Automatic Generation of Story Highlights

Automatic Generation of Story Highlights

Kristian Woodsend and **Mirella Lapata**
School of Informatics, University of Edinburgh
Edinburgh EH8 9AB, United Kingdom
k.woodsend@ed.ac.uk, mlap@inf.ed.ac.uk

Abstract

In this paper we present a joint content selection and compression model for single-document summarization. The model operates over a phrase-based representation of the source document which we obtain by merging information from PCFG parse trees and dependency graphs. Using an integer linear programming formulation, the model learns to select and combine phrases subject to length, coverage and grammar constraints. We evaluate the approach on the task of generating "story highlights"—a small number of brief, self-contained sentences that allow readers to quickly gather information on news stories. Experimental results show that the model's output is comparable to human-written highlights in terms of both grammaticality and content.

1 Introduction

Summarization is the process of condensing a source text into a shorter version while preserving its information content. Humans summarize on a daily basis and effortlessly, but producing high quality summaries automatically remains a challenge. The difficulty lies primarily in the nature of the task which is complex, must satisfy many constraints (e.g., summary length, informativeness, coherence, grammaticality) and ultimately requires wide-coverage text understanding. Since the latter is beyond the capabilities of current NLP technology, most work today focuses on extractive summarization, where a summary is created simply by identifying and subsequently concatenating the most important sentences in a document.

Without a great deal of linguistic analysis, it is possible to create summaries for a wide range of documents. Unfortunately, extracts are often documents of low readability and text quality and contain much redundant information. This is in marked contrast with hand-written summaries which often combine several pieces of information from the original document (Jing, 2002) and exhibit many rewrite operations such as substitutions, insertions, deletions, or reorderings.

Sentence compression is often regarded as a promising first step towards ameliorating some of the problems associated with extractive summarization. The task is commonly expressed as a word deletion problem. It involves creating a short grammatical summary of a *single* sentence, by removing elements that are considered extraneous, while retaining the most important information (Knight and Marcu, 2002). Interfacing extractive summarization with a sentence compression module could improve the conciseness of the generated summaries and render them more informative (Jing, 2000; Lin, 2003; Zajic et al., 2007).

Despite the bulk of work on sentence compression and summarization (see Clarke and Lapata 2008 and Mani 2001 for overviews) only a handful of approaches attempt to do both in a joint model (Daumé III and Marcu, 2002; Daumé III, 2006; Lin, 2003; Martins and Smith, 2009). One reason for this might be the performance of sentence compression systems which falls short of attaining grammaticality levels of human output. For example, Clarke and Lapata (2008) evaluate a range of state-of-the-art compression systems across different domains and show that machine generated compressions are consistently perceived as worse than the human gold standard. Another reason is the summarization objective itself. If our goal is to summarize news articles, then we may be better off selecting the first n sentences of the document. This "lead" baseline may err on the side of verbosity but at least will be grammatical, and it has indeed proved extremely hard to outperform by more sophisticated methods (Nenkova, 2005).

In this paper we propose a model for sum-

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 565–574,
Uppsala, Sweden, 11-16 July 2010.

marization that incorporates compression into the task. A key insight in our approach is to formulate summarization as a *phrase* rather than *sentence* extraction problem. Compression falls naturally out of this formulation as only phrases deemed important should appear in the summary. Obviously, our output summaries must meet additional requirements such as sentence length, overall length, topic coverage and, importantly, grammaticality. We combine phrase and dependency information into a single data structure, which allows us to express grammaticality as constraints across phrase dependencies. We encode these constraints through the use of integer linear programming (ILP), a well-studied optimization framework that is able to search the entire solution space efficiently.

We apply our model to the task of generating highlights for a single document. Examples of CNN news articles with human-authored highlights are shown in Table 1. Highlights give a brief overview of the article to allow readers to quickly gather information on stories, and usually appear as bullet points. Importantly, they represent the gist of the *entire* document and thus often differ substantially from the first n sentences in the article (Svore et al., 2007). They are also highly compressed, written in a telegraphic style and thus provide an excellent testbed for models that generate compressed summaries. Experimental results show that our model's output is comparable to hand-written highlights both in terms of grammaticality and informativeness.

2 Related work

Much effort in automatic summarization has been devoted to sentence extraction which is often formalized as a classification task (Kupiec et al., 1995). Given appropriately annotated training data, a binary classifier learns to predict for each document sentence if it is worth extracting. Surface-level features are typically used to single out important sentences. These include the presence of certain key phrases, the position of a sentence in the original document, the sentence length, the words in the title, the presence of proper nouns, etc. (Mani, 2001; Sparck Jones, 1999).

Relatively little work has focused on extraction methods for units smaller than sentences. Jing and McKeown (2000) first extract sentences, then remove redundant phrases, and use (manual) recombination rules to produce coherent output. Wan and Paris (2008) segment sentences heuristically into clauses *before* extraction takes place, and show that this improves summarization quality. In the context of multiple-document summarization, heuristics have also been used to remove parenthetical information (Conroy et al., 2004; Siddharthan et al., 2004). Witten et al. (1999) (among others) extract keyphrases to capture the gist of the document, without however attempting to reconstruct sentences or generate summaries.

