Multiple Aspect Summarization Using Integer Linear Programming

Kristian Woodsend and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
k.woodsend@ed.ac.uk, mlap@inf.ed.ac.uk

Abstract

Multi-document summarization involves many aspects of content selection and surface realization. The summaries must be informative, succinct, grammatical, and obey stylistic writing conventions. We present a method where such individual aspects are learned separately from data (without any hand-engineering) but optimized jointly using an integer linear programme. The ILP framework allows us to combine the decisions of the expert learners and to select and rewrite source content through a mixture of objective setting, soft and hard constraints. Experimental results on the TAC-08 data set show that our model achieves state-of-the-art performance using ROUGE and significantly improves the informativeness of the summaries.

1 Introduction

Automatic summarization has enjoyed wide popularity in natural language processing (see the proceedings of the Document Understanding and Text Analysis conferences) due to its potential for practical applications but also because it incorporates many important aspects of both natural language understanding and generation. Of the many summarization paradigms that have been identified over the years (see Sparck Jones (1999) and Mani (2001) for comprehensive overviews), multi-document summarization — the task of producing summaries from clusters of thematically related documents — has consistently attracted attention.

Despite considerable research effort, the automatic generation of multi-document summaries that resemble those written by humans remains challenging. This is primarily due to the task itself which is complex and subject to several constraints: the summary must be maximally informative and minimally redundant, grammatical, coherent, adhere to a pre-specified length and stylistic conventions. An ideal model would learn to output summaries that simultaneously meet all these constraints from data (i.e., document clusters and their corresponding summaries). This *global inference problem* is, however, hard — the solution space is large and the lack of easily accessible datasets an obstacle to joint learning. It is thus no surprise that previous work has focused on specific aspects of joint learning.

Initial global formulations of the multi-document summarization task focused on extractive summarization and used approximate greedy algorithms for finding the sentences of the summary. Goldstein et al. (2000) search for the set of sentences that are both relevant and non-redundant, whereas Filatova and Hatzivassiloglou (2004) model multi-document summarization as an instance of the maximum coverage set problem.[1] More recent work improves on the search problem by considering exact solutions and permits a limited amount of rewriting. McDonald (2007) proposes an integer linear programming formulation that maximizes the sum of relevance scores of the selected sentences penalized by the

[1] Given C, a finite set of weighted elements, a collection T of subsets of C, and an integer k, find those k sets that maximize the total number of elements in the union of T's members (Hochba, 1997).

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 233–243, Jeju Island, Korea, 12–14 July 2012.

sum of redundancy scores of all pairs of selected sentences. Gillick et al. (2008) develop an exact solution for a model similar to Filatova and Hatzivassiloglou (2004) under the assumption that the value of a summary is the sum of values of the unique concepts (approximated by bigrams) it contains. Subsequent work (Gillick et al., 2009; Berg-Kirkpatrick et al., 2011) extends this model to allow sentence compression in the form of word or constituent deletion.

In this paper we propose a model for multi-document summarization that attempts to cover many different aspects of the task such as content selection, surface realization, paraphrasing, and stylistic conventions. These aspects are learned separately using specific "expert" predictors, but are optimized jointly using an integer linear programming model (ILP) to generate the output summary.[2] All experts are learned from data without requiring additional annotation over and above the summaries written for each document cluster. Our predictors include the use of unique bigram information to model content and avoid redundancy, positional information to model important and poor locations of content, and language modeling to capture stylistic conventions. Learning each predictor separately gives better generalization, while the ILP framework allows us to combine the decisions of the expert learners through the use of objectives, hard and soft constraints.

The experts work collaboratively to rewrite the content using rules extracted from document clusters and model summaries. We adopt the synchronous tree substitution grammar (STSG) formalism (Eisner, 2003) which can model non-isomorphic tree structures (the grammar rules can comprise trees of arbitrary depth) and is thus suited to text-rewriting tasks which typically involve a number of local modifications to the input text. Specifically, we propose quasi-synchronous tree substitution grammar (QTSG) as a flexible formalism to learn general tree-edits from *loosely-aligned* phrase structure trees.

We evaluate our model on the 100-word "non-update" summarization task as defined in the the Text Analysis Conference (TAC 2008). Experimental results show that our method obtains performance comparable and in some cases superior to state-of-the-art, in terms of ROUGE and human ratings of summary grammaticality and informativeness. Importantly, there is nothing inherent in our model that is specific to this particular summarization task. As all of the different experts are learned from data, it could easily adapt to other summarization styles or conventions as needed.

[2]Our task is standard multi-document summarization and should not be confused with "guided" summarization where system and human summarizers are given a list of important aspects to cover in the summary. Our usage of the term aspects broadly refers to the different types of constraints (e.g., relating to content or style) a summary must meet, but these are learned rather than specified in advance.

