A Scalable Global Model for Summarization

Dan Gillick[1,2], **Benoit Favre**[2]
[1] Computer Science Division, University of California Berkeley, USA
[2] International Computer Science Institute, Berkeley, USA
{dgillick,favre}@icsi.berkeley.edu

Abstract

We present an Integer Linear Program for exact inference under a maximum coverage model for automatic summarization. We compare our model, which operates at the sub-sentence or “concept”-level, to a sentence-level model, previously solved with an ILP. Our model scales more efficiently to larger problems because it does not require a quadratic number of variables to address redundancy in pairs of selected sentences. We also show how to include sentence compression in the ILP formulation, which has the desirable property of performing compression and sentence selection simultaneously. The resulting system performs at least as well as the best systems participating in the recent Text Analysis Conference, as judged by a variety of automatic and manual content-based metrics.

1 Introduction

Automatic summarization systems are typically extractive or abstractive. Since abstraction is quite hard, the most successful systems tested at the Text Analysis Conference (TAC) and Document Understanding Conference (DUC)[1], for example, are extractive. In particular, sentence selection represents a reasonable trade-off between linguistic quality, guaranteed by longer textual units, and summary content, often improved with shorter units.

Whereas the majority of approaches employ a greedy search to find a set of sentences that is both relevant and non-redundant (Goldstein et al., 2000; Nenkova and Vanderwende, 2005), some recent work focuses on improved search (McDonald, 2007; Yih et al., 2007). Among them, McDonald is the first to consider a non-approximated maximization of an objective function through Integer Linear Programming (ILP), which improves on a greedy search by 4-12%. His formulation assumes that the quality of a summary is proportional to the sum of the relevance scores of the selected sentences, penalized by the sum of the redundancy scores of all pairs of selected sentences. Under a maximum summary length constraint, this problem can be expressed as a quadratic knapsack (Gallo et al., 1980) and many methods are available to solve it (Pisinger et al., 2005). However, McDonald reports that the method is not scalable above 100 input sentences and discusses more practical approximations. Still, an ILP formulation is appealing because it gives exact solutions and lends itself well to extensions through additional constraints.

Methods like McDonald’s, including the well-known Maximal Marginal Relevance (MMR) algorithm (Goldstein et al., 2000), are subject to another problem: Summary-level redundancy is not always well modeled by pairwise sentence-level redundancy. Figure 1 shows an example where the combination of sentences (1) and (2) overlaps completely with sentence (3), a fact not captured by pairwise redundancy measures. Redundancy, like content selection, is a global problem.

Here, we discuss a model for sentence selection with a globally optimal solution that also addresses redundancy globally. We choose to represent infor-

[1] TAC is a continuation of DUC, which ran from 2001-2007.

(1) The cat is in the kitchen.
(2) The cat drinks the milk.
(3) The cat drinks the milk in the kitchen.

Figure 1: Example of sentences redundant as a group. Their redundancy is only partially captured by sentence-level pairwise measurement.

mation at a finer granularity than sentences, with concepts, and assume that the value of a summary is the sum of the values of the unique concepts it contains. While the concepts we use in experiments are word n-grams, we use the generic term to emphasize that this is just one possible definition. Only crediting each concept once serves as an implicit global constraint on redundancy. We show how the resulting optimization problem can be mapped to an ILP that can be solved efficiently with standard software.

We begin by comparing our model to McDonald's (section 2) and detail the differences between the resulting ILP formulations (section 3), showing that ours can give competitive results (section 4) and offer better scalability[2] (section 5). Next we demonstrate how our ILP formulation can be extended to include efficient parse-tree-based sentence compression (section 6). We review related work (section 7) and conclude with a discussion of potential improvements to the model (section 8).

2 Models

The model proposed by McDonald (2007) considers information and redundancy at the sentence level. The score of a summary is defined as the sum of the relevance scores of the sentences it contains minus the sum of the redundancy scores of each pair of these sentences. If s_i is an indicator for the presence of sentence i in the summary, Rel_i is its relevance, and Red_{ij} is its redundancy with sentence j, then a summary is scored according to:

$$\sum_i Rel_i s_i - \sum_{ij} Red_{ij} s_i s_j$$

Generating a summary under this model involves maximizing this objective function, subject to a length constraint. A variety of choices for Rel_i and Red_{ij} are possible, from simple word overlap metrics to the output of feature-based classifiers trained to perform information retrieval and textual entailment.

