Extractive Multi-Document Summarization with Integer Linear Programming and Support Vector Regression

Dimitrios Galanis, Gerasimos Lampouras and Ion Androutsopoulos
Department of Informatics
Athens University of Economics and Business
Patission 76, GR-104 34 Athens, Greece
galanisd@aueb.gr, lampouras06@aueb.gr, ion@aueb.gr

ABSTRACT

We present a new method to generate extractive multi-document summaries. The method uses Integer Linear Programming to jointly maximize the importance of the sentences it includes in the summary and their diversity, without exceeding a maximum allowed summary length. To obtain an importance score for each sentence, it uses a Support Vector Regression model trained on human-authored summaries, whereas the diversity of the selected sentences is measured as the number of distinct word bigrams in the resulting summary. Experimental results on widely used benchmarks show that our method achieves state of the art results, when compared to competitive extractive summarizers, while being computationally efficient as well.

KEYWORDS: Text Summarization; Integer Linear Programming; Support Vector Regression.

Proceedings of COLING 2012: Technical Papers, pages 911–926,
COLING 2012, Mumbai, December 2012.

1 Introduction

A multi-document summarization system aims to generate a single summary from an input set of documents. The input documents may have been obtained, for example, by submitting a query to an information retrieval engine and retaining the most highly ranked documents, or by clustering the documents of a large collection and then using each cluster as a set of documents to be summarized. Although evaluations with human judges also examine the coherence, referential clarity, grammaticality, and readability of the summaries (Dang, 2005, 2006; Dang and Owczarzak, 2008), and some of these factors have also been considered in recent summarization algorithms (Nishikawa et al., 2010b; Woodsend and Lapata, 2012), most current multi-document summarization systems consider only the importance of the summary's sentences, their non-redundancy (also called diversity), and the summary length (McDonald, 2007; Berg-Kirkpatrick et al., 2011; Lin and Bilmes, 2011).

An *extractive* multi-document summarizer forms summaries by extracting (selecting) sentences from the input documents, without modifying the selected sentences. By contrast, an *abstractive* summarizer may also shorten or, more generally, rephrase the selected sentences. In practice, the additional processing of the selected sentences may only marginally improve or even reduce the perceived quality of the resulting summaries (Gillick and Favre, 2009), though recent work has produced abstractive summarization methods that perform better than extractive ones (Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012). Nevertheless, the difference in the performance of extractive and abstractive summarizers is often small, and abstractive summarizers typically require more processing time, as well as tools and resources (e.g., reliable large coverage parsers, paraphrasing rules) that are often not available in less widely spoken languages. Hence, it is still worth trying to improve extractive summarizers, at least from a practical, application-oriented point of view.

Many multi-document summarizers, especially extractive ones, adopt a greedy search when constructing summaries. For example, they may rank the sentences of the input documents by importance, and then iteratively add to the summary (and remove from the ranked list of input sentences) the sentence with the highest importance score, until the maximum allowed summary length has been reached, possibly discarding sentences that are too similar to sentences already included in the summary. Recent work has shown that adopting more principled optimization methods based on Integer Linear Programming (ILP), instead of greedy search, can lead to summaries that are better or at least comparable to those of state of the art summarizers (McDonald, 2007; Gillick and Favre, 2009; Nishikawa et al., 2010a).

In this paper, we introduce a new extractive multi-document summarization method that uses ILP to jointly optimize the importance of the summary's sentences and their diversity (non-redundancy), also respecting the maximum allowed summary length. Our method is more efficient than the seminal ILP-based summarizer of McDonald (2007), because of its simpler ILP model. The main competitor of our method, if we exclude abstractive summarizers, is the extractive version of Berg-Kirkpatrick et al.'s (2011) summarizer, which has the best previously reported results in extractive multi-document summarization. Inspired by Berg-Kirkpatrick et al.'s work, we include in the objective function of our ILP model the number of distinct word bigrams (of the input documents) that occur in the summary, but we use that number to measure diversity, unlike Berg-Kirkpatrick et al.'s work, where bigrams are weighted to measure importance. To obtain an importance score for each sentence, we use a Support Vector Regression (SVR) model (Vapnik, 1998), which has no direct counter-part in Berg-Kirkpatrick et al.'s method. We show that our ILP method achieves state of the art ROUGE scores (Lin, 2004) on widely used benchmark datasets, when compared to Berg-Kirkpatrick's and other competitive extractive summarizers, also outperforming two greedy baselines that use only the importance scores of the SVR. For completeness, we also discuss and compare against the abstractive version of Berg-Kirkpatrick et al.'s summarizer, and the state of the art abstractive summarizer of Woodsend and Lapata (2012).

Section 2 below discusses previous work on ILP methods for summarization. Section 3 presents our own ILP model, after first introducing the SVR model of sentence importance and the greedy baselines. Section 4 presents the experiments that we conducted and discusses their results. Section 5 concludes and proposes directions for further research.