2 Related work

Recent years have seen increased interest in global inference methods for summarization. ILP-based models have been developed for several subtasks ranging from sentence compression (Clarke and Lapata, 2008), to single- and multi-document summarization (McDonald, 2007; Martins and Smith, 2009; Gillick and Favre, 2009; Woodsend and Lapata, 2010; Berg-Kirkpatrick et al., 2011), and headline generation (Deshpande et al., 2007; Woodsend et al., 2010). Most of these approaches are either purely extractive or implement a single rewrite operation, namely word deletion. Although it is well-known that hand-written summaries often exhibit additional edits and sentence recombinations (Jing, 2002), the challenges involved in acquiring the rewrite rules, interfacing them with inference, and ensuring grammatical output make the development of abstractive models non-trivial.

Our work is closest to Gillick et al. (2008) who also develop an ILP model for multi-document summarization. A key assumption in their model which we also follow is that input documents contain a variety of concepts, each of which are allocated a value, and the goal of a good summary is to maximize the sum of these values subject to the length constraint. The authors use bigrams as concepts and their frequency in the input documents as a proxy for their value. This model can also perform sentence compression (see also Gillick et al. (2009)), however, the deletion rules are hand-coded. Berg-Kirkpatrick et al. (2011) build on this work by recasting it as a structured prediction problem. They essentially combine the same bigram content scoring system with features relating to the parse tree

which they learn using a maximum-margin SVM trained on annotated gold-standard compressions.

Our multi-document summarization model jointly optimizes different aspects of the task involving both content selection and surface realization. Each individual aspect has its own dedicated expert, which we argue is advantageous as it renders inference simpler and affords flexibility (e.g., additional aspects can be incorporated into the model or trained separately on different datasets). Our work differs from Gillick et al. (2009) and Berg-Kirkpatrick et al. (2011) in three important respects. Firstly, we develop a genuinely abstractive model that is not limited to deletion. Our rewrite rules are encoded in quasi-synchronous tree substitution grammar and learned automatically from source documents and their summaries. Unlike previous applications of STSG to sentence compression (Cohn and Lapata, 2009; Cohn and Lapata, 2008) our quasi-synchronous TSG does not attempt to learn the complete translation from source to target sentence; it only loosely links the syntactic structure of the two (Smith and Eisner, 2006), and is therefore well suited to describing the relationship between documents and their abstracts. Secondly, our content selection component extends to features beyond the bigram horizon, as we learn to identify important concepts based on syntactic and positional information. We also learn which words are unlikely to appear in a summary. Thirdly, unlike Berg-Kirkpatrick et al. (2011) our model does not try to learn all the parameters (e.g., content, rewrite rules, style) of the summarization problem jointly; although decoupling learning from inference is perhaps less elegant from a modeling perspective, the learning process is more robust and reliable.

3 Modeling

There are many aspects to producing a good summary of multiple documents. The important content needs to be captured, typically key facts in each individual document, and information seen across the cluster. Stylistic features may be different in the summary from original documents. For instance, summaries tend to use more concise language, sources are not attributed as they are in news articles, and relative dates are not included. In addition, the summary must be fluent, coherent, and respect a pre-specified maximum length requirement.

We present an approach where elements of all the above considerations are learned from training data by separate dedicated components, and then combined in an integer linear programme. Content selection is performed partly through identifying the most salient topics (bigrams); an additional component learns to identify which information from the source documents should be in the summary based on positional information. Meanwhile, in terms of surface realization, a language model identifies the words that *should not* be in the output summaries, whereas a separate component learns to exclude sentences that are poor candidates for summaries. QTSG rules, learned from the training corpus, are used to generate alternative compressions and paraphrases of the source sentences, in the style suitable for the summaries. Finally, an ILP model combines the output of these components into a summary, jointly optimizing content selection and surface realization preferences, and providing the flexibility to treat some components as soft while others as hard constraints.

3.1 Document Representation

Given an input sentence, our approach deconstructs it into component phrases and clauses, typical of a phrase structure parser. In our experiments, we obtain this representation from the output of the Stanford parser (Klein and Manning, 2003) but any other broadly similar parser could be used instead. Nodes in the parse tree represent points where QTSG rules can be applied (and paraphrases generated), and they also represent decision points for the ILP. In the following, we will refer to these decision nodes as the set $\mathcal{N}$, and decisions for each node using the binary variable z_i, $i \in \mathcal{N}$.

3.2 Content Selection Using Bigrams

We follow Gillick et al. (2008) in modeling the information content of the summary as the weighted sum of the individual information units it contains. We represent information units as the set of bigrams $\mathcal{B}$ seen in the source documents. The weight w of each bigram is calculated from the number of source documents where the bigram was seen. The summary is thus given the score $f_{\mathcal{B}}(z)$, i.e., the weighted sum of

its information units:

$$f_{\mathcal{B}}(z) = \sum_{j \in \mathcal{B}} w_j b_j \quad (1)$$

where w_j is the weight of concept j, b_j a binary variable to indicate if concept j is present in the summary, and $j \in \mathcal{B}$.