A few previous approaches have attempted to interface sentence compression with summarization. A straightforward way to achieve this is by adopting a two-stage architecture (e.g., Lin 2003) where the sentences are first extracted and then compressed or the other way round. Other work implements a *joint* model where words and sentences are deleted simultaneously from a document. Using a noisy-channel model, Daumé III and Marcu (2002) exploit the discourse structure of a document and the syntactic structure of its sentences in order to decide which constituents to drop but also which discourse units are unimportant. Martins and Smith (2009) formulate a joint sentence extraction and summarization model as an ILP. The latter optimizes an objective function consisting of two parts: an extraction component, essentially a non-greedy variant of maximal marginal relevance (McDonald, 2007), and a sentence compression component, a more compact reformulation of Clarke and Lapata (2008) based on the output of a dependency parser. Compression and extraction models are trained separately in a max-margin framework and then interpolated. In the context of multi-document summarization, Daumé III's (2006) vine-growth model creates summaries incrementally, either by starting a new sentence or by growing already existing ones.

Our own work is closest to Martins and Smith (2009). We also develop an ILP-based compression and summarization model, however, several key differences set our approach apart. Firstly, content selection is performed at the phrase rather than sentence level. Secondly, the combination of phrase and dependency information into a single data structure is new, and important in allowing us to express grammaticality as constraints across phrase dependencies, rather than resorting to a lan-

Most blacks say MLK's vision fulfilled, poll finds	**9/11 billboard draws flak from Florida Democrats, GOP**
WASHINGTON (CNN) – More than two-thirds of African-Americans believe Martin Luther King Jr.'s vision for race relations has been fulfilled, a CNN poll found – a figure up sharply from a survey in early 2008. The CNN-Opinion Research Corp. survey was released Monday, a federal holiday honoring the slain civil rights leader and a day before Barack Obama is to be sworn in as the first black U.S. president. The poll found 69 percent of blacks said King's vision has been fulfilled in the more than 45 years since his 1963 'I have a dream' speech – roughly double the 34 percent who agreed with that assessment in a similar poll taken last March. But whites remain less optimistic, the survey found.	(CNN) – A Florida man is using billboards with an image of the burning World Trade Center to encourage votes for a Republican presidential candidate, drawing criticism for politicizing the 9/11 attacks. 'Please Don't Vote for a Democrat' reads the type over the picture of the twin towers after hijacked airliners hit them on September, 11, 2001. Mike Meehan, a St. Cloud, Florida, businessman who paid to post the billboards in the Orlando area, said former President Clinton should have put a stop to Osama bin Laden and al Qaeda before 9/11. He said a Republican president would have done so.
• *69 percent of blacks polled say Martin Luther King Jr's vision realized.* • *Slim majority of whites say King's vision not fulfilled.* • *King gave his "I have a dream" speech in 1963.*	• *Billboards use image from 9/11 to encourage GOP votes.* • *9/11 image wrong for ad, say Florida political parties.* • *Floridian praises President Bush, says ex-President Clinton failed to stop al Qaeda.*

Table 1: Two example CNN news articles, showing the title and the first few paragraphs, and below, the original highlights that accompanied each story.

guage model. Lastly, our model is more compact, has fewer parameters, and does not require two training procedures. Our approach bears some resemblance to headline generation (Dorr et al., 2003; Banko et al., 2000), although we output several sentences rather than a single one. Headline generation models typically extract individual words from a document to produce a very short summary, whereas we extract phrases and ensure that they are combined into grammatical sentences through our ILP constraints.

Svore et al. (2007) were the first to foreground the highlight generation task which we adopt as an evaluation testbed for our model. Their approach is however a purely extractive one. Using an algorithm based on neural networks and third-party resources (e.g., news query logs and Wikipedia entries) they rank sentences and select the three highest scoring ones as story highlights. In contrast, we aim to generate rather than extract highlights. As a first step we focus on deleting extraneous material, but other more sophisticated rewrite operations (e.g., Cohn and Lapata 2009) could be incorporated into our framework.

3 The Task

Given a document, we aim to produce three or four short sentences covering its main topics, much like the "Story Highlights" accompanying the (online) CNN news articles. CNN highlights are written by humans; we aim to do this automatically.

	Documents	Highlights
Sentences	37.2 ± 39.6	3.5 ± 0.5
Tokens	795.0 ± 744.8	47.0 ± 9.6
Tokens/sentence	22.4 ± 4.2	13.3 ± 1.7

Table 2: Overview statistics on the corpus of documents and highlights (mean and standard deviation). A minority of documents are transcripts of interviews and speeches, and can be very long; this accounts for the very large standard deviation.