2 Related work

The first ILP method for summarization was proposed by McDonald (2007). It constructs summaries by maximizing the importance of the selected sentences and minimizing their pairwise similarity, as shown below. No sentence ordering is performed.

$$\max_{x,y} \sum_{i=1}^{n} imp(s_i) \cdot x_i - \sum_{i=1}^{n} \sum_{j=i+1}^{n} sim(s_i, s_j) \cdot y_{i,j} \tag{1}$$

subject to:

$$\sum_{i=1}^{n} l_i \cdot x_i \leq L_{max} \tag{2}$$

and (for $i = 1, \ldots, n$ and $j = i + 1, \ldots, n$):

$$\begin{aligned} y_{i,j} - x_i &\leq 0 \qquad (3) \\ y_{i,j} - x_j &\leq 0 \qquad (4) \\ y_i + x_j - y_{i,j} &\leq 1 \qquad (5) \end{aligned}$$

Here, n is the number of sentences in the input documents; $imp(s_i)$ is the importance score of sentence s_i; l_i is the length of s_i; $sim(s_i, s_j)$ is the similarity of sentences s_i and s_j; and L_{max} is the maximum allowed length. The x_i variables, jointly denoted x, are binary and indicate whether or not the corresponding sentences s_i are included (selected) in the summary. The $y_{i,j}$ variables, jointly denoted y, are also binary and indicate whether or not both s_i and s_j are included in the summary. Constraint 2 ensures that the maximum total length is not exceeded. Constraints 3–5 ensure that the values of x_i, x_j, and $y_{i,j}$ are consistent (e.g., if $y_{i,j} = 1$, then $x_i = x_j = 1$; and if $y_{i,j} = 0$, then $x_i = 0$ or $x_j = 0$).

McDonald showed experimentally that the ILP model above achieves better ROUGE scores (Lin, 2004) than a greedy method that attempts to maximize the same objective (1). However, McDonald also showed that the ILP model above corresponds to an NP-hard problem and is, therefore, intractable for a large number of sentences. A set of experiments by McDonald confirmed that the model does not scale up well in practice, mostly because of the $O(n^2)$ $y_{i,j}$ variables that are used to model the redundancy between sentences. Furthermore, the ROUGE scores of McDonald's ILP model were not always better than those obtained using a modified version of the Knapsack dynamic programming algorithm (Cormen et al., 2001).

In a more recent approach, Berg-Kirkpatrick et al. (2011) presented an ILP method based on the notion of 'concepts', a notion initially introduced by Gillick and Favre (2009). The so called 'concepts' are actually word bigrams, all the word bigrams of the documents to be summarized. Each bigram has a weight w_i that indicates its importance. The ILP objective (6) of Berg-Kirkpatrick et al. prefers summaries with many important concepts, i.e., summaries whose bigrams have a large sum of weights w_i; below b_i are binary variables indicating which bigrams ($|B|$ in total) are present in the summary. An additional constraint, not shown here, ensures that the maximum allowed summary length is not exceeded.

$$\max_{b,c} h(b,c) = \max_{b,c} \sum_{i=1}^{|B|} w_i \cdot b_i + \sum_{i=1}^{|C|} u_i \cdot c_i = \sum_{i=1}^{|B|} (W^T \cdot \Phi_i) \cdot b_i + \sum_{i=1}^{|C|} (U^T \cdot \Psi_i) \cdot c_i \qquad (6)$$

Berg-Kirkpatrick et al.'s model also takes into account the possible subtree cuts (deletions) of the parse trees of the sentences of the input documents. The cuts give rise to different compressions (shortenings) of the sentences; hence, Berg-Kirkpatrick et al.'s summarizer is an abstractive one. In the objective (6), u_i are the weights of the possible subtree cuts of all the sentences of the input documents, and c_i are binary variables indicating which cuts ($|C|$ in total) are used. Additional contraints, not shown above, ensure that the values of b_i and c_i are consistent. Overall, Berg-Kirkpatrick et al.'s method aims to produce summaries that contain many important bigrams, while also performing many desirable subtree cuts.

The weights w_i and u_i are themselves estimated as weighted sums of features, i.e., $w_i = W^T \cdot \Phi_i$, where Φ_i is a feature vector describing the bigram of the binary variable b_i, and W is a vector of feature weights; similarly, $u_i = U^T \cdot \Psi_i$, where Ψ_i is a feature vector describing the subtree cut of the binary variable c_i, and U is a vector of feature weights.[1] The feature vector Φ_i includes, for example, the frequency of the corresponding bigram in the documents to be summarized, and the minimum sentence position (e.g., 3rd sentence in a document) of the sentences that contain that bigram in the input documents. The features of Ψ_i show, for example, if a relative clause or a temporal phrase was cut.

Berg-Kirkpatrick et al. use a structured Support Vector Machine (SVM) (Vapnik, 1998; Tsochantaridis et al., 2004) that assigns to each candidate summary the score $h(b,c)$ of the objective function (6), given W and U. During training, the SVM searches for the values of W and U that allow it to prefer the gold summary (of each training set of input documents) to all the other possible summaries (of the same input documents) by a margin determined by a loss function. Berg-Kirkpatrick et al. use a bigram recall loss function similar to ROUGE-2 (Lin, 2004). The loss function causes more emphasis (larger margin) to be placed on preferring the gold summaries to other summaries that share many bigrams with the gold ones. The learnt W and U are then used in the objective (6).

Berg-Kirkpatrick et al. report that their full method achieves higher ROUGE scores than an extractive version of their method (without the subtree cuts, i.e., without sentence compression) with no significant decrease in grammaticality (when sentence compression is used), unlike other work (Gillick and Favre, 2009), where sentence compression was found to reduce grammaticality. The extractive version of Berg-Kirkpatrick et al.'s method omits the second term of Formula (6), as in the previous work of Gillick and Favre (2009). As already noted, the extractive version of Berg-Kirkpatrick et al.'s method has the best previously published results in extractive multi-document summarization.

[1] Berg-Kirkpatrick et al. (2011) use different terminology. A minor difference from their description of their method is that they seem to set $W = U$, but in a more general formulation this does not seem to be necessary.