Importantly, each information unit is counted only once; this encourages wide coverage of the source documents, and removes any drive towards redundant information without actively discouraging it, contrary to other global formulations where redundancy measures form part of the objective (McDonald, 2007). The counting mechanism is achieved by linking the variables z indicating nodes in the parse tree and b indicating bigrams:

$$b_j \leq \sum_{i \in \mathcal{N}:\, j \in \mathcal{B}_i} z_i \qquad \forall j \in \mathcal{B} \quad (2)$$

where $\mathcal{B}_i \subset \mathcal{B}$ is the subset of bigrams that are contained in node i. A drawback of the global nature of this counting mechanism, however, is that it cannot be integrated with local features such as those described below; our approach takes local features into account but these are weighted by other components.

3.3 Content Selection Using Salience

The bigram approach is a powerful method for identifying important concepts within the document cluster. It works particularly well in the sentence extraction paradigm. However, additional elements are known to be good predictors of important information. Examples include the position of a sentence in the document (e.g., first sentences often contain salient information), whether it contains proper nouns, numbers, pronouns, mentions of money, and so on. We decided to learn which of these elements (represented as nodes in the parse tree) are informative from training data. Specifically, sentences in the cluster documents were aligned to sentences from corresponding human summaries. Alignment was based rather simply on identifying the sentence pairs with the highest number of overlapping bigrams, without compensating for sentence length, or matching the sequence of information in the summaries and source documents (Nelken and Schieber, 2006). Matched sentences in the source documents were given positive labels, while unaligned sentences were given negative labels. These labels were then propagated to phrase structure nodes.

Weight	Feature
1.21	From first sentence in document
0.73	Contains proper nouns
0.68	Contains nouns
0.57	From first paragraph
0.53	From first three sentences
0.51	Contains numbers
-0.50	Contains pronouns
0.32	Contains money

Table 1: Weights and features of SVM that predicts the salience of summary content. Negative weights indicate information that should not be included in the summary.

We trained an SVM on this data (tree nodes and their labels) using surface features that do not overlap with bigram information: sentence and paragraph position, POS-tag information. Table 1 shows the most important features learned by the model as predictors of salient content.

The summary can be given a salience score $f_{\mathcal{S}}(z)$ using the raw SVM prediction scores of the individual parse tree nodes:

$$f_{\mathcal{S}}(z) = \sum_{i \in \mathcal{N}} (\Phi(i) \cdot \theta)\, z_i \quad (3)$$

where $\Phi(i)$ is the feature vector for node i, and θ the weights learned by the SVM.

3.4 Surface Realization Using Style

Some sentences in the source documents will make poor summary sentences, despite the information they contain, and therefore contrary to the predictions of the content selection indicators described above. This may be because the source sentence is very short, or is expressed as a quotation, or contains many pronouns that will not be resolved when the sentence is extracted.

Our idea is to learn which sentences are poor from a stylistic perspective using again aligned training data. We train a second SVM on the aligned sentences and their labels using surface features at the sentence level, such as sentence length and POS-tag information. The most important features learned by

Weight	Feature
-1.04	Word count less than 10
-0.83	Word count less than 20
-0.30	Question
-0.30	Quotation
-0.14	Personal pronouns

Table 2: Weights and features of SVM that predicts poor candidate sentences.

the model as predictors of poor sentences, and the weights assigned to them, are shown in Table 2.

The predictions of the SVM are incorporated into the ILP as a hard constraint, by forcing all parse tree nodes within those sentences predicted as poor (the set $\mathcal{N}^-$) to be zero:

$$z_i = 0 \quad \forall i \in \mathcal{N}^-. \tag{4}$$

3.5 Surface Realization Using Lexical Preferences

Human-written summaries differ from the source news articles in a number of ways. They delete extraneous information, merge material from several sentences, employ paraphrases and syntactic transformations, change the order of the source sentences and replace phrases or clauses with more general or specific descriptions. We could attempt to learn the "language of summaries" with a language model which we could then use to guide the generation process (e.g., by producing maximally probable output). Aside from the logistics of gathering training data large enough to provide robust estimates, we believe that a more compelling approach is to focus on the words that are *unlikely* to appear in the summary despite appearing in the source documents.

A comparison of the language models generated from the source documents and model summaries, even at the unigram level, is revealing. Table 3 shows lexemes that appear in both source and summary documents, but where the likelihood of the lexeme appearing in the summary is much less than that of it appearing the document, taking into account that the summary is much shorter anyway. The final column shows the $\log_{10}$-ratio ($L(w)$) between the two probabilities. We can see that least probable words are those that correspond to attributing information sources (e.g., *said, told, according to, spokesman*), dates described relatively (e.g., *last Monday*), and events that are in the process of happening (e.g., *coming, going*).