Two examples of a news story and its associated highlights, are shown in Table 1. As can be seen, the highlights are written in a compressed, almost telegraphic manner. Articles, auxiliaries and forms of the verb *be* are often deleted. Compression is also achieved through paraphrasing, e.g., substitutions and reorderings. For example, the document sentence "*The poll found 69 percent of blacks said King's vision has been fulfilled.*" is rephrased in the highlight as "*69 percent of blacks polled say Martin Luther King Jr's vision realized.*". In general, there is a fair amount of lexical overlap between document sentences and highlights (42.44%) but the correspondence between document sentences and highlights is not always one-to-one. In the first example in Table 1, the second paragraph gives rise to two highlights. Also note that the highlights need not form a coherent summary, each of them is relatively stand-alone, and there is little co-referencing between them.

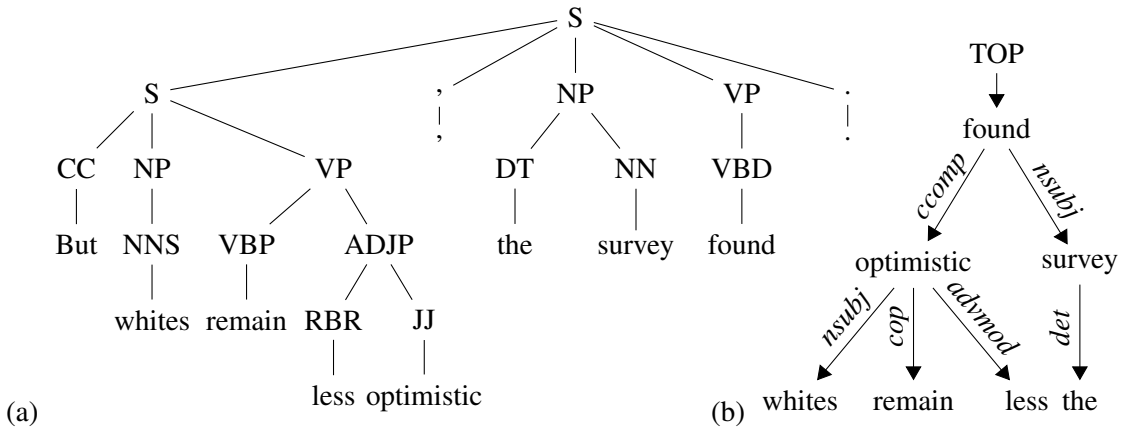

Figure 1: An example phrase structure (a) and dependency (b) tree for the sentence "*But whites remain less optimistic, the survey found.*".

In order to train and evaluate the model presented in the following sections we created a corpus of document-highlight pairs (approximately 9,000) which we downloaded from the CNN.com website.[1] The articles were randomly sampled from the years 2007–2009 and covered a wide range of topics such as business, crime, health, politics, showbiz, etc. The majority were news articles, but the set also contained a mixture of editorials, commentary, interviews and reviews. Some overview statistics of the corpus are shown in Table 2. Overall, we observe a high degree of compression both at the document and sentence level. The highlights summary tends to be ten times shorter than the corresponding article. Furthermore, individual highlights have almost half the length of document sentences.

4 Modeling

The objective of our model is to create the most informative story highlights possible, subject to constraints relating to sentence length, overall summary length, topic coverage, and grammaticality. These constraints are *global* in their scope, and cannot be adequately satisfied by optimizing each one of them individually. Our approach therefore uses an ILP formulation which will provide a globally optimal solution, and which can be efficiently solved using standard optimization tools. Specifically, the model selects phrases from which to form the highlights, and each highlight is created from a single sentence through phrase deletion. The model operates on parse trees augmented with dependency labels. We first describe how we obtain this representation and then move on to discuss the model in more detail.

Sentence Representation We obtain syntactic information by parsing every sentence twice, once with a phrase structure parser and once with a dependency parser. The phrase structure and dependency-based representations for the sentence "*But whites remain less optimistic, the survey found.*" (from Table 1) are shown in Figures 1(a) and 1(b), respectively.

We then combine the output from the two parsers, by mapping the dependencies to the edges of the phrase structure tree in a greedy fashion, shown in Figure 2(a). Starting at the top node of the dependency graph, we choose a node i and a dependency arc to node j. We locate the corresponding words i and j on the phrase structure tree, and locate their nearest shared ancestor p. We assign the label of the dependency $i \rightarrow j$ to the first unlabeled edge from p to j in the phrase structure tree. Edges assigned with dependency labels are shown as dashed lines. These edges are important to our formulation, as they will be represented by binary decision variables in the ILP. Further edges from p to j, and all the edges from p to i, are marked as *fixed* and shown as solid lines. In this way we keep the correct ordering of leaf nodes. Finally, leaf nodes are merged into parent phrases, until each phrase node contains a minimum of two tokens, shown in Figure 2(b). Because of this minimum length rule, it is possible for a merged node to be a clause rather than a phrase, but in the subsequent description we will use the term *phrase* rather loosely to describe any merged leaf node.

[1] The corpus is available from `http://homepages.inf.ed.ac.uk/mlap/resources/index.html`.