More recently, Woodsend and Lapata (2012) proposed an ILP-based method that forms a summary by maximizing the objective function shown below. The objective function combines the importance $f_B(z)$ of the bigrams in the summary's sentences, the salience $f_S(z)$ of the parse tree nodes of the summary's sentences, and a unigram language model $f_{LR}(z)$, which penalizes sentences containing words that are unlikely to appear in summaries; we do not discuss $f_{LR}(z)$ further to save space.

$$max_z f_B(z) + f_S(z) + f_{LR}(z) \tag{7}$$

The argument z collectively denotes binary variables z_i, one z_i for each node of the parse tree of each sentence of the input documents. Each z_i shows whether or not the corresponding node has been retained or deleted. By deleting nodes, the method can compress sentences; hence, this is also an abstractive summarization method. The $f_B(z)$ component is the same as in Berg-Kirkpatrick et al.'s work ($f_B(z) = \sum_{i=1}^{|B|} w_i \cdot b_i$). Additional constraints ensure that a bigram can be selected only if at least a parse tree node that subsumes it has been selected, that the maximum summary length is not exceeded etc.

To compute $f_S(z)$, Woodsend and Lapata train a linear SVM, with separating hyperplane $W^T \cdot \Phi_i = 0$, to predict whether or not a sentence s_i of an input set of documents would be selected by a human creating a summary. The $f_S(z)$ score, defined below, is the sum of the trained SVM's predictions, for all the phrases that correspond to the retained parse tree nodes of the input sentences; Φ_i is a feature vector describing each retained phrase, with features indicating, for example, if the phrase was obtained from the first sentence of an input document, if it contains pronouns etc.; W are the feature weights learnt by the SVM.

$$f_S(z) = \sum_i (W^T \cdot \Phi_i) \cdot z_i \tag{8}$$

A second SVM is trained to exclude sentences that are too long, contain quotations etc. The predictions of the second SVM are used to add hard constraints to the ILP model, rather than including the predictions in the ILP objective function (7).

The method of Woodsend and Lapata optionally employs a quasi-synchronous tree grammar (QSTG), to generate candidate compressions and paraphrases of the source sentences. The QSTG grammar is learnt from aligned summary and source sentences. When the grammar is used, the rest of the method does not operate only on the sentences of the input documents, but also on rephrasings of these sentences, produced by the grammar. Hence, in its full form, the method of Woodsend and Lapata is abstractive not only because it can delete tree nodes of the parse trees, but also because it can rephrase sentences using the grammar. It can be turned into an extractive summarization method by disabling the QSTG grammar and disallowing tree node deletions. Woodsend and Lapata provide experimental results of their method without the QSTG grammar, but not without tree node deletions; hence, we could not compare directly to a purely extractive version of Woodsend and Lapata'a summarizer.

Lin and Bilmes (2011) construct summaries by maximizing a monotone submodular function. This is an NP-hard problem; however, there is a greedy algorithm that approximates the optimum by a constant factor. Lin and Bilmes show that several previous summarization approaches can be described in terms of submodular functions. They also propose their own submodular functions for summarization, which combine importance and diversity.

3 Our method

In this section, we first discuss our SVR model that assigns importance scores to the sentences of the input documents, and two greedy baseline summarizers that use the SVR without ILP. We then introduce our ILP method, which jointly maximizes the importance and diversity of the selected sentences, while respecting the maximum allowed summary length.

3.1 The SVR model of sentence importance

A Support Vector Regression (SVR) model aims to learn a function $f : \mathbb{R}^n \mapsto \mathbb{R}$, which will be used to predict the value of a variable $y \in \mathbb{R}$ given a feature vector $X \in \mathbb{R}^n$. In particular, given l training instances $(X_1, y_1), \ldots, (X_l, y_l)$, an SVR model is learnt by solving the following optimization problem (Vapnik, 1998); W is a vector of feature weights; ϕ is a function that maps feature vectors to a new vector space of higher dimensionality to allow non-linear functions to be learnt in the original space; $C > 0$ and $\epsilon > 0$ are given.

$$\min_{W,b,\xi,\xi^*} \frac{1}{2} \|W\|^2 + C \sum_{i=1}^{l} \xi_i + C \sum_{i=1}^{l} \xi_i^* \tag{9}$$

subject to (for $i = 1, \ldots, l$):

$$\begin{aligned} W^T \cdot \phi(X_i) + w_0 - y_i &\leq \epsilon + \xi_i && (10)\\ y_i - W^T \cdot \phi(X_i) - w_0 &\leq \epsilon + \xi_i^* && (11)\\ \xi_i &\geq 0 && (12)\\ \xi_i^* &\geq 0 && (13) \end{aligned}$$

The goal is to learn a linear (in the new space) function, whose prediction (value) $W^T \cdot \phi(X_i) + w_0$ for each training instance X_i will not to be farther than ϵ from the target (correct) value y_i. Since this is not always feasible, two slack variables ξ_i and ξ_i^* are used to measure the prediction's error above or below the target y_i. The objective (9) jointly minimizes the total prediction error and $\|W\|$, to avoid overfitting.[2]