Lexeme w	Source count	Summary count	$L(w)$
say	5670	88	-1.63
go	638	11	-1.52
last	616	9	-1.69
get	543	15	-1.05
tell	512	8	-1.62
come	488	12	-1.17
know	404	9	-1.27
monday	391	8	-1.35
think	382	7	-1.46
next	239	7	-0.99
spokesman	197	4	-1.36

Table 3: Counts of lexemes in the source news articles and summaries, and measure of the ratio of their probabilities (for most common lexemes with ratio < -0.95).

As the amount of training data tends to be limited — there are usually only a few human-written summaries available per document cluster — we use a unigram language model, but conceivably a longer-range n-gram could be employed in the same vein. We incorporate preferences about summary language into the model as a soft constraint. The log-ratio values $f_{\mathcal{LR}}(z)$ are included in the objective and defined at the tree node level:

$$f_{\mathcal{LR}}(z) = \sum_{i \in \mathcal{N}} \sum_{w \in \mathcal{W}_i'} L(w) z_i \tag{5}$$

where $L(w)$, $w \in \mathcal{W}_i'$ is the log-ratio value for an individual word w:

$$L(w) = \log_{10} \frac{P_{\text{src}}(w)}{P_{\text{sum}}(w)},$$

$P_{\text{src}}(w)$ and $P_{\text{sum}}(w)$ are the probabilities of word w appearing in the source and summary documents respectively, and $\mathcal{W}_i'$ is the set of words at parse tree node i. Importantly, we include only those those lexemes with negative $L(w)$ values. This guides the model *away from* the kind of phrases described above, but not towards any particular language preferences.

3.6 Quasi-synchronous Tree Substitution Grammar

Rewrite rules involving substitutions, deletions and reorderings are captured in our model using a quasi-synchronous tree substitution grammar. Given an input (source) sentence S1 or its parse tree T1, the QTSG contains rules for generating possible translation trees T2. A grammar node in the target tree T2 is modeled on a subset of nodes in the source tree, with a rather loose alignment between the trees.

We extract QTSG rules from aligned source and summary sentence pairs represented by their phrase structure trees. Our algorithm builds up a list of leaf node alignments based on lexical identity. Direct parent nodes are aligned where more than one child node aligns. This quasi-synchronous "bottom-up" process gives us better ability to match non-isomorphic structures. We do not assume an alignment between source and target root nodes, nor do we require a surjective alignment of all target nodes to the source tree. QTSG rules are then created from aligned nodes above the leaf node level if all the nodes in the target tree can be explained using nodes from the source. Individual rewrite rules describe the mapping of source tree fragments into target tree fragments, and so the grammar represents the space of valid target trees that can be produced from a given source tree (Eisner, 2003; Cohn and Lapata, 2009).

Examples of the most frequent QTSG rules learned by the above process are shown in Figure 1. Many of the rules relate to the compression of noun phrases through deletion, and examples are shown in the upper box. Others capture the compression of verb phrases (middle box). An important rewrite operation is the abstraction of a sentence from a more complex source sentence, adding final punctuation if necessary (lower box).

At generation, paraphrases are created from source sentence parse trees by identifying and applying QTSG rules with matching structure. The transduction process starts at the root node of the parse tree, applying QTSG rules to sub-trees until leaf nodes are reached. Note that we do not use the Bayesian probability model normally associated with quasi-synchronous grammars (Smith and Eisner, 2006); instead, we ask the QTSG to provide

$\langle$NP, NP$\rangle \rightarrow \langle$[NP$_{\boxed{1}}$ PP], [NP$_{\boxed{1}}$]$\rangle$
$\langle$NP, NP$\rangle \rightarrow \langle$[NP$_{\boxed{1}}$ VP], [NP$_{\boxed{1}}$]$\rangle$
$\langle$NP, NP$\rangle \rightarrow \langle$[NP$_{\boxed{1}}$ SBAR], [NP$_{\boxed{1}}$]$\rangle$
$\langle$NP, NP$\rangle \rightarrow \langle$[NP$_{\boxed{1}}$, NP ,], [NP$_{\boxed{1}}$]$\rangle$
$\langle$NP, NP$\rangle \rightarrow \langle$[NP$_{\boxed{1}}$ CC NP], [NP$_{\boxed{1}}$]$\rangle$
$\langle$NP, NP$\rangle \rightarrow \langle$[NNP NNP$_{\boxed{1}}$], [NNP$_{\boxed{1}}$]$\rangle$
$\langle$NP, NP$\rangle \rightarrow \langle$[DT$_{\boxed{1}}$ JJ NN$_{\boxed{2}}$], [DT$_{\boxed{1}}$ NN$_{\boxed{2}}$]$\rangle$
$\langle$VP, VP$\rangle \rightarrow \langle$[VP$_{\boxed{1}}$ CC VP], [VP$_{\boxed{1}}$]$\rangle$
$\langle$VP, VP$\rangle \rightarrow \langle$[VP CC VP$_{\boxed{1}}$], [VP$_{\boxed{1}}$]$\rangle$
$\langle$VP, VP$\rangle \rightarrow \langle$[VP$_{\boxed{1}}$, CC VP], [VP$_{\boxed{1}}$]$\rangle$
$\langle$S, S$\rangle \rightarrow \langle$[NP$_{\boxed{1}}$ VP$_{\boxed{2}}$], [NP$_{\boxed{1}}$ VP$_{\boxed{2}}$.]$\rangle$
$\langle$S, S$\rangle \rightarrow \langle$[ADVP , NP$_{\boxed{1}}$ VP$_{\boxed{2}}$.], [NP$_{\boxed{1}}$ VP$_{\boxed{2}}$.]$\rangle$