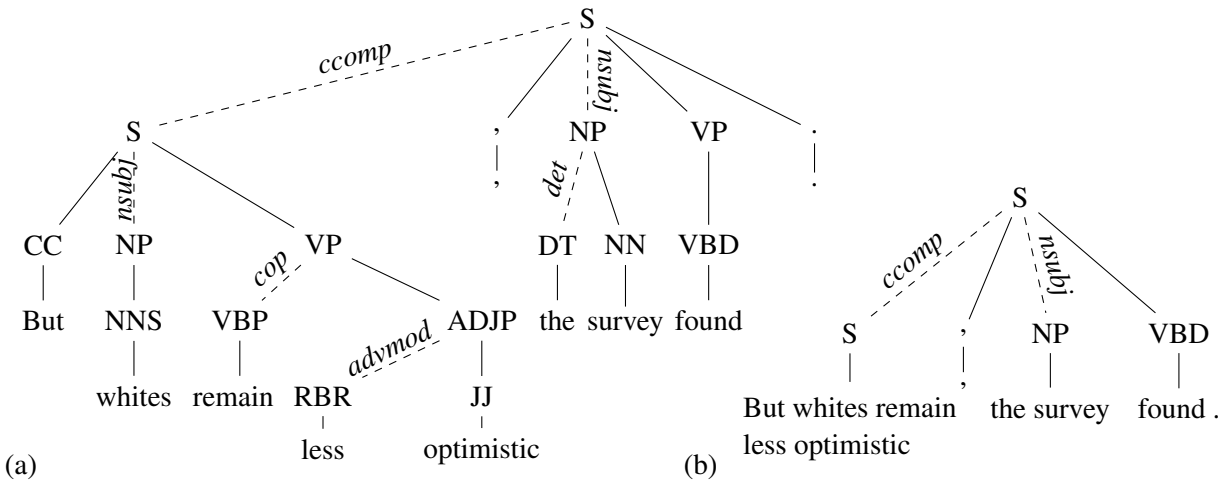

Figure 2: Dependencies are mapped onto phrase structure tree (a) and leaf nodes are merged with parent phrases (b).

ILP model The merged phrase structure tree, such as shown in Figure 2(b), is the actual input to our model. Each phrase in the document is given a *salience score*. We obtain these scores from the output of a supervised machine learning algorithm that predicts for each phrase whether it should be included in the highlights or not (see Section 5 for details). Let $\mathcal{S}$ be the set of sentences in a document, $\mathcal{P}$ be the set of phrases, and $\mathcal{P}_s \subset \mathcal{P}$ be the set of phrases in each sentence $s \in \mathcal{S}$. $\mathcal{T}$ is the set of words with the highest tf.idf scores, and $\mathcal{P}_t \subset \mathcal{P}$ is the set of phrases containing the token $t \in \mathcal{T}$. Let f_i denote the salience score for phrase i, determined by the machine learning algorithm, and l_i is its length in tokens.

We use a vector of binary variables $x \in \{0,1\}^{|\mathcal{P}|}$ to indicate if each phrase is to be within a highlight. These are either top-level nodes in our merged tree representation, or nodes whose edge to the parent has a dependency label (the dashed lines). Referring to our example in Figure 2(b), binary variables would be allocated to the top-level S node, the child S node and the NP node. The vector of auxiliary binary variables $y \in \{0,1\}^{|\mathcal{S}|}$ indicates from which sentences the chosen phrases come (see Equations (1i) and (1j)). Let the sets $\mathcal{D}_i \subset \mathcal{P}, \forall i \in \mathcal{P}$ capture the phrase dependency information for each phrase i, where each set $\mathcal{D}_i$ contains the phrases that depend on the presence of i. Our objective function function is given in Equation (1a): it is the sum of the salience scores of all the phrases chosen to form the highlights of a given document, subject to the constraints in Equations (1b)–(1j). The latter provide a natural way of describing the requirements the output must meet.

$$\max_x \quad \sum_{i \in \mathcal{P}} f_i x_i \tag{1a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{P}} l_i x_i \leq L_T \tag{1b}$$

$$\sum_{i \in \mathcal{P}_s} l_i x_i \leq L_M y_s \qquad \forall s \in \mathcal{S} \tag{1c}$$

$$\sum_{i \in \mathcal{P}_s} l_i x_i \geq L_m y_s \qquad \forall s \in \mathcal{S} \tag{1d}$$

$$\sum_{i \in \mathcal{P}_t} x_i \geq 1 \qquad \forall t \in \mathcal{T} \tag{1e}$$

$$x_j \rightarrow x_i \qquad \forall i \in \mathcal{P}, j \in \mathcal{D}_i \tag{1f}$$

$$x_i \rightarrow y_s \qquad \forall s \in \mathcal{S}, i \in \mathcal{P}_s \tag{1g}$$