As an alternative, we consider information and redundancy at a sub-sentence, "concept" level, modeling the value of a summary as a function of the concepts it covers. While McDonald uses an explicit redundancy term, we model redundancy implicitly: a summary only benefits from including each concept once. With c_i an indicator for the presence of concept i in the summary, and its weight w_i, the objective function is:

$$\sum_i w_i c_i$$

We generate a summary by choosing a set of sentences that maximizes this objective function, subject to the usual length constraint.

In summing over concept weights, we assume that the value of including a concept is not effected by the presence of any other concept in the summary. That is, concepts are assumed to be independent. Choosing a suitable definition for concepts, and a mapping from the input documents to concept weights, is both important and difficult. Concepts could be words, named entities, syntactic subtrees or semantic relations, for example. While deeper semantics make more appealing concepts, their extraction and weighting are much more error-prone. Any error in concept extraction can result in a biased objective function, leading to poor sentence selection.

3 Inference by ILP

Each model presented above can be formalized as an Integer Linear Program, with a solution representing an optimal selection of sentences under the objective function, subject to a length constraint. McDonald observes that the redundancy term makes for a quadratic objective function, which he coerces to a linear function by introducing additional variables s_{ij} that represent the presence of both sentence i and sentence j in the summary. Additional constraints ensure the consistency between the sentence variables (s_i, s_j) and the quadratic term (s_{ij}). With l_i the length of sentence i and L the length limit for

[2] Strictly speaking, exact inference for the models discussed in this paper is NP-hard. Thus we use the term "scalable" in a purely practical sense.

the whole summary, the resulting ILP is:

$$\begin{aligned}
\text{Maximize: } & \sum_i Rel_i s_i - \sum_{ij} Red_{ij} s_{ij} \\
\text{Subject to: } & \sum_j l_j s_j \leq L \\
& s_{ij} \leq s_i \quad s_{ij} \leq s_j \quad \forall i, j \\
& s_i + s_j - s_{ij} \leq 1 \quad \forall i, j \\
& s_i \in \{0, 1\} \quad \forall i \\
& s_{ij} \in \{0, 1\} \quad \forall i, j
\end{aligned}$$

To express our concept-based model as an ILP, we maintain our notation from section 2, with c_i an indicator for the presence of concept i in the summary and s_j an indicator for the presence of sentence j in the summary. We add Occ_{ij} to indicate the occurrence of concept i in sentence j, resulting in a new ILP:

$$\begin{aligned}
\text{Maximize: } & \sum_i w_i c_i \\
\text{Subject to: } & \sum_j l_j s_j \leq L \\
& s_j Occ_{ij} \leq c_i, \quad \forall i, j & (1) \\
& \sum_j s_j Occ_{ij} \geq c_i \quad \forall i & (2) \\
& c_i \in \{0, 1\} \quad \forall i \\
& s_j \in \{0, 1\} \quad \forall j
\end{aligned}$$

Note that Occ, like Rel and Red, is a constant parameter. The constraints formalized in equations (1) and (2) ensure the logical consistency of the solution: selecting a sentence necessitates selecting all the concepts it contains and selecting a concept is only possible if it is present in at least one selected sentence. Constraint (1) also prevents the inclusion of concept-less sentences.

4 Performance

Here we compare both models on a common summarization task. The data is part of the Text Analysis Conference (TAC) multi-document summarization evaluation and involves generating 100-word summaries from 10 newswire documents, each on a given topic. While the 2008 edition of TAC also includes an update task—additional summaries assuming some prior knowledge—we focus only on the standard task. This includes 48 topics, averaging 235 input sentences (ranging from 47 to 652). Since the mean sentence length is around 25 words, a typical summary consists of 4 sentences.