Figure 1: Examples of most frequently learned QTSG rules. Boxed subscripts show aligned nodes.

paraphrases that are *acceptable* rather than *probable*, and generate all paraphrases licensed by the QTSG.

The alternative paraphrases are incorporated into the target phrase structure tree as choices that the ILP can make. We use the set $\mathcal{C} \subset \mathcal{N}$ to be the set of nodes where a choice of paraphrases is available, and $\mathcal{C}_i \subset \mathcal{N}, i \in \mathcal{C}$ to be the actual paraphrases of i. Where there are alternatives, it makes sense of course to select only one, which we implement using the constraint:

$$\sum_{j \in \mathcal{C}_i} z_j = z_i \qquad \forall i \in \mathcal{C},\, j \in \mathcal{C}_i \tag{6}$$

More generally, we need to constrain the output to ensure that a parse tree structure is maintained. For each node $i \in \mathcal{N}$, the set $\mathcal{D}_i \subset \mathcal{N}$ contains the list of dependent nodes (both ancestors and descendants) of node i, so that each set $\mathcal{D}_i$ contains the nodes that depend on the presence of i. We introduce a constraint to force node i to be present if any of its dependent nodes are chosen:

$$z_j \rightarrow z_i \qquad \forall i \in \mathcal{N},\, j \in \mathcal{D}_i \tag{7}$$

3.7 The ILP Objective

The model we propose for generating a multi-document summary is expressed as an integer linear programme and incorporates the content selection and surface realization preferences, as well as the

soft and hard constraints described in the preceding sections. The objective of the optimization problem is to maximize the score contributed by the various elements of content selection ($f_{\mathcal{B}}(z)$ and $f_{\mathcal{S}}(z)$) and soft surface realization constraints ($f_{\mathcal{LR}}(z)$) :

$$\max_{z} f_{\mathcal{B}}(z) + f_{\mathcal{S}}(z) + f_{\mathcal{LR}}(z) \quad (8)$$

This objective is subject to the constraints (2), (4), (6), and (7) that represent hard constraint decisions, or maintain the logical integrity of the model. An overall length constraint completes the model:

$$\sum_{i \in \mathcal{N}} l_i z_i \leq l_{\max} \quad (9)$$

where l_i is the number of words generated by choosing node i, and $l_{\max}$ is the global word length limit.

Note that the scores in the objective are for each tree node and not each sentence. This affords the model flexibility: the content selection elements are generally not competing with each other to give a decision on a sentence (see McDonald (2007)). Instead, components are marking positive and negative nodes. The ILP is implicitly searching the grammar rules for ways to rewrite the sentence, with the aim of including the salient nodes while removing negative-scoring nodes (deleting them increases the score of the node to zero). Figure 2 shows an example of a source sentence where the bigram, salience and language preference components of the ILP work together to score nodes in the parse tree. The nodes $\text{NP}_{\boxed{1}}$, $\text{VP}_{\boxed{3}}$ and $\text{VP}_{\boxed{4}}$ all have positive scores, while "said Tuesday" is negative. As a rewrite possibility, the rewrite rule shown bottom left is available, which will remove the negative node. Further rewrite rules allow $\text{VP}_{\boxed{2}}$ to be compressed. The output actually generated by the model used sub-trees (b) and (d) — the final text is included in Table 6.