$$\sum_{s \in \mathcal{S}} y_s \leq N_S \tag{1h}$$

$$x_i \in \{0,1\} \qquad \forall i \in \mathcal{P} \tag{1i}$$

$$y_s \in \{0,1\} \qquad \forall s \in \mathcal{S}. \tag{1j}$$

Constraint (1b) ensures that the generated highlights do not exceed a total budget of L_T tokens. This constraint may vary depending on the application or task at hand. Highlights on a small screen device would presumably be shorter than highlights for news articles on the web. It is also possible to set the length of each highlight to be within the range $[L_m, L_M]$. Constraints (1c) and (1d) enforce this requirement. In particular, these constraints stop highlights formed from sentences at the beginning of the document (which tend to have

high salience scores) from being too long. Equation (1e) is a set-covering constraint, requiring that each of the words in $\mathcal{T}$ appears at least once in the highlights. We assume that words with high tf.idf scores reveal to a certain extent what the document is about. Constraint (1e) ensures that some of these words will be present in the highlights.

We enforce grammatical correctness through constraint (1f) which ensures that the phrase dependencies are respected. Phrases that depend on phrase i are contained in the set $\mathcal{D}_i$. Variable x_i is true, and therefore phrase i will be included, if any of its dependents $x_j \in \mathcal{D}_i$ are true. The phrase dependency constraints, contained in the set $\mathcal{D}_i$ and enforced by (1f), are the result of two rules based on the typed dependency information:

1. Any child node j of the current node i, whose connecting edge $i \rightarrow j$ is of type *nsubj* (nominal subject), *nsubjpass* (passive nominal subject), *dobj* (direct object), *pobj* (preposition object), *infmod* (infinitival modifier), *ccomp* (clausal complement), *xcomp* (open clausal complement), *measure* (measure phrase modifier) and *num* (numeric modifier) must be included if node i is included.

2. The parent node p of the current node i must always be included if i is, unless the edge $p \rightarrow i$ is of type *ccomp* (clausal complement) or *advcl* (adverbial clause), in which case it is possible to include i without including p.

Consider again the example in Figure 2(b). There are only two possible outputs from this sentence. If the phrase "*the survey*" is chosen, then the parent node "*found*" will be included, and from our first rule the *ccomp* phrase must also be included, which results in the output: "*But whites remain less optimistic, the survey found.*" If, on the other hand, the clause "*But whites remain less optimistic*" is chosen, then due to our second rule there is no constraint that forces the parent phrase "*found*" to be included in the highlights. Without other factors influencing the decision, this would give the output: "*But whites remain less optimistic.*" We can see from this example that encoding the possible outputs as decisions on branches of the phrase structure tree provides a more compact representation of many options than would be possible with an explicit enumeration of all possible compressions. Which output is chosen (if any) depends on the scores of the phrases involved, and the influence of the other constraints.

Constraint (1g) tells the ILP to create a highlight if one of its constituent phrases is chosen. Finally, note that a maximum number of highlights N_S can be set beforehand, and (1h) limits the highlights to this maximum.

5 Experimental Set-up

Training We obtained phrase-based salience scores using a supervised machine learning algorithm. 210 document-highlight pairs were chosen randomly from our corpus (see Section 3). Two annotators manually aligned the highlights and document sentences. Specifically, each sentence in the document was assigned one of three alignment labels: must be in the summary (1), could be in the summary (2), and is not in the summary (3). The annotators were asked to label document sentences whose content was identical to the highlights as "must be in the summary", sentences with partially overlapping content as "could be in the summary" and the remainder as "should not be in the summary". Inter-annotator agreement was .82 ($p < 0.01$, using Spearman's ρ rank correlation). The mapping of sentence labels to phrase labels was unsupervised: if the phrase came from a sentence labeled (1), and there was a unigram overlap (excluding stop words) between the phrase and any of the original highlights, we marked this phrase with a positive label. All other phrases were marked negative.

Our feature set comprised surface features such as sentence and paragraph position information, POS tags, unigram and bigram overlap with the title, and whether high-scoring tf.idf words were present in the phrase (66 features in total). The 210 documents produced a training set of 42,684 phrases (3,334 positive and 39,350 negative). We learned the feature weights with a linear SVM, using the software SVM-OOPS (Woodsend and Gondzio, 2009). This tool gave us directly the feature weights as well as support vector values, and it allowed different penalties to be applied to positive and negative misclassifications, enabling us to compensate for the unbalanced data set. The penalty hyper-parameters chosen were the ones that gave the best F-scores, using 10-fold validation.

Highlight generation We generated highlights for a test set of 600 documents. We created and

solved an ILP for each document. Sentences were first tokenized to separate words and punctuation, then parsed to obtain phrases and dependencies as described in Section 4 using the Stanford parser (Klein and Manning, 2003). For each phrase, features were extracted and salience scores calculated from the feature weights determined through SVM training. The distance from the SVM hyperplane represents the salience score. The ILP model (see Equation (1)) was parametrized as follows: the maximum number of highlights N_S was 4, the overall limit on length L_T was 75 tokens, the length of each highlight was in the range of $[8,28]$ tokens, and the topic coverage set $\mathcal{T}$ contained the top 5 tf.idf words. These parameters were chosen to capture the properties seen in the majority of the training set; they were also relaxed enough to allow a feasible solution of the ILP model (with hard constraints) for all the documents in the test set. To solve the ILP model we used the ZIB Optimization Suite software (Achterberg, 2007; Koch, 2004; Wunderling, 1996). The solution was converted into highlights by concatenating the chosen leaf nodes in order. The ILP problems we created had on average 290 binary variables and 380 constraints. The mean solve time was 0.03 seconds.