In our case, X_i is the feature vector of a sentence and y_i is the sentence's importance score. During training, the target score y_i of each sentence s, i.e., the score that the SVR should ideally return, is taken to be the average of the ROUGE-2 and ROUGE-SU4 scores (Lin, 2004) of s, comparing s against the corresponding gold (human-written) summaries; the latter are included in the training datasets that we used. We took the average of ROUGE-2 and ROUGE-SU4, because they are the two most commonly used measures to automatically evaluate machine-generated summaries against gold ones. Roughly speaking, both measures compute the bigram recall of a summary (or individual sentence) being evaluated against multiple gold summaries (provided by different human authors), but ROUGE-SU4 also considers skip bigrams with a maximum distance of 4 words between the words of each skip bigram. Both measures have been found to correlate well with human judgements in extractive summarization (Lin, 2004). Hence, training the SVR to predict the average ROUGE-2 and ROUGE-SU4 of each sentence can be particularly useful. Intuitively, a sentence with a high ROUGE score has a high overlap with the gold summaries; and since the gold summaries

[2] We use the SVR implementation of LIBSVM (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`) with an RBF (non-linear) kernel and LIBSVM's parameter tuning facilities.

contain the sentences that human authors considered most important, a sentence with a high ROUGE score is most likely also important. This is why we view our SVR, which attempts to predict the ROUGE score (average ROUGE-2 and ROUGE-SU4) of each sentence, as a component that assigns an importance score to each sentence.

The idea to use ROUGE during training is also present in the work of Berg-Kirkpatrick et al. (Section 2). The SVM that Berg-Kirkpatrick et al. use, however, in effect attempts to separate (prefer) the gold summaries from the other possible summaries; ROUGE (more precisely, a modified version of ROUGE-2) is included in the SVM as a loss function to force the SVM to place more emphasis on separating gold summaries from other possible summaries with high ROUGE scores. By contrast, the SVR that we use attempts to directly output the ROUGE score of each sentence. Furthermore, the RBF kernel that we use in the SVR allows the SVR to learn non-linear functions, whereas the linear SVM of Berg-Kirkpatrick et al. can learn only linear functions. We also note that the two SVMs used by Woodsend and Lapata (Section 2) perform binary classification (not regression), attempting to separate sentences that a human would include in a summary from sentences that would not be included. The (unsigned) distance from the learnt separating hyperplane of the first SVM is included in the objective function of the ILP model, in effect treating the distance as a confidence score. We believe that our use of a regression model (SVR) is a better choice, because the distance from an SVM's separating hyperplane is often a poor confidence estimate. We also note that the second SVM of Woodsend and Lapata contributes only hard constraints to the ILP model, without taking into account the SVM's confidence.

We include the following features in the feature vector X of each sentence s:

- Sentence position $SP(s)$:
$$SP(s) = \frac{pos(s, d(s))}{|d(s)|}$$
where $pos(s, d(s))$ is the position (sentence order) of sentence s in its document $d(s)$, and $|d(s)|$ is the number of sentences in $d(s)$.

- Named entities $NE(s)$:
$$NE(s) = \frac{n(s)}{len(s)}$$
$n(s)$ is the number of named entities in s, and $len(s)$ is the number of words in s.[3]

- Levenshtein distance $LD(s, q)$: The Levenshtein Distance (Levenshtein, 1966) between the user's query q and sentence s; insertions, deletions, and replacements affect entire words. In the datasets we experimented with, the documents to be summarized were relevant to a query q, which was always available.

- Word overlap $WO(s, q)$: The number of words shared by the query q and sentence s, after removing stop words and duplicate words from both q and s.

- Content word frequency $CF(s)$ and document frequency $DF(s)$: We use these measures as defined by Schilder and Ravikumar (2008). $CF(s)$ is defined as follows:
$$CF(s) = \frac{\sum_{i=1}^{c_s} p_c(w_i)}{c_s}$$

[3] We use Stanford University's named entity recognizer (consult `http://nlp.stanford.edu/`).

where c_s is the number of content words in sentence s, $p_c(w) = \frac{m}{M}$, m is the number of occurrences of content word w in the input documents, and M is the total number of content word occurences in the input documents. Similarly, $DF(s)$ is defined as:

$$DF(s) = \frac{\sum_{i=1}^{c_s} p_d(w_i)}{c_s}$$

where $p_d(w) = \frac{d}{D}$, d is the number of input documents the content word w occurs in, and D is the number of all input documents.

Experiments on the development set (see below) confirmed that all the features have a positive impact, i.e., the results are worse, if any of the features are removed.

3.2 The baseline summarizers

We compare against two greedy baselines that use the SVR model of sentence importance, but not ILP. The first one, called GREEDY, uses the trained SVR model of the previous section to assign importance scores to all the sentences of the documents to be summarized. It then ranks the sentences by decreasing importance score and constructs the summary by iteratively selecting (and removing from the ranked list of sentences) the sentence with the highest importance score that fits in the summary space left.

The second baseline, called GREEDY-RED, operates in the same way, but it also takes into account redundancy. When a new sentence (with the highest importance score among the remaining sentences in the ranked list) is about to be added to the summary, its cosine similarity (computed on words) to all the sentences that have already been included in the summary is computed. If the similarity between the new and any of the already selected sentences exceeds a threshold t, the new sentence is discarded and a new iteration starts, where the next sentence of the ranked list of remaining sentences is considered. In our experiments, t was determined by tuning GREEDY-RED on development data (see below).