4 Experimental Set-up

Data Our model was evaluated on the TAC non-update multi-document summarization task which involves generating a 100-word-limited summary from a cluster of 10 related input documents; additionally, TAC provides a set of four model summaries for each cluster, written by human experts. We used the 44 document clusters from TAC-2009 as training data, to learn the different elements of the model. The 48 document clusters of TAC-2008 were reserved for the generation of test summaries.[3]

Training The two components described in Sections 3.3 and 3.4 were trained using binary SVM classifiers, with labels inferred automatically via alignment. The salience classifier was trained on 102,754 node instances (16,042 positive and 86,712 negative). The style classifier was trained on 20,443 sentence instances (2,083 positive and 18,360 negative). We learned the feature weights with a linear SVM, using the software SVM-OOPS (Woodsend and Gondzio, 2009). Because of the high compression rate in this task, sentence alignment leads to an unbalanced data set. We compensated for this by using different SVM hyper-parameters C^+ and C^- as the loss multiplier for misclassification of positive and negative training samples respectively. SVM hyper-parameters were chosen that gave the highest F1 values using 10-fold cross-validation. The salience SVM obtained a precision of 0.28 and recall of 0.43. Precision for the style SVM was 0.20 and recall 0.63, respectively. The classifiers on their own would thus not be great predictors of salience or style, but in practice they were useful for breaking ties in bigram scores.

Aligned sentences from the training data were also used to learn the quasi-synchronous tree substitution grammar, using the process described in Section 3.6. Rules seen fewer than 3 times were removed, resulting in a total of 339 QTSG rules. Two unigram language models (see Section 3.5) were trained on the source articles and summaries, respectively. Their probabilities were compared to give the word list shown in Table 3. We removed words with a source count less than 50, providing a list of 60 lexemes. The resulting integer linear programmes were solved using SCIP,[4] and it took 55 seconds on average to read in and solve a document cluster problem.

Evaluation We compared our model against two systems. As a baseline, we used the ICSI-1 extractive system (Gillick et al., 2008) which is also based on ILP and was highly ranked in the TAC-2008 evaluation. We also compared against the "learned phrase compression" system of Berg-Kirkpatrick et

[3] This split follows Berg-Kirkpatrick et al. (2011).

[4] http://scip.zib.de/

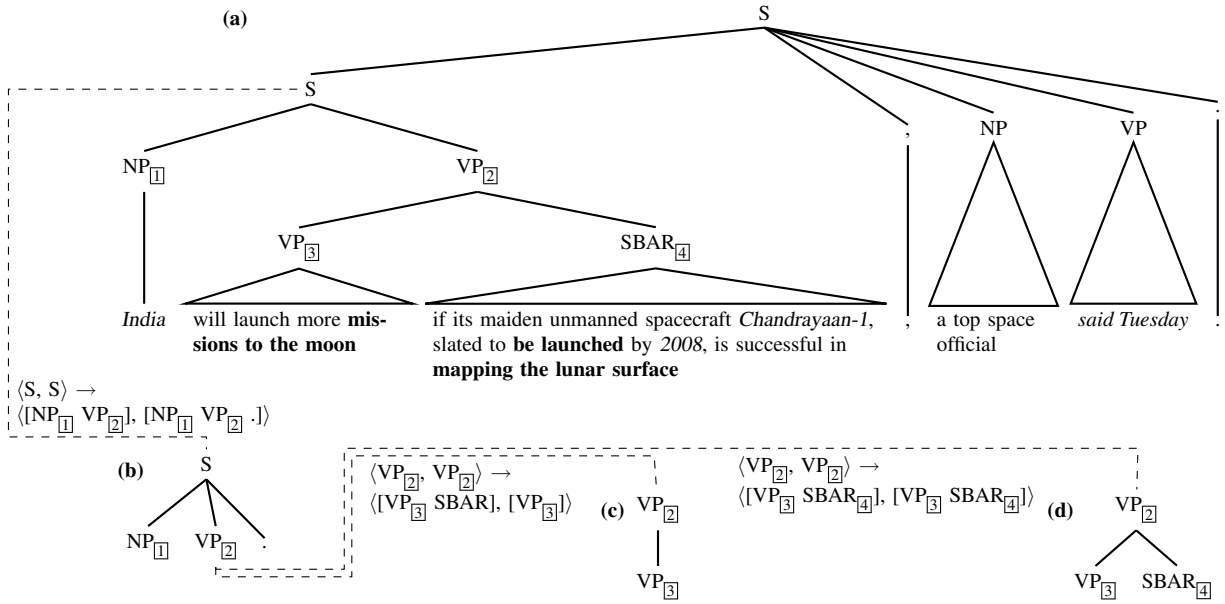

Figure 2: Sentence representation provided to the ILP. (a) The source sentence representation (child nodes condensed for space reasons). **Bigrams** are shown in bold, slanted text indicates *phrases with high salience scores* f_S, while *said Tuesday* is penalized by $f_{\mathcal{LR}}$. Alternative sub-trees (b), (c) and (d) are created using QTSG rules (dashed lines). The output sentence (see Table 6) was generated from sub-trees (b) and (d).

al. (2011) (henceforth B-K), which has the highest reported ROUGE scores that we are aware of.[5] In addition to the full model described in Section 3, we also produced outputs where each of the five components described in Sections 3.2–3.6 were removed, to assess their individual contribution.