Summarization In order to examine the generality of our model and compare with previous work, we also evaluated our system on a vanilla summarization task. Specifically, we used the same model (trained on the CNN corpus) to generate summaries for the DUC-2002 corpus[2]. We report results on the entire dataset and on a subset containing 140 documents. This is the same partition used by Martins and Smith (2009) to evaluate their ILP model.[3]

Baselines We compared the output of our model to two baselines. The first one simply selects the "leading" three sentences from each document (without any compression). The second baseline is the output of a sentence-based ILP model, similar to our own, but simpler. The model is given in (2). The binary decision variables $x \in \{0,1\}^{|\mathcal{S}|}$ now represent sentences, and f_i the salience score for each sentence. The objective again is to maximize the total score, but now subject only to tf.idf coverage (2b) and a limit on the number of highlights (2c) which we set to 3. There are no sentence length or grammaticality constraints, as there is no sentence compression.

$$\max_{x} \quad \sum_{i \in \mathcal{S}} f_i x_i \tag{2a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{S}_t} x_i \geq 1 \qquad \forall t \in \mathcal{T} \tag{2b}$$

$$\sum_{i \in \mathcal{S}} x_i \leq N_S \tag{2c}$$

$$x_i \in \{0,1\} \qquad \forall i \in \mathcal{S}. \tag{2d}$$

The SVM was trained with the same features used to obtain phrase-based salience scores, but with sentence-level labels (labels (1) and (2) positive, (3) negative).

Evaluation We evaluated summarization quality using ROUGE (Lin and Hovy, 2003). For the highlight generation task, the original CNN highlights were used as the reference. We report unigram overlap (ROUGE-1) as a means of assessing informativeness and the longest common subsequence (ROUGE-L) as a means of assessing fluency.

In addition, we evaluated the generated highlights by eliciting human judgments. Participants were presented with a news article and its corresponding highlights and were asked to rate the latter along three dimensions: informativeness (do the highlights represent the article's main topics?), grammaticality (are they fluent?), and verbosity (are they overly wordy and repetitive?). The subjects used a seven point rating scale. An ideal system would receive high numbers for grammaticality and informativeness and a low number for verbosity. We randomly selected nine documents from the test set and generated highlights with our model and the sentence-based ILP baseline. We also included the original highlights as a gold standard. We thus obtained ratings for 27 (9×3) document-highlights pairs.[4] The study was conducted over the Internet using WebExp (Keller et al., 2009) and was completed by 34 volunteers, all self reported native English speakers.

With regard to the summarization task, following Martins and Smith (2009), we used ROUGE-1 and ROUGE-2 to evaluate our system's output. We also report results with ROUGE-L. Each document in the DUC-2002 dataset is paired with

[2] http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html

[3] We are grateful to André Martins for providing us with details of their testing partition.

[4] A Latin square design ensured that subjects did not see two different highlights of the same document.

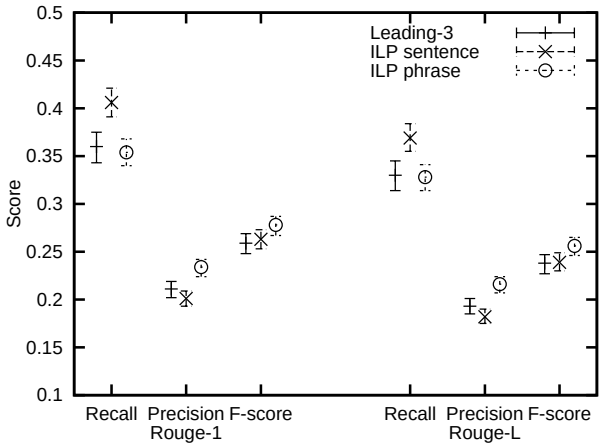

Figure 3: ROUGE-1 and ROUGE-L results for phrase-based ILP model and two baselines, with error bars showing 95% confidence levels.

a human-authored summary (approximately 100 words) which we used as reference.

6 Results

We report results on the highlight generation task in Figure 3 with ROUGE-1 and ROUGE-L (error bars indicate the 95% confidence interval). In both measures, the ILP sentence baseline has the best recall, while the ILP phrase model has the best precision (the differences are statistically significant). F-score is higher for the phrase-based system but not significantly. This can be attributed to the fact that the longer output of the sentence-based model makes the recall task easier. Average highlight lengths are shown in Table 3, and the compression rates they represent. Our phrase model achieves the highest compression rates, whereas the sentence-based model tends to select long sentences even in comparison to the lead baseline. The sentence ILP model outperforms the lead baseline with respect to recall but not precision or F-score. The phrase ILP achieves a significantly better F-score over the lead baseline with both ROUGE-1 and ROUGE-L.

