarke and Lapata, 2008), adopting the same partitions as (Martins and Smith, 2009), i.e. 1, 188 sentences for training and 441 for testing. Our compression models are compared with Hedge Trimmer (Dorr et al., 2003), a discriminative model proposed by McDonald (2006) and a

<sup>7</sup>Lacatusu et al. (2006) obtain the best scores in three linguistic quality metrics (i.e. grammaticality, focus, structure and coherence), and overall responsiveness on DUC 2006.



System	C Rate	Uni-Prec	Uni-Rec	Uni-F1	Rel-F1
HedgeTrimmer	57.64%	0.72	0.65	0.64	0.50
McDonald (2006)	70.95%	0.77	0.78	<i>0.77</i>	0.55
Martins and Smith (2009)	71.35%	0.77	0.78	<i>0.77</i>	0.56
Rule-based	87.65%	0.74	0.91	0.80	0.63
Sequence	70.79%	0.77	0.80	<i>0.76</i>	0.58
Tree (BASIC)	69.65%	0.77	0.79	0.75	0.56
Tree (CONTEXT)	67.01%	<b>0.79</b>	0.78	<i>0.76</i>	0.57
Tree (HEAD)	68.06%	<b>0.79</b>	0.80	<i>0.77</i>	<b>0.59</b>

Table 7: Sentence compression comparison. The true c rate is 69.06% for the test set. Tree-based approaches all use single-scorer. Our context-aware and head-driven tree-based approaches outperform all the other systems significantly ( $p < 0.01$ ) in precision (**Uni-Prec**) without sacrificing the recalls (i.e. there is no statistically significant difference between our models and McDonald (2006) / M & S (2009) with  $p > 0.05$ ). *Italicized* numbers for unigram F1 (**Uni-F1**) are statistically indistinguishable ( $p > 0.05$ ). Our head-driven tree-based approach also produces significantly better grammatical relations F1 scores (**Rel-F1**) than all the other systems except the rule-based method ( $p < 0.01$ ).

Topic <i>D0626H</i> : How were the bombings of the US embassies in Kenya and Tanzania conducted? What terrorist groups and individuals were responsible? How and where were the attacks planned?
WASHINGTON, August 13 (Xinhua) – President Bill Clinton Thursday condemned terrorist bomb attacks at U.S. embassies in Kenya and Tanzania and vowed to find the bombers and bring them to justice. Clinton met with his top aides Wednesday in the White House to assess the situation following the twin bombings at U.S. embassies in Kenya and Tanzania, which have killed more than 250 people and injured over 5,000, most of them Kenyans and Tanzanians. Local sources said the plan to bomb U.S. embassies in Kenya and Tanzania took three months to complete and bombers destined for Kenya were dispatched through Somali and Rwanda. FBI Director Louis Freeh, Attorney General Janet Reno and other senior U.S. government officials will hold a news conference at 1 p.m. EDT (1700GMT) at FBI headquarters in Washington “to announce developments in the investigation of the bombings of the U.S. embassies in Kenya and Tanzania,” the FBI said in a statement. ...

Figure 3: Part of the summary generated by the multi-scorer based summarizer for topic *D0626H* (DUC 2006). Grayed out words are removed. Query-irrelevant phrases, such as temporal information or source of the news, have been removed.

dependency-tree based compressor (Martins and Smith, 2009)<sup>8</sup>. We adopt the metrics in Martins and Smith (2009) to measure the unigram-level macro precision, recall, and F1-measure with respect to human annotated compression. In addition, we also compute the F1 scores of grammatical relations which are annotated by RASP (Briscoe and Carroll, 2002) according to Clarke and Lapata (2008).

In Table 7, our context-aware and head-driven tree-based compression systems show statistically significantly ( $p < 0.01$ ) higher precisions (**Uni-**

**Prec**) than all the other systems, without decreasing the recalls (**Uni-Rec**) significantly ( $p > 0.05$ ) based on a paired *t*-test. Unigram F1 scores (**Uni-F1**) in italics indicate that the corresponding systems are not statistically distinguishable ( $p > 0.05$ ). For grammatical relation evaluation, our head-driven tree-based system obtains statistically significantly ( $p < 0.01$ ) better F1 score (**Rel-F1**) than all the other systems except the rule-based system).

## 7 Conclusion

We have presented a framework for query-focused multi-document summarization based on sentence compression. We propose three types of compression approaches. Our tree-based compression method can easily incorporate measures of query relevance, content importance, redundancy and language quality into the compression process. By testing on a standard dataset using the automatic metric ROUGE, our models show substantial improvement over pure extraction-based methods and state-of-the-art systems. Our best system also yields better results for human evaluation based on Pyramid and achieves comparable linguistic quality scores.

## Acknowledgments

This work was supported in part by National Science Foundation Grant IIS-0968450 and a gift from Boeing. We thank Ding-Jung Han, Young-Suk Lee, Xiaoqiang Luo, Sameer Maskey, Myle Ott, Salim Roukos, Yiye Ruan, Ming Tan, Todd Ward, Bowen Zhou, and the ACL reviewers for valuable suggestions and advice on various aspects of this work.

<sup>8</sup>Thanks to André F.T. Martins for system outputs.