We evaluated the output summaries in two ways, using automatic measures and human judgements. Automatic evaluation was performed with ROUGE (Lin and Hovy, 2003) using TAC-2008 parameter settings. We report bigram overlap (ROUGE-2) and skip-bigram (ROUGE-SU4) recall values. We also used Translation Edit Rate (TER, Snover et al. (2006)) to examine the systems' rewrite potential. TER is defined as the minimum number of edits (insertions, deletions, substitutions, and shifts) required to change the system output so that it exactly matches a reference (here, the reference is the most closely aligning source sentence). The perfect TER score is 0, however note that it can be higher than 1 due to insertions.

Our judgement elicitation study was conducted as follows. We randomly selected ten document clusters from the test set and generated summaries with our model (and its lesser variations). We also included the corresponding ICSI-1 and B-K summaries, and one randomly-selected model summary. The study was conducted over the Internet using Mechanical Turk and was completed by 54 volunteers, all self reported native English speakers. Participants were first asked to read the documents in each cluster. Next, they were asked a few comprehension questions to ensure they had understood and processed the documents. Finally, they were presented with a summary and asked to rate it along two dimensions: grammaticality (is the summary fluent and grammatical?), and informativeness (are the main topics captured in the summary?). The subjects used a 1–5 rating scale, with half-points allowed. Participants who declared themselves as non-native English speakers, did not answer the comprehension questions correctly or took only a few minutes to complete the task were eliminated.

5 Results

Our results are summarized in Table 4. Let us first discuss those obtained using ROUGE-2 (2-R) and ROUGE-SU4 (SU4-R) recall values. As can be seen

[5] We are grateful to Taylor Berg-Kirkpatrick for making his system output available to us.

Models	ROUGE		TER (%)				Sentences		
	2-R	SU4-R	Ins	Del	Sub	Shift	Count	CR (%)	Mod (%)
ICSI-1	11.03	13.96	—	—	—	—	200	—	—
B-K	**11.71**	**14.47**	0.2	26.2	2.3	0.4	216	74.0	63.9
MA-ILP	11.37	**14.47**	0.7	11.6	5.3	0.6	191	89.1	61.8
ILP w/o bigrams	9.24	12.66	0.8	15.4	11.8	1.2	205	85.4	80.0
ILP w/o salience	11.38	14.71	1.1	19.1	12.0	1.3	233	82.1	92.3
ILP w/o style	**11.83**	**15.09**	1.4	17.4	18.9	1.7	271	84.1	86.3
ILP w/o log-ratio	11.41	14.70	1.2	16.9	12.5	1.5	223	84.3	90.1
ILP w/o QTSG	10.32	13.68	0	0	0	0	163	100.0	0

Table 4: Performance of the multiple-aspect ILP model against comparison systems using ROUGE and the four components of TER (insertion, deletion, substitution, shifts). In the lower section, performance of our model without (w/o) each component in turn. The final columns show the number of source sentences, the average compression ratio, and the proportion of sentences modified.

from the upper section of Table 4, the systems incorporating some form of rewriting gain slightly higher ROUGE scores than ICSI-1. The multiple aspects ILP system (MA-ILP) yields ROUGE scores similar to B-K, despite performing rewriting operations which increase the scope for error and without requiring any hand-crafted compression rules or manually annotated training data. Indeed, the outputs of the two systems are not significantly different under ROUGE (using a paired t-test, $p > 0.5$).

In the lower section of Table 4, we show the performance of our model when each of the contributing components described in Section 3 are removed. Clearly the bigram content indicators are an important element for the ROUGE scores, as their removal yields a reduction of 2.46 points (see the row ILP w/o bigrams in Table 4). The model without QTSG rules (ILP w/o QTSG) is effectively limited to sentence extraction, and removing rewrite rules also lowers ROUGE scores to levels similar to ICSI-1. ROUGE scores are increased by allowing the model to select "poor quality" sentences (ILP w/o style), higher indeed than those of the B-K system. The inclusion of non-summary language (ILP w/o log-ratio) does not affect ROUGE scores to the same extent that bigrams and QTSG do.

Table 4 includes a break-down of the systems' rewrite operations as measured by TER. We also show the number of source sentences (Count), the average compression ratio (CR %) and the proportion of sentences modified (Mod %) by each system. As can be seen, MA-ILP draws on fewer sentences, performs less deletion and more rewriting than B-K. The number of deletions increases when individual ILP components are removed and so does the number of substitutions. All the subsystems are more aggressive in their rewriting than when used in combination (higher TER, higher compression rate and a larger number of sentences are modified). Expectedly, when removing the QTSG rules, the ILP is limited to a pure extractive system (last row in Table 4).

Models	Grammar	Inform
ICSI-1	**4.68**	2.55
B-K	4.40	2.70
MA-ILP	**4.68**	**3.90**
ILP w/o style	3.30	2.67
Gold	4.90	4.75

Table 5: Mean ratings on system output output.