3.3 Our ILP summarization model

Instead of directly using the importance score $f_{SVR}(s_i)$ of each sentence s_i, as returned by the SVR model of Section 3.1, we normalize it using the maximum and mimimum values that the SVR model returns for the $j = 1, \ldots, n$ sentences of the input documents:

$$a_i = \frac{f_{SVR}(s_i) - \min_j f_{SVR}(s_j)}{\max_j f_{SVR}(s_j) - \min_j f_{SVR}(s_j)} \tag{14}$$

The objective (15) of our summarization ILP model sums the normalized relevance scores a_i of the selected sentences to estimate the overall importance $imp(S)$ of the resulting summary S. It also estimates the diversity $div(S)$ of S by calculating how many word bigrams of the documents being summarized are present in the selected sentences; when more bigrams are present in the summary, the summary's sentences share fewer bigrams, i.e., they are less redundant. Notice that we do not assign importance scores to the bigrams, unlike the work of Berg-Kirkpatrick et al. and Woodsend and Lapata (Section 2). The binary variables x_i

and b_j indicate which sentences s_i and which word bigrams g_j, respectively, are present in the summary; see Figure 1 for an example of the relations between the x_i and b_j variables.

$$\max_{b,x} \lambda_1 \cdot imp(S) + \lambda_2 \cdot div(S) = \max_{b,x} \lambda_1 \cdot \sum_{i=1}^{n} \frac{a_i}{k_{max}} \cdot x_i + \lambda_2 \cdot \sum_{j=1}^{|B|} \frac{b_j}{n} \tag{15}$$

$$\sum_{i=1}^{n} l_i \cdot x_i \leq L_{max} \tag{16}$$

$$\sum_{g_j \in B_i} b_j \geq |B_i| \cdot x_i, \text{ for } i = 1, \ldots, n \tag{17}$$

$$\sum_{s_i \in S_j} x_i \geq b_j, \text{ for } j = 1, \ldots, |B| \tag{18}$$

Again, l_i is the length of sentence s_i, L_{max} is the maximum allowed summary length, and n is the number of input sentences. $imp(S)$ is normalized to $[0, 1]$ using the maximum number of sentences k_{max} that can be included in the summary. To estimate k_{max} we divide the maximum available space L_{max} by the length of the shortest input sentence. We also divide $div(S)$ by n, which causes $div(S)$ to range mostly in $[0, 1]$ in our experiments. The values of λ_1 and λ_2 are tuned on development data. We set $\lambda_1 + \lambda_2 = 1$. Constraint 16 guarantees that L_{max} is not exceeded. The other two constraints are explained below:

- Constraint 17: B_i is the set of bigrams that appear in sentence s_i, $|B_i|$ is the cardinality of B_i, g_j ranges over the bigrams in B_i, and b_j is the binary variable that shows if bigram g_j has been selected. If a sentence s_i is selected ($x_i = 1$), then all of its bigrams must also be selected, i.e., $\sum_{g_j \in B_i} b_j = |B_i|$ and Constraint 17 holds. If sentence s_i is not selected ($x_i = 0$), then some of its bigrams may still be selected, if they occur in another selected sentence; hence $\sum_{g_j \in B_i} b_j \geq 0$ and Constraint 17 holds again.
- Constraint 18: Again, b_j is the binary variable that shows if g_j has been selected. $|B|$ is the total number of (distinct) bigrams of the n input sentences; S_j is the set of sentences that bigram g_j appears in; and x_i is the binary variable that shows if sentence s_i has been selected. If a bigram g_j is selected ($b_j = 1$), then at least one sentence that contains that bigram must also be selected; hence, $\sum_{s_i \in S_j} x_i \geq 1$ and Constraint 18 holds. If bigram g_j is not selected ($b_j = 0$), then none of the sentences that contain it may be selected; hence, $\sum_{s_i \in S_j} x_i = 0$ and Constraint 18 holds again.

In preliminary experiments, we noticed that our ILP model above, called ILP1, tended to select many short sentences, which had a poor ROUGE match with the gold summaries. To address this issue, we developed an alternative ILP model, called ILP2, whose objective function (19) rewards longer sentences by multiplying their importance scores a_i with their lengths l_i (in words). The constraints of ILP2 remain as in ILP1 (Constraints 16–18).

$$\max_{b,x} \lambda_1 \cdot \sum_{i=1}^{n} a_i \cdot \frac{l_i}{L_{max}} \cdot x_i + \lambda_2 \cdot \sum_{j=1}^{|B|} \frac{b_j}{n} \tag{19}$$

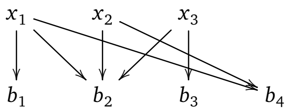

Figure 1: There are 3 sentences (corresponding to the binary variables x_1, x_2, x_3) containing 4 word bigrams (corresponding to variables b_1, b_2, b_3, b_4). For example, sentence s_1 contains the first, second, and fourth bigrams; and if sentences s_1 and s_2 are selected ($x_1 = 1$ and $x_2 = 1$), then the bigrams they contain must also be selected ($b_1 = 1$, $b_2 = 1$, $b_4 = 1$).

4 Experiments

We now present the experiments that we performed, starting from the datasets we used.

4.1 Datasets and experimental setup

We used the datasets of DUC 2005, DUC 2006, DUC 2007, and TAC 2008 (Dang, 2005, 2006; Dang and Owczarzak, 2008).[4] Each dataset contains document clusters. Each cluster contains documents relevant to a query (a question or topic description), which is also given. For each cluster, a summary not exceeding a maximum allowed length has to be produced, so that the summary will provide an answer to the corresponding query. Multiple reference (gold, human-authored) summaries are also provided per cluster. Table 1 provides more information on the datasets we used. For our experiments, we extracted all the sentences from the documents of each cluster, discarding sentences shorter than or equal to 7 words. We also applied a small set of cleanup rules to remove unnecessary formatting tags.

dataset	documents per cluster	clusters	reference summaries	word limit (in words)
DUC 2005	25–50	50	4–9	250
DUC 2006	25	50	4	250
DUC 2007	25	45	4	250
TAC 2008	10	48	4	100

Table 1: Datasets used in our experiments.