In order to facilitate comparison, we generate summaries from both models using a common pipeline:

1. Clean input documents. A simple set of rules removes headers and formatting markup.

2. Split text into sentences. We use the unsupervised Punkt system (Kiss and Strunk, 2006).

3. Prune sentences shorter than 5 words.

4. Compute parameters needed by the models.

5. Map to ILP format and solve. We use an open source solver[3].

6. Order sentences picked by the ILP for inclusion in the summary.

The specifics of step 4 are described in detail in (McDonald, 2007) and (Gillick et al., 2008). McDonald's sentence relevance combines word-level cosine similarity with the source document and the inverse of its position (early sentences tend to be more important). Redundancy between a pair of sentences is their cosine similarity. For sentence i in document D,

$$\begin{aligned}
Rel_i &= cosine(i, D) + 1/pos(i, D) \\
Red_{ij} &= cosine(i, j)
\end{aligned}$$

In our concept-based model, we use word bigrams, weighted by the number of input documents in which they appear. While word bigrams stretch the notion of a concept a bit thin, they are easily extracted and matched (we use stemming to allow slightly more robust matching). Table 1 provides some justification for document frequency as a weighting function. Note that bigrams gave consistently better performance than unigrams or trigrams for a variety of ROUGE measures. Normalizing by document frequency measured over a generic set (TFIDF weighting) degraded ROUGE performance.

[3] gnu.org/software/glpk

Bigrams consisting of two stopwords are pruned, as are those appearing in fewer than three documents.

We largely ignore the sentence ordering problem, sorting the resulting sentences first by source document date, and then by position, so that the order of two originally adjacent sentences is preserved, for example.

Doc. Freq. (D)	1	2	3	4	5	6
In Gold Set	156	48	25	15	10	7
Not in Gold Set	5270	448	114	42	21	11
Relevant (P)	0.03	0.10	0.18	0.26	0.33	0.39

Table 1: There is a strong relationship between the document frequency of input bigrams and the fraction of those bigrams that appear in the human generated "gold" set: Let d_i be document frequency i and p_i be the percent of input bigrams with d_i that are actually in the gold set. Then the correlation $\rho(D, P) = 0.95$ for DUC 2007 and 0.97 for DUC 2006. Data here averaged over all problems in DUC 2007.

The summaries produced by the two systems have been evaluated automatically with ROUGE and manually with the Pyramid metric. In particular, ROUGE-2 is the recall in bigrams with a set of human-written abstractive summaries (Lin, 2004). The Pyramid score arises from a manual alignment of basic facts from the reference summaries, called Summary Content Units (SCUs), in a hypothesis summary (Nenkova and Passonneau, 2004). We used the SCUs provided by the TAC evaluation.

Table 2 compares these results, alongside a baseline that uses the first 100 words of the most recent document. All the scores are significantly different, showing that according to both human and automatic content evaluation, the concept-based model outperforms McDonald's sentence-based model, which in turn outperforms the baseline. Of course, the relevance and redundancy functions used for McDonald's formulation in this experiment are rather primitive, and results would likely improve with better relevance features as used in many TAC systems. Nonetheless, our system based on word bigram concepts, similarly primitive, performed at least as well as any in the TAC evaluation, according to two-tailed t-tests comparing ROUGE, Pyramid, and manually evaluated "content responsiveness" (Dang and Owczarzak, 2008) of our system and the highest scoring system in each category.

System	**ROUGE-2**	**Pyramid**
Baseline	0.058	0.186
McDonald	0.072	0.295
Concepts	0.110	0.345

Table 2: Scores for both systems and a baseline on TAC 2008 data (Set A) for ROUGE-2 and Pyramid evaluations.

5 Scalability

McDonald's sentence-level formulation corresponds to a quadratic knapsack, and he shows his particular variant is NP-hard by reduction to 3-D matching. The concept-level formulation is similar in spirit to the classical maximum coverage problem: Given a set of items X, a set of subsets S of X, and an integer k, the goal is to pick at most k subsets from S that maximizes the size of their union. Maximum coverage is known to be NP-hard by reduction to the set cover problem (Hochbaum, 1996).

Perhaps the simplest way to show that our formulation is NP-hard is by reduction to the knapsack problem (Karp, 1972). Consider the special case where sentences do not share any overlapping concepts. Then, the value of each sentence to the summary is independent of every other sentence. This is a knapsack problem: trying to maximize the value in a container of limited size. Given a solver for our problem, we could solve all knapsack problem instances, so our problem must also be NP-hard.

With n input sentences and m concepts, both formulations generate a quadratic number of constraints. However, McDonald's has $O(n^2)$ variables while ours has $O(n + m)$. In practice, scalability is largely determined by the sparsity of the redundancy matrix Red and the sentence-concept matrix Occ. Efficient solutions thus depend heavily on the choice of redundancy measure in McDonald's formulation and the choice of concepts in ours. Pruning to reduce complexity involves removing low-relevance sentences or ignoring low redundancy values in the former, and corresponds to removing low-weight concepts in the latter. Note that pruning concepts may be more desirable: Pruned sentences are irretrievable, but pruned concepts may well appear in the selected sentences through co-occurrence.