The results of our human evaluation study are summarized in Table 4. There was no statistically significant difference in the grammaticality between the highlights generated by the phrase ILP system and the original CNN highlights (means differences were compared using a Post-hoc Tukey test). The grammaticality of the sentence ILP was significantly higher overall as no compression took place ($\alpha < 0.05$). All three

	s	toks/s	C.R.
Articles	36.5	22.2 ± 4.0	100%
CNN highlights	3.5	13.3 ± 1.7	5.8%
ILP phrase	3.8	18.0 ± 2.9	8.4%
Leading-3	3.0	25.1 ± 7.4	9.3%
ILP sentence	3.0	31.3 ± 7.9	11.6%

Table 3: Comparison of output lengths: number of sentences, tokens per sentence, and compression rate, for CNN articles, their highlights, the ILP phrase model, and two baselines.

Model	Grammar	Importance	Verbosity
CNN highlights	4.85	4.88	3.14
ILP sentence	6.41	5.47	3.97
ILP phrase	5.53	5.05	3.38

Table 4: Average human ratings for original CNN highlights, and two ILP models.

systems performed on a similar level with respect to importance (differences in the means were not significant). The highlights created by the sentence ILP were considered significantly more verbose ($\alpha < 0.05$) than those created by the phrase-based system and the CNN abstractors. Overall, the highlights generated by the phrase ILP model were not significantly different from those written by humans. They capture the same content as the full sentences, albeit in a more succinct manner. Table 5 shows the output of the phrase-based system for the documents in Table 1.

Our results on the complete DUC-2002 corpus are shown in Table 6. Despite the fact that our model has not been optimized for the original task of generating 100-word summaries—instead it is trained on the CNN corpus, and generates highlights—the results are comparable with the best of the original participants[5] in each of the ROUGE measures. Our model is also significantly better than the lead sentences baseline.

Table 7 presents our results on the same DUC-2002 partition (140 documents) used by Martins and Smith (2009). The phrase ILP model achieves a significantly better F-score (for both ROUGE-1 and ROUGE-2) over the lead baseline, the sentence ILP model, and Martins and Smith. We should point out that the latter model is not a straw man. It significantly outperforms a pipeline

[5]The list of participants is on page 12 of the slides available from `http://duc.nist.gov/pubs/2002slides/overview.02.pdf`.

- *More than two-thirds of African-Americans believe Martin Luther King Jr.'s vision for race relations has been fulfilled.*
- *69 percent of blacks said King's vision has been fulfilled in the more than 45 years since his 1963 'I have a dream' speech.*
- *But whites remain less optimistic, the survey found.*

- *A Florida man is using billboards with an image of the burning World Trade Center to encourage votes for a Republican presidential candidate, drawing criticism.*
- *'Please Don't Vote for a Democrat' reads the type over the picture of the twin towers.*
- *Mike Meehan said former President Clinton should have put a stop to Osama bin Laden and al Qaeda before 9/11.*

Table 5: Generated highlights for the stories in Table 1 using the phrase ILP model.

Participant	ROUGE-1	ROUGE-2	ROUGE-L
28	0.464	0.222	0.432
19	0.459	0.221	0.431
21	0.458	0.216	0.426
29	0.449	0.208	0.419
27	0.445	0.209	0.417
Leading-3	0.416	0.200	0.390
ILP phrase	0.454	0.213	0.428

Table 6: ROUGE results on the complete DUC-2002 corpus, including the top 5 original participants. For all results, the 95% confidence interval is $\pm$0.008.

approach that first creates extracts and then compresses them. Furthermore, as a standalone sentence compression system it yields state of the art performance, comparable to McDonald's (2006) discriminative model and superior to Hedge Trimmer (Zajic et al., 2007), a less sophisticated deterministic system.

7 Conclusions

In this paper we proposed a joint content selection and compression model for single-document summarization. A key aspect of our approach is the representation of content by phrases rather than entire sentences. Salient phrases are selected to form the summary. Grammaticality, length and coverage requirements are encoded as constraints in an integer linear program. Applying the model to the generation of "story highlights" (and single document summaries) shows that it is a viable alternative to extraction-based systems. Both ROUGE scores and the results of our human study

	ROUGE-1	ROUGE-2	ROUGE-L
Leading-3	.400 $\pm$ .018	.184 $\pm$ .015	.374 $\pm$ .017
M&S (2009)	.403 $\pm$ .076	.180 $\pm$ .076	—
ILP sentence	.430 $\pm$ .014	.191 $\pm$ .015	.401 $\pm$ .014
ILP phrase	.445 $\pm$ .014	.200 $\pm$ .014	.419 $\pm$ .014

Table 7: ROUGE results on DUC-2002 corpus (140 documents). —: only ROUGE-1 and ROUGE-2 results are given in Martins and Smith (2009).

confirm that our system manages to create summaries at a high compression rate and yet maintain the informativeness and grammaticality of a competitive extractive system. The model itself is relatively simple and knowledge-lean, and achieves good performance without reference to any resources outside the corpus collection.