## References

- Alfred V. Aho and Jeffrey D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *J. Comput. Syst. Sci.*, 3(1):37–56.
- Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Int. Res.*, 17(1):35–55, August.
- Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. ACL '11, pages 481–490, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, March.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.
- T. Briscoe and J. Carroll. 2002. Robust accurate statistical annotation of general text.
- Christopher J.C. Burges, Robert Ragno, and Quoc Viet Le. 2007. Learning to rank with nonsmooth cost functions. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 193–200. MIT Press, Cambridge, MA.
- Christopher J. C. Burges. 2010. From RankNet to LambdaRank to LambdaMART: An overview. Technical report, Microsoft Research.
- Asli Celikyilmaz and Dilek Hakkani-Tür. 2011. Discovery of topically coherent sentences for extractive summarization. ACL '11, pages 491–499, Stroudsburg, PA, USA. Association for Computational Linguistics.
- James Clarke and Mirella Lapata. 2008. Global inference for sentence compression an integer linear programming approach. *J. Artif. Int. Res.*, 31(1):399–429, March.
- John M. Conroy, Judith D. Schlesinger, Dianne P. O'Leary, and Jade Goldstein, 2006. *Back to Basics: CLASSY 2006*. U.S. National Inst. of Standards and Technology.
- Hoa T. Dang. 2005. Overview of DUC 2005. In *Document Understanding Conference*.
- Hoa Tran Dang. 2006. Overview of DUC 2006. In *Proc. Document Understanding Workshop*, page 10 pages. NIST.
- Hoa T. Dang. 2007. Overview of DUC 2007. In *Document Understanding Conference*.
- Van Dang. 2011. RankLib. Online.
- Hal Daumé, III and Daniel Marcu. 2006. Bayesian query-focused summarization. ACL '06, pages 305–312, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. 2012. Occams - an optimal combinatorial covering algorithm for multi-document summarization. In *ICDM Workshops*, pages 454–463.
- Bonnie J Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: a parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop - Volume 5*, HLT-NAACL-DUC '03, pages 1 – 8, Stroudsburg, PA, USA. Association for Computational Linguistics, Association for Computational Linguistics.
- Günes Erkan and Dragomir R. Radev. 2004. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December.
- Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, Nicolas Nicolov, and Salim Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *HLT-NAACL*, pages 1–8.
- Maria Fuentes, Enrique Alfonseca, and Horacio Rodríguez. 2007. Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 57–60, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Michel Galley and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. NAACL '07, pages 180–187, Rochester, New York, April. Association for Computational Linguistics.
- Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, ILP '09, pages 10–18, Stroudsburg, PA, USA. Association for Computational Linguistics.
- David Graff. 2003. English Gigaword.
- Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. NAACL '09, pages 362–370, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explorer Newsl.*, 11(1):10–18, November.
- Jagadeesh Jagarlamudi, Prasad Pingali, and Vasudeva Varma, 2006. *Query Independent Sentence Scoring approach to DUC 2006*.
- J. Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel. Technical report, February.
- Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. AAI '00, pages 703–710. AAI Press.
- Finley Lacatusu, Andrew Hickl, Kirk Roberts, Ying Shi, Jeremy Bensley, Bryan Rink, Patrick Wang, and Lara Taylor, 2006. *LCC's gistexter at duc 2006: Multi-strategy multi-document summarization*.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In

- Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- P. M. Lewis, II and R. E. Stearns. 1968. Syntax-directed transduction. *J. ACM*, 15(3):465–488, July.
- Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11*, pages 510–520, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics - Volume 1, COLING '00*, pages 495–501, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 71–78.
- Chin-Yew Lin. 2003. Improving summarization performance by sentence compression: a pilot study. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages - Volume 11, AsianIR '03*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Xiaoqiang Luo and Imed Zitouni. 2005. Multi-lingual coreference resolution with syntactic features. In *HLT/EMNLP*.
- Xiaoqiang Luo, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *ACL*, pages 135–142.
- André F. T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing, ILP '09*, pages 1–9, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Ryan McDonald. 2006. Discriminative Sentence Compression with Soft Syntactic Constraints. In *Proceedings of the 11th EACL*, Trento, Italy, April.
- Michael Mozer, Michael I. Jordan, and Thomas Petsche, editors. 1997. *Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA, December 2-5, 1996*. MIT Press.
- Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 145–152, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.
- Jahna Otterbacher, Güneş Erkan, and Dragomir R. Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05*, pages 915–922, Stroudsburg, PA, USA. Association for Computational Linguistics.
- You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. 2011. Applying regression models to query-focused multi-document summarization. *Inf. Process. Manage.*, 47(2):227–237, March.
- Prasad Pingali, Rahul K, and Vasudeva Varma, 2007. *IIIT Hyderabad at DUC 2007*. U.S. National Inst. of Standards and Technology.
- Chao Shen and Tao Li. 2011. Learning to rank for query-focused multi-document summarization. In Diane J. Cook, Jian Pei, Wei Wang 0010, Osmar R. Zaane, and Xindong Wu, editors, *ICDM*, pages 626–634. IEEE.
- Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904, Denver, USA.
- Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. 2007. The PYTHY Summarization System: Microsoft Research at DUC 2007. In *Proc. of DUC*.
- Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. *ACL '05*, pages 290–297, Stroudsburg, PA, USA. Association for Computational Linguistics.
- David Zajic, Bonnie J Dorr, Jimmy Lin, and R. Schwartz. 2006. Sentence compression as a component of a multi-document summarization system. *Proceedings of the 2006 Document Understanding Workshop, New York*.