Figure 2 compares ILP run-times for the two formulations, using a set of 25 topics from DUC 2007, each of which have at least 500 input sentences. These are very similar to the TAC 2008 topics, but more input documents are provided for each topic, which allowed us to extend the analysis to larger problems. While the ILP solver finds optimal solutions efficiently for our concept-based formulation, run-time for McDonald's approach grows very rapidly. The plot includes timing results for 250-word summaries as well, showing that our approach is fast even for much more complex problems: A rough estimate for the number of possible summaries has $\binom{500}{4} = 2.6 \times 10^9$ for 100-word summaries and $\binom{500}{10} = 2.5 \times 10^{20}$ for 250 words summaries.

While exact solutions are theoretically appealing, they are only useful in practice if fast approximations are inferior. A greedy approximation of our objective function gives 10% lower ROUGE scores than the exact solution, a gap that separates the highest scoring systems from the middle of the pack in the TAC evaluation. The greedy solution (linear in the number of sentences, assuming a constant summary length) marks an upper bound on speed and a lower bound on performance; The ILP solution marks an upper bound on performance but is subject to the perils of exponential scaling. While we have not experimented with much larger documents, approximate methods will likely be valuable in bridging the performance gap for complex problems. Preliminary experiments with local search methods are promising in this regard.

6 Extensions

Here we describe how our ILP formulation can be extended with additional constraints to incorporate sentence compression. In particular, we are interested in creating compressed alternatives for the original sentence by manipulating its parse tree (Knight and Marcu, 2000). This idea has been applied with some success to summarization (Turner and Charniak, 2005; Hovy et al., 2005; Nenkova, 2008) with the goal of removing irrelevant or redundant details, thus freeing space for more relevant information. One way to achieve this end is to generate compressed candidates for each sentence, creating an expanded pool of input sentences, and employ some redundancy removal on the final selection (Madnani et al., 2007).

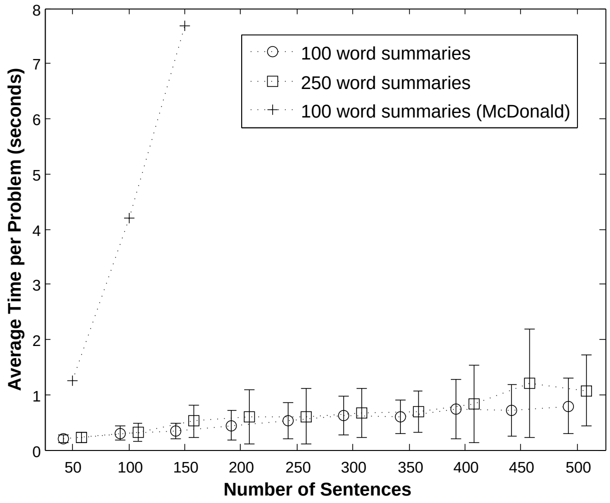


Figure 2: A comparison of ILP run-times (on an AMD 1.8Ghz desktop machine) of McDonald's sentence-based formulation and our concept-based formulation with an increasing number of input sentences.

We adapt this approach to fit the ILP formulations so that the optimization procedure decides which compressed alternatives to pick. Formally, each compression candidate belongs to a group g_k corresponding to its original sentence. We can then craft a constraint to ensure that at most one sentence can be selected from group g_k, which also includes the original:

$$\sum_{i \in g_k} s_i \leq 1, \forall g_k$$

Assuming that all the compressed candidates are themselves well-formed, meaningful sentences, we would expect this approach to generate higher quality summaries. In general, however, compression algorithms can generate an exponential number of candidates. Within McDonald's framework, this can increase the number of variables and constraints tremendously. Thus, we seek a compact representation for compression in our concept framework.

Specifically, we assume that compression involves some combination of three basic operations on sentences: extraction, removal, and substitution. In extraction, a sub-sentence (perhaps the content of a quotation) may be used independently, and the rest of the sentence is dropped. In removal, a substring is dropped (a temporal clause, for example) that preserves the grammaticality of the sentence. In substitution, one substring is replaced by another (US replaces United States, for example).