Future extensions are many and varied. An obvious next step is to examine how the model generalizes to other domains and text genres. Although coherence is not so much of an issue for highlights, it certainly plays a role when generating standard summaries. The ILP model can be straightforwardly augmented with discourse constraints similar to those proposed in Clarke and Lapata (2007). We would also like to generalize the model to arbitrary rewrite operations, as our results indicate that compression rates are likely to improve with more sophisticated paraphrasing.

Acknowledgments

We would like to thank Andreas Grothey and members of ICCS at the School of Informatics for the valuable discussions and comments throughout this work. We acknowledge the support of EPSRC through project grants EP/F055765/1 and GR/T04540/01.

References

Achterberg, Tobias. 2007. *Constraint Integer Programming*. Ph.D. thesis, Technische Universität Berlin.

Banko, Michele, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th ACL*. Hong Kong, pages 318–325.

Clarke, James and Mirella Lapata. 2007. Modelling compression with discourse constraints. In *Proceedings of EMNLP-CoNLL*. Prague, Czech Republic, pages 1–11.

Clarke, James and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research* 31:399–429.

Cohn, Trevor and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research* 34:637–674.

Conroy, J. M., J. D. Schlesinger, J. Goldstein, and D. P. O'Leary. 2004. Left-brain/right-brain multi-document summarization. In *DUC 2004 Conference Proceedings*.

Daumé III, Hal. 2006. *Practical Structured Learning Techniques for Natural Language Processing*. Ph.D. thesis, University of Southern California.

Daumé III, Hal and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the 40th ACL*. Philadelphia, PA, pages 449–456.

Dorr, Bonnie, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Summarization*. pages 1–8.

Jing, Hongyan. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the 6th ANLP*. Seattle, WA, pages 310–315.

Jing, Hongyan. 2002. Using hidden Markov modeling to decompose human-written summaries. *Computational Linguistics* 28(4):527–544.

Jing, Hongyan and Kathleen McKeown. 2000. Cut and paste summarization. In *Proceedings of the 1st NAACL*. Seattle, WA, pages 178–185.

Keller, Frank, Subahshini Gunasekharan, Neil Mayo, and Martin Corley. 2009. Timing accuracy of web experiments: A case study using the WebExp software package. *Behavior Research Methods* 41(1):1–12.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st ACL*. Sapporo, Japan, pages 423–430.

Knight, Kevin and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence* 139(1):91–107.

Koch, Thorsten. 2004. *Rapid Mathematical Prototyping*. Ph.D. thesis, Technische Universität Berlin.

Kupiec, Julian, Jan O. Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of SIGIR-95*. Seattle, WA, pages 68–73.

Lin, Chin-Yew. 2003. Improving summarization performance by sentence compression — a pilot study. In *Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages*. Sapporo, Japan, pages 1–8.

Lin, Chin-Yew and Eduard H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT NAACL*. Edmonton, Canada, pages 71–78.

Mani, Inderjeet. 2001. *Automatic Summarization*. John Benjamins Pub Co.

Martins, André 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*. Boulder, Colorado, pages 1–9.

McDonald, Ryan. 2006. Discriminative sentence compression with soft syntactic constraints. In *Proceedings of the 11th EACL*. Trento, Italy.

McDonald, Ryan. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th ECIR*. Rome, Italy.

Nenkova, Ani. 2005. Automatic text summarization of newswire: Lessons learned from the Document Understanding Conference. In *Proceedings of the 20th AAAI*. Pittsburgh, PA, pages 1436–1441.

Siddharthan, Advaith, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*. pages 896–902.

Sparck Jones, Karen. 1999. Automatic summarizing: Factors and directions. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, MIT Press, Cambridge, pages 1–33.

Svore, Krysta, Lucy Vanderwende, and Christopher Burges. 2007. Enhancing single-document summarization by combining RankNet and third-party sources. In *Proceedings of EMNLP-CoNLL*. Prague, Czech Republic, pages 448–457.

Wan, Stephen and Cécile Paris. 2008. Experimenting with clause segmentation for text summarization. In *Proceedings of the 1st TAC*. Gaithersburg, MD.

Witten, Ian H., Gordon Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM International Conference on Digital Libraries*. Berkeley, CA, pages 254–255.

Woodsend, Kristian and Jacek Gondzio. 2009. Exploiting separability in large-scale linear support vector machine training. *Computational Optimization and Applications* .

Wunderling, Roland. 1996. *Paralleler und objektorientierter Simplex-Algorithmus*. Ph.D. thesis, Technische Universität Berlin.

Zajic, David, Bonnie J. Door, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing Management Special Issue on Summarization* 43(6):1549–1570.