The SVR model of sentence importance (Section 3.1) was trained on the sentences of DUC 2006 (i.e., DUC 2006 was our training dataset) and it was used to assign importance scores to the sentences of the clusters of DUC 2005, DUC 2007, and TAC 2008. For each document cluster, we used the $n = 100$ sentences with the highest importance scores as input to the baseline and ILP summarizers of Sections 3.2 and 3.3.

We note that ILP problems are in the worst case (for the most difficult ILP problems) NP-hard. Our ILP1 and ILP2 models (Section 3.3) are generalizations of the 0-1 Knapsack problem, which is known to be NP-hard; hence, our models also constitute NP-hard problems. Nevertheless, very efficient ILP solvers are available.[5] In the worst case, the off-the-shelf solver that we use finds a solution (for ILP1 or ILP2, per summary) in 1.25 seconds and

[4] Consult also `http://duc.nist.gov/` and `http://www.nist.gov/tac/`.

[5] We use the implementation of the Branch and Cut algorithm of the GNU Linear Programming Kit (GLPK); consult `http://sourceforge.net/projects/winglpk/`.

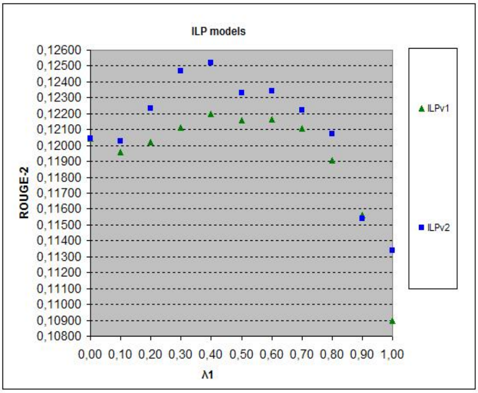

Figure 2: ROUGE-2 scores of the two versions of our ILP model on DUC 2007 data, used as development data, with the SVR model of sentence importance trained on DUC 2006 data.

0.9 seconds in the DUC 2007 and TAC 2008 datasets, respectively. The solver takes more time in DUC 2007 than in TAC 2008, because DUC 2007 summaries are longer (cf. Table 1) and, therefore, the search space is larger. This efficiency is mostly due to the fact that the x_i and b_j variables of ILP1 and ILP2 are in the order of hundreds and grow approximately linearly to the number and size (word bigrams) of the input sentences, as opposed to the quadratic (to the number of sentences) growth of the number of variables in McDonald's model (Section 2). Berg-Kirkpatrick et al. (2011) report very similar execution times; they report that the solver they use finds the solution of their extractive formulation in less than a second for most summaries of TAC 2008 and TAC 2009. Our method (ILP1 or ILP2) takes on average 10–11 seconds to form each summary, including the time to read and preprocess the input documents, formulate the ILP model etc. By contrast Woodsend and Lapata (2012) report that their method takes 55 seconds on average for each summary, though presumably this also includes parsing the input and applying the QSTG grammar.

4.2 Experiments on development data

To determine which of the two versions (ILP1 or ILP2) of our ILP model performs best and to tune their parameters, we experimented on the DUC 2007 dataset, i.e., we used the DUC 2007 dataset as our development data; recall that the DUC 2006 dataset was used as training data in all cases. We used 11 different values of λ_1 ($\lambda_2 = 1 - \lambda_1$) in both ILP1 and ILP2, and we evaluated the generated summaries using ROUGE-2. The results of these experiments are presented in Figure 2. ILP2 is better than ILP1 for all values of λ_1, and its best ROUGE-2 score is obtained for $\lambda_1 = 0.4$ ($\lambda_2 = 0.6$). The fact that the best results were obtained for non-zero λ_1 and λ_2 values also shows that both the sentence importance component (SVR) and the diversity component (bigram count) contribute to the results of our ILP models.

We also compared the average number of selected sentences per cluster of ILP1 and ILP2 on DUC 2007 data. As already noted and illustrated in Figure 3, ILP1 tends to select more and, therefore, shorter sentences than ILP2; these shorter sentences have worse ROUGE matches

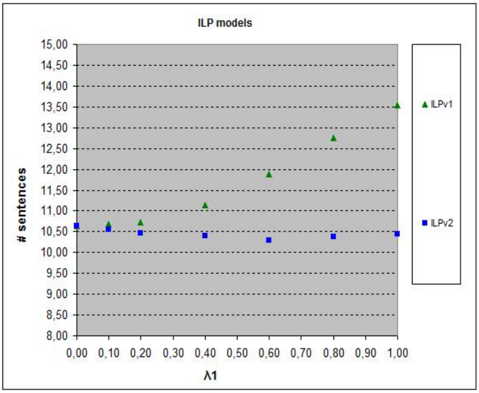

Figure 3: Average number of sentences selected by the two versions of our ILP model on DUC 2007 data, used as development data, with the SVR model trained on DUC 2006 data.

with the reference summaries, which is why ILP1 performs worse than ILP2. Figure 3 also shows that ILP2 selects approximately the same number of sentences for all λ_1 values; this is because ILP2 tends to always select relatively long sentences and, hence, the number of selected sentences that fit in the available space cannot vary as much as in ILP1.