Arbitrary combinations of these operations are too general to be represented efficiently in an ILP. In particular, we need to compute the length of a sentence and the concepts it covers for all compression candidates. Thus, we insist that the operations can only affect non-overlapping spans of text, and end up with a tree representation of each sentence: Nodes correspond to compression operations and leaves map to the words. Each node holds the length it contributes to the sentence recursively, as the sum of the lengths of its children. Similarly, the concepts covered by a node are the union of the concepts covered by its children. When a node is activated in the ILP, we consider that the text attached to it is present in the summary and update the length constraint and concept selection accordingly. Figure 3 gives an example of this tree representation for a sentence from the TAC data, showing the derivations of some compressed candidates.

For a given sentence j, let N_j be the set of nodes in its compression tree, $E_j \subseteq N_j$ be the set of nodes that can be extracted (used as independent sentences), $R_j \subseteq N_j$ be the set of nodes that can be removed, and $S_j \subseteq N_j$ be the set of substitution group nodes. Let x and y be nodes from N_j; we create binary variables n_x and n_y to represent the inclusion of x or y in the summary. Let $x \succ y$ denote the fact that $x \in N_j$ is a direct parent of $y \in N_j$. The constraints corresponding to the compression tree are:

$$\sum_{x \in E_j} n_x \leq 1 \quad \forall j \tag{3}$$

$$\sum_{x \succ y} n_y = n_x \quad \forall x \in S_j \quad \forall j \tag{4}$$

$$n_x \geq n_y \quad \forall (y \succ x \wedge x \notin \{R_j \cup S_j\}) \quad \forall j \tag{5}$$

$$n_x \leq n_y \quad \forall (y \succ x \wedge x \notin \{E_j \cup S_j\}) \quad \forall j \tag{6}$$

Eq. (3) enforces that only one sub-sentence is extracted from the original sentence; eq. (4) enforces that one child of a substitution group is selected if and only if the substitution node is selected; eq. (5) ensures that a child node is selected when its parent is selected unless the child is removable (or a substitution group); eq. (6) ensures that if a child node is selected, its parent is also selected unless the child is an extraction node (that can be used as a root).

Each node is associated with the words and the concepts it contains directly (which are not contained by a child node) in order to compute the new length constraints and activate concepts in the objective function. We set Occ_{ix} to represent the occurrence of concept i in node x as a direct child. Let l_x be the length contributed to node x as direct children. The resulting ILP for performing sentence compression jointly with sentence selection is:

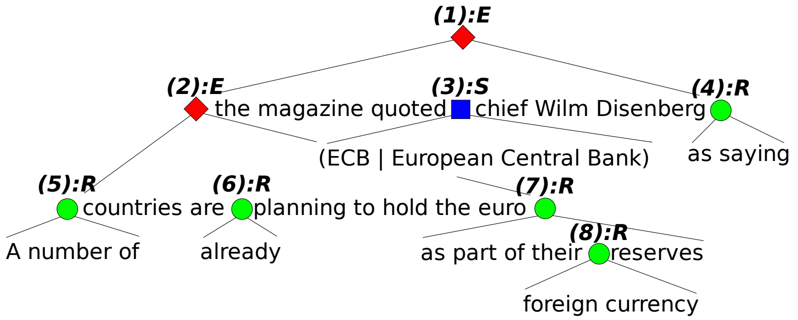
$$\begin{aligned}
\text{Maximize: } & \sum_i w_i c_i \\
\text{Subject to: } & \sum_j l_x n_x \leq L \\
& n_x Occ_{ix} \leq c_i, \quad \forall i, x \\
& \sum_x n_x Occ_{ix} \geq c_i \quad \forall i \\
& \text{idem constraints (3) to (6)} \\
& c_i \in \{0, 1\} \quad \forall i \\
& n_x \in \{0, 1\} \quad \forall x
\end{aligned}$$

While this framework can be used to implement a wide range of compression techniques, we choose to derive the compression tree from the sentence's parse tree, extracted with the Berkeley parser (Petrov and Klein, 2007), and use a set of rules to label parse tree nodes with compression operations. For example, declarative clauses containing a subject and a verb are labeled with the extract (E) operation; adverbial clauses and non-mandatory prepositional clauses are labeled with the remove (R) operation; Acronyms can be replaced by their full form by using substitution (S) operations and a primitive form of co-reference resolution is used to allow the substitution of noun phrases by their referent.