The results of our human evaluation study are shown in Table 5. We elicited grammaticality and informativeness ratings for a randomly selected model summary, ICSI-1, B-K, the multiple aspect ILP (MA-ILP), and the ILP w/o style which we included in this study as it performed best under ROUGE. ICSI-1, B-K, and MA-ILP are rated highly on the grammaticality dimension. MA-ILP is indistinguishable from the sentence extraction system (ICSI-1). Both systems are significantly more grammatical than B-K ($\alpha < 0.05$, using a Post-hoc Tukey test). Notice that summaries created by the ILP w/o style are rated poorly by humans, contrary to ROUGE. The style component stops very short

Florida's Governor Jeb Bush asked the US Supreme Court to intervene to keep a comatose woman alive, over the wishes of her husband, who wants to disconnect the feeding tube that has sustained her for 14 years. Her husband, Michael Schiavo, and her parents, Robert and Mary Schindler, have conflicts of interest that prevent them from fairly deciding whether to keep her alive. Some doctors have testified that Terri Schiavo is in a persistent vegetative state with no hope for recovery. The state House in Florida passed a bill Thursday to extend life support for a brain-damaged woman.
The space agencies of India and France signed an agreement to cooperate in launching a satellite in four years that will help make climate predictions more accurate. The Indian Space Research Organization (ISRO) has short-listed experiments from five nations including the United States, Britain and Germany, for a slot on India's unmanned moon mission Chandrayaan-1 to be undertaken by 2006-2007, the Press Trust of India (PTI) reported Monday. India will launch more missions to the moon if its maiden unmanned spacecraft Chandrayaan-1, slated to be launched by 2008, is successful in mapping the lunar surface.

Table 6: Example summaries generated by the multiple aspects model (MA-ILP).

sentences and quotations from being included in the summary even if they have quite high bigram or content scores. Without it, the model tends to generate summaries that are fragmentary and lacking proper context, resulting in lower grammaticality (and informativeness) when judged by humans. The MA-ILP system obtains the highest rating with respect to information content. It is significantly better ($\alpha < 0.05$) than ICSI-1 and B-K. This is not entirely surprising as our model includes additional content selection elements over and above the bigram units. There is still a significant gap from all systems to the gold-standard human-authored summaries. Example output summaries of the full ILP model are shown in Table 6.

Overall, we obtain best results when considering the contributions from the individual model experts collectively. This suggests that additional improvements could be obtained with more experts. It is also possible that optimizing the relative weightings of experts in the ILP objective would improve output. The TER analysis shows that the experts have a tempering effect on each other, resulting in less aggressive, but qualitatively better, rewriting than when used individually. Generally, experts work together to shape an output sentence, but they can also compete. In the future, we also plan to test the ability of the model to adapt to other multi-document summarization tasks, where the location of summary information is not as regular as it is in news articles. We would also like interface our model with sentence ordering and more generally with some notion of the coherence of the generated summary.

Acknowledgments We are grateful to Micha Elsner for his input on earlier versions of this work. We would also like to thank members of the ILCC at the School of Informatics for valuable discussions and comments. We acknowledge the support of EPSRC through project grants EP/I032916/1 and EP/I017127/1.

References

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 481–490, Portland, Oregon.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:273–381.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 137–144, Manchester, UK.

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

Pawan Deshpande, Regina Barzilay, and David Karger. 2007. Randomized decoding for selection-and-ordering problems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics;*

Proceedings of the Main Conference, pages 444–451, Rochester, New York.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the ACL Interactive Poster/Demonstration Sessions*, pages 205–208, Sapporo, Japan.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 397–403, Geneva, Switzerland.

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*, pages 10–18, Boulder, Colorado.

Dan Gillick, Benoit Favre, and Dilek Hakkani-tür. 2008. The ICSI summarization system at TAC 2008. In *Proceedings of the Text Analysis Conference*.

Dan Gillick, Benoit Favre, Dilek Hakkani-tür, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/UTD summarization system at TAC 2009. In *Proceedings of the Text Analysis Conference*.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL–ANLP Workshop on Automatic Summarization*, pages 40–48, Seattle, Washington.

Dorit S. Hochba. 1997. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. In Dorit S. Hochba, editor, *Approximation Algorithms for NP-Hard Problems*, pages 94–143. PWS Publishing Company, Boston, MA.

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

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

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

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

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

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European conference on IR Research*, pages 557–564, Rome, Italy.

Rani Nelken and Stuart Schieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 161–168, Trento, Italy.

David Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of Workshop on Statistical Machine Translation*, pages 23–30, NYC.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge.

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

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

Kristian Woodsend and Mirella Lapata. 2010. Automatic generation of story highlights. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 565–574, Uppsala, Sweden.

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Title generation with quasi-synchronous grammar. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 513–523, Cambridge, MA.