In Table 2, we present the ROUGE scores of the two versions of our ILP model (for $\lambda_1 = 0.4$) on the DUC 2007 dataset, along with the corresponding scores of the GREEDY and GREEDY-RED baselines (Section 3.2), which use only the SVR without ILP. We also show the scores of several state of the art systems, both extractive and abstractive, as they were reported in the corresponding articles; more recently published results are shown first. Our ILP2 model has the best reported ROUGE-2 score on the DUC 2007 dataset, and the second best ROUGE-SU4 score, though one should keep in mind that the DUC 2007 dataset was our development set.

4.3 Experiments on test data

We then evaluated ILP2 with $\lambda_1 = 0.4$, which was our best system in the experiments on the development data (DUC 2007), against the systems with the highest published ROUGE scores on TAC 2008 and DUC 2005 data, our two test datasets.[6] The results of these experiments are listed in Tables 3 and 4, respectively. On the TAC 2008 dataset (Table 3), the most recent of the datasets we experimented with, our ILP2 method achieves the second best ROUGE-SU4 score and the second best ROUGE-2 score, following the method of Woodsend and Lapata with the QSTG grammar enabled, and the abstractive (full) method of Berg-Kirkpatrick et al., respectively (see Section 2). Our ILP2 method performs better than the method of Woodsend and Lapata *without* the QSTG grammar, even though the method of Woodsend and Lapata is still an abstractive one, even without the QSTG grammar (it can still delete parse tree nodes), whereas our method is purely extractive. If we exclude abstractive summarizers, our ILP2 method has the best ROUGE-2 and ROUGE-SU4 scores.

[6] We used Set A of TAC 2008.

system	ROUGE-2	ROUGE-SU4
ILP2	**0.12517**	0.17603
ILP1	0.12201	0.17283
GREEDY-RED	0.11591	0.16908
GREEDY	0.11408	0.16651
Lin and Bilmes 2011	0.12380	N/A
Celikyilmaz and Hakkani-Tur 2010	0.11400	0.17200
Haghighi and Vanderwende 2009	0.11800	0.16700
Schilder and Ravikumar 2008	0.11000	N/A
Pingali et al. 2007 (DUC 2007)	0.12448	**0.17711**
Toutanova et al. 2007 (DUC 2007)	0.12028	0.17074
Conroy et al. 2007 (DUC 2007)	0.11793	0.17593
Amini and Usunier 2007 (DUC 2007)	0.11887	0.16999

Table 2: Comparison of our ILP method against greedy baselines that use the same SVR model of sentence importance without ILP, and against other state of the art summarizers on DUC 2007 data (our development dataset). Our ILP method was trained on DUC 2006 data.

On the DUC 2005 dataset (Table 4), our ILP2 method has the best reported ROUGE-2 and ROUGE-SU4 scores. Berg-Kirkpatrick et al. and Woodsend and Lapata provide no results of their systems for this dataset. They also provide no results for the more recent TAC 2009 dataset, because they used it as their training set. We did not experiment with the TAC 2009 dataset, because our main competitors have not published results for that dataset.

We used paired t-tests ($p < 0.05$) to check if the differences between the scores of ILP2 and the other systems were statistically significant. In Tables 3–4, + and − denote the existence or absence of statistical significance, respectively. Unfortunately, the tests were possible only when comparing ILP2 against the few systems we had ROUGE scores for per topic.

5 Conclusions

We presented a new ILP method (in two versions) for multi-document summarization. Our method jointly maximizes the importance of the sentences it includes in a summary and their diversity, without exceeding a maximum allowed summary length. To obtain an importance score for each sentence, it uses an SVR model, trained on human-authored summaries to predict the ROUGE score of each sentence. Diversity is measured as the number of word bigrams of the input documents that occur in the resulting summary. Experimental results on widely used benchmarks for news summarization show that our ILP method achieves state of the art results among extractive summarizers. It also outperforms two greedy baselines that use the same SVR model of sentence importance without ILP, and it performs better than some abstractive summarizers. Our method is also very fast, and it does not require a parser or other resources that are not always available in less widely spoken languages.

We are already experimenting with an extended version of our method that also performs sentence compression. In future work, we hope to extend our ILP model to consider discourse coherence, sentence aggregation, and referring expression generation.

Acknowledgements

This research was funded by the Research Centre of the Athens University of Economics and Business.

system	ROUGE-2	ROUGE-SU4
ILP2	0.11168	0.14413
Woodsend and Lapata 2012 (with QSTG)	0.11370	**0.14470**
Woodsend and Lapata 2012 (without QSTG)	0.10320	0.13680
Berg-Kirkpatrick et al. 2011 (with subtree cuts)	**0.11700**	0.14380
Berg-Kirkpatrick et al. 2011 (without subtree cuts)	0.11050	0.13860
Shen and Li 2010	0.09012	0.12094
Gillick and Favre 2009 (with sentence compression)	0.11100	N/A
Gillick and Favre 2009 (without sentence compression)	0.11000	N/A
Gillick et al. 2008 (run 43 in TAC 2008)	0.11140^{-}	0.14298^{-}
Gillick et al. 2008 (run 13 in TAC 2008)	0.11044^{-}	0.13985^{-}
Conroy and Schlesinger 2008 (run 60 in TAC 2008)	0.10379^{-}	0.14200^{-}
Conroy and Schlesinger 2008 (run 37 in TAC 2008)	0.10338^{-}	0.14277^{-}
Conroy and Schlesinger 2008 (run 06 in TAC 2008)	0.10133^{+}	0.13977^{-}
Galanis and Malakasiotis 2008 (run 02 in TAC 2008)	0.10012^{+}	0.13694^{-}