System	**R-2**	**Pyr.**	**LQ**
No comp.	0.110	0.345	2.479
Comp.	0.111	0.323	2.021

Table 3: Scores of the system with and without sentence compression included in the ILP (TAC'08 Set A data).

When implemented in the system presented in section 4, this approach gives a slight improvement



Node	Len.	Concepts
(1):E	6	{the_magazine, magazine_quoted, chief_Wilm, Wilm_Disenberg}
(2):E	7	{countries_are, planning_to, to_hold, hold_the, the_euro}
(3):S	0	{}
(3a)	1	{ECB}
(3b)	3	{European_Central, Central_Bank}
(4):R	2	{as_saying}
(5):R	3	{a_number, number_of}
(6):R	1	{}
(7):R	5	{as_part, part_of, reserves}
(8):R	2	{foreign_currency}

- Original: *A number of Countries are already planning to hold the euro as part of their foreign currency reserves, the magazine quoted European Central Bank chief Wim Duisenberg as saying.*
- [1,2,5,3a]: *A number of countries are planning to hold the euro, the magazine quoted ECB chief Wim Duisenberg.*
- [2,5,6,7,8]: *A number of countries are already planning to hold the euro as part of their foreign currency reserves.*
- [2,7,8]: *Countries are planning to hold the euro as part of their foreign currency reserves.*
- [2]: *Countries are planning to hold the euro.*

Figure 3: A compression tree for an example sentence. E-nodes (diamonds) can be extracted and used as an independent sentences, R-nodes (circles) can be removed, and S-nodes (squares) contain substitution alternatives. The table shows the word bigram concepts covered by each node and the length it contributes to the summary. Examples of resulting compression candidates are given on the right side, with the list of nodes activated in their derivations.

in ROUGE-2 score (see Table 3), but a reduction in Pyramid score. An analysis of the resulting summaries showed that the rules used for implementing sentence compression fail to ensure that all compression candidates are valid sentences, and about 60% of the summaries contain ungrammatical sentences. This is confirmed by the linguistic quality[4] score drop for this system. The poor quality of the compressed sentences explains the reduction in Pyramid scores: Human judges tend to not give credit to ungrammatical sentences because they obscure the SCUs.

We have shown in this section how sentence compression can be implemented in a more scalable way under the concept-based model, but it remains to be shown that such a technique can improve summary quality.

7 Related work

In addition to proposing an ILP for the sentence-level model, McDonald (2007) discusses a kind of summary-level model: The score of a summary is determined by its cosine similarity to the collection of input documents. Though this idea is only implemented with approximate methods, it is similar in spirit to our concept-based model since it relies on weights for individual summary words rather than sentences.

Using a maximum coverage model for summarization is not new. Filatova (2004) formalizes the idea, discussing its similarity to the classical NP-hard problem, but in the end uses a greedy approximation to generate summaries. More recently, Yih et al. (2007) employ a similar model and uses a stack decoder to improve on a greedy search. Globally optimal summaries are also discussed by Liu (2006) and Jaoua Kallel (2004) who apply genetic algorithms for finding selections of sentences that maximize summary-level metrics. Hassel (2006) uses hill climbing to build summaries that maximize a global information criterion based on random indexing.

The general idea of concept-level scoring for summarization is employed in the SumBasic system (Nenkova and Vanderwende, 2005), which chooses sentences greedily according to the sum of their word values (values are derived from fre-

[4] As measured according to the TAC'08 guidelines.

quency). Conroy (2006) describes a bag-of-words model, with the goal of approximating the distribution of words from the input documents in the summary. Others, like (Yih et al., 2007) train a model to learn the value of each word from a set of features including frequency and position. Filatova's model is most theoretically similar to ours, though the concepts she chooses are "events".

8 Conclusion and Future Work

We have synthesized a number of ideas from the field of automatic summarization, including concept-level weighting, a maximum coverage model to minimize redundancy globally, and sentence compression derived from parse trees. While an ILP formulation for summarization is not novel, ours provides reasonably scalable, efficient solutions for practical problems, including those in recent TAC and DUC evaluations. We have also shown how it can be extended to perform sentence compression and sentence selection jointly.

In ROUGE and Pyramid evaluation, our system significantly outperformed McDonald's ILP system. However, we would note that better design of sentence-level scoring would likely yield better results as suggested by the success of greedy sentence-based methods at the DUC and TAC conferences (see for instance (Toutanova et al., 2007)). Still, the performance of our system, on par with the current state-of-the-art, is encouraging.