Table 3: Comparison of our best ILP summarizer (ILP2) against state of the art summarizers on TAC 2008 data (one of our two test datasets). Our ILP method was trained on DUC 2006 data. It has the best ROUGE-2 and ROUGE-SU4 scores among extractive summarizers.

system	ROUGE-2	ROUGE-SU4
ILP2	**0.08174**	**0.13640**
Lin and Bilmes 2011	0.07820	N/A
Shen and Li 2010	0.07311	0.13061
McDonald 2007 (ILP)	0.06100	N/A
McDonald 2007 (Knapsack)	0.06700	N/A
Ye et al. 2005	0.0744^{+}	0.13461^{-}
Li et al. 2005	0.07313^{+}	0.13158^{-}
Daume and Marcu 2005	0.07089^{+}	0.12649^{+}

Table 4: Comparing our best ILP summarizer (ILP2) against state of the art summarizers on DUC 2005 data (one of our two test datasets). Our method was trained on DUC 2006 data.

References

Amini, M. R. and Usunier, N. (2007). A contextual query expansion approach by term clustering for robust text summarization. In *Proceedings of the Document Understanding Conference*, Rochester, NY.

Berg-Kirkpatrick, T., Gillick, D., and Klein, D. (2011). Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the ACL-HLT*, pages 481–490, Portland, OR.

Celikyilmaz, A. and Hakkani-Tur, D. (2010). A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the ACL*, pages 815–824, Uppsala, Sweden.

Conroy, H. and Schlesinger, J. (2008). CLASSY and TAC 2008 Metrics. In *Proceedings of the Text Analysis Conference*, Gaithersburg, MD.

Conroy, H., Schlesinger, J., and O'Leary, D. (2007). CLASSY 2007 at DUC 2007. In *Proceedings of the Document Understanding Conference*, Rochester, NY.

Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press.

Dang, H. (2005). Overview of DUC 2005. In *Proceedings of the Document Understanding Conference*, Vancouver, B.C., Canada.

Dang, H. (2006). Overview of DUC 2006. In *Proceedings of the Document Understanding Conference*, Brooklyn, NY.

Dang, H. and Owczarzak, K. (2008). Overview of the TAC 2008 update summarization task. In *Proceedings of Text Analysis Conference*, Gaithersburg, MD.

Daume, H. and Marcu, D. (2005). Bayesian summarization at DUC and suggestion for extrinsic evaluation. In *Proceedings of the Document Understanding Conference*, Vancouver, B.C., Canada.

Galanis, D. and Malakasiotis, P. (2008). AUEB at TAC 2008. In *Proceedings of the Text Analysis Conference*, Gaithersburg, MD.

Gillick, D. and Favre, B. (2009). A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18, Boulder, CO.

Gillick, D., Favre, B., and Hakkani-Tur, D. (2008). The ICSI Summarization System at TAC 2008. In *Proceedings of the Text Analysis Conference*, Gaithersburg, MD.

Haghighi, A. and Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The Annual Conference of the NAACL*, pages 362–370, Boulder, CO.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physice-Doklady*, 10(8):707–710.

Li, W., Li, B., Chen, Q., and Wu, M. (2005). The Hong Kong Polytechnic University at DUC 2005. In *Proceedings of the Document Understanding Conference*, Vancouver, B.C., Canada.

Lin, C. (2004). ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the ACL Workshop "Text Summarization Branches Out"*, pages 74–81, Barcelona, Spain.

Lin, H. and Bilmes, J. (2011). A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the ACL-HLT*, volume 1, pages 510–520, Portland, OR.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. In *Proceedings of the European Conference on Information Retrieval*, pages 557–564, Rome, Italy.

Nishikawa, H., Hasegawa, T., Matsuo, Y., and Kikui, G. (2010a). Opinion summarization with integer linear programming formulation for sentence extraction and ordering. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 910–918.

Nishikawa, H., Hasegawa, T., Matsuo, Y., and Kikui, G. (2010b). Optimizing informativeness and readability for sentiment summarization. In *Proceedings of the ACL Conference Short Papers*, pages 325–330, Uppsala, Sweden.

Pingali, P., Rahul, K., and Vasudeva, V. (2007). IIIT Hyderabad at DUC 2007. In *Proceedings of the Document Understanding Conference*, Rochester, NY.

Schilder, F. and Kondadadi, R. (2008). FastSum:Fast and Accurate Query-based Multi-document Summarization. In *Proceedings of the 46th Annual Meeting of the ACL-HLT: Short Papers*, pages 205–208, Columbus, OH.

Shen, C. and Li, T. (2010). Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 984–992, Beijing, China.

Toutanova, K., Brockett, C., Gamon, M., Jagarlamudi, J., Suzuki, H., and Vanderwende, L. (2007). The PYTHY summarization system: Microsoft Research at DUC 2007. In *Proceedings of the Document Understanding Conference*, Rochester, NY.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support Vector Machine learning for independent and structured output spaces. *Machine Learning Research*, 6:1453–1484.

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley.

Woodsend, K. and Lapata, M. (2012). Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 233–243, Jesu Island, Korea.

Ye, S., Qiu, L., Chua, T., and Kan, M. (2005). NUS at DUC 2005: Understanding documents via concept links. In *Proceedings of the Document Understanding Conference*, Vancouver, B.C., Canada.