There are three principal directions for future work. First, word bigram concepts are convenient, but semantically unappealing. We plan to explore concepts derived from parse trees, where weights may be a function of frequency as well as hierarchical relationships.

Second, our current approach relies entirely on word frequency, a reasonable proxy for relevance, but likely inferior to learning weights from training data. A number of systems have shown improvements by learning word values, though preliminary attempts to improve on our frequency heuristic by learning bigram values have not produced significant gains. Better features may be necessary. However, since the ILP gives optimal solutions so quickly, we are more interested in discriminative training where we learn weights for features that push the resulting summaries in the right direction, as opposed to the individual concept values.

Third, our rule-based sentence compression is more of a proof of concept, showing that joint compression and optimal selection is feasible. Better statistical methods have been developed for producing high quality compression candidates (McDonald, 2006), that maintain linguistic quality, some recent work even uses ILPs for exact inference (Clarke and Lapata, 2008). The addition of compressed sentences tends to yield less coherent summaries, making sentence ordering more important. We would like to add constraints on sentence ordering to the ILP formulation to address this issue.

Acknowledgements

This work is supported by the Defense Advanced Research Projects Agency (DARPA) GALE project, under Contract No. HR0011-06-C-0023. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

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

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

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 Update Summarization Task. In *Proceedings of Text Analysis Conference*.

E. Filatova and V. Hatzivassiloglou. 2004. Event-based extractive summarization. In *Proceedings of ACL Workshop on Summarization*, volume 111.

G. Gallo, PL Hammer, and B. Simeone. 1980. Quadratic knapsack problems. *Mathematical Programming Study*, 12:132–149.

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

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

Martin Hassel and Jonas Sjöbergh. 2006. Towards holistic summarization: Selecting summaries, not sentences. In *Proceedings of Language Resources and Evaluation*.

D.S. Hochbaum. 1996. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. *PWS Publishing Co. Boston, MA, USA*, pages 94–143.

E. Hovy, C.Y. Lin, and L. Zhou. 2005. A BE-based multi-document summarizer with sentence compression. In *Proceedings of Multilingual Summarization Evaluation*.

Fatma Jaoua Kallel, Maher Jaoua, Lamia Belguith Hadrich, and Abdelmajid Ben Hamadou. 2004. Summarization at LARIS Laboratory. In *Proceedings of the Document Understanding Conference*.

Richard Manning Karp. 1972. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 43:85–103.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32.

K. Knight and D. Marcu. 2000. Statistics-Based Summarization-Step One: Sentence Compression. In *Proceedings of the National Conference on Artificial Intelligence*, pages 703–710. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26.

D. Liu, Y. Wang, C. Liu, and Z. Wang. 2006. Multiple Documents Summarization Based on Genetic Algorithm. *Lecture Notes in Computer Science*, 4223:355.

N. Madnani, D. Zajic, B. Dorr, N.F. Ayan, and J. Lin. 2007. Multiple Alternative Sentence Compressions for Automatic Text Summarization. In *Proceedings of the Document Understanding Conference at NLT/NAACL*.

R. McDonald. 2006. Discriminative sentence compression with soft syntactic constraints. In *Proceedings of the 11th EACL*, pages 297–304.

R. McDonald. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. *Lecture Notes in Computer Science*, 4425:557.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of HLT-NAACL*.

A. Nenkova and L. Vanderwende. 2005. The impact of frequency on summarization. Technical Report MSR-TR-2005-101, Microsoft Research, Redmond, Washington.

A. Nenkova. 2008. Entity-driven rewrite for multidocument summarization. *Proceedings of IJCNLP*.

Slav Petrov and Dan Klein. 2007. Learning and inference for hierarchically split PCFGs. In *AAAI 2007 (Nectar Track)*.

D. Pisinger, A.B. Rasmussen, and R. Sandvik. 2005. Solution of large-sized quadratic knapsack problems through aggressive reduction. *INFORMS Journal on Computing*.

K. Toutanova, C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende. 2007. The Pythy Summarization System: Microsoft Research at DUC 2007. In *Proceedings of the Document Understanding Conference*.

J. Turner and E. Charniak. 2005. Supervised and Unsupervised Learning for Sentence Compression. In *Proceedings of ACL*.

W. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. 2007. Multi-document summarization by maximizing informative content-words. In *International Joint Conference on Artificial Intelligence*.