The ICSI Summarization System at TAC 2008

Dan Gillick, Benoit Favre, Dilek Hakkani-Tür
International Computer Science Institute, Berkeley, USA
{dgillick,favre,dilek}@icsi.berkeley.edu

Abstract

The ICSI multi-document summarization system relies on a general framework that casts summarization as a global optimization problem with an integer linear programming solution. Our primary submission, a simple sentence extractor with an n-gram frequency heuristic, gives results at least as good as any reported on the non-update part of the main task. Our secondary submission adds compressed sentence alternatives, achieving high ROUGE scores but lower manual scores. We also observe that an oracle version of our sentence extractor is nearly a direct optimization of ROUGE. We show oracle results for the TAC data set and discuss their significance. Finally, we provide a detailed analysis of the linguistic quality of our two systems, suggesting specifically where improvements might be most useful.

1 Introduction

Conventional wisdom suggests that successful multi-document summarization involves fitting as many important facts into a summary with as little redundancy as possible. Maximum Marginal Relevance (Carbonell and Goldstein, 1998) and SumBasic (Nenkova and Vanderwende, 2005), for example, are greedy-search variations on this theme, and have performed quite well.

Here we introduce a new framework that allows us to formalize this basic idea as a single optimization problem. Generally speaking, the model assumes that the input documents contain a variety of concepts, each with some underlying value. Further, any collection of concepts has a total value equal to the sum of the values of the unique concepts it contains. The goal of summarization, then, is to find the collection with maximum value, subject to a length constraint. This problem is related to the famous knapsack problem, and can be solved efficiently with an integer linear programming (ILP) solver.

In order to preserve readability, we test two possible constraints: our primary system only allows full sentences from the input documents, and our secondary system allows the ILP to select among compressed variants of each input sentence.

To build these systems, we only need to choose concepts and a method for assigning them values. Ideally, a concept is a logical, independent fact, much like those used by the Pyramid method (Nenkova and Passonneau, 2004). As these require manual identification, we opt for word n-gram concepts, valued by their frequency in the input documents, though it is clear that more sophisticated concepts would be preferable. We also observe that a simple modification to our system – using n-grams from the human "gold" summaries – gives oracle summaries that effectively maximize ROUGE score.

In the next section, we formalize summarization as an ILP that maximizes relevance, while non-redundancy and linguistic quality are enforced by global constraints. Next, we describe our systems in detail and show results relative to the other participants in the update task. We also give oracle results and discuss their significance. Finally, we provide a more detailed analysis of the summaries produced by our systems, in particular suggesting how linguistic quality is effected by sentence compression.

2 A Global Framework for Summarization

We start by defining a function that attributes values to summaries. Summarization can thus be expressed as an optimization problem, a search over summary space subject to constraints. For clarity, we refer to sentences (indexed by j), though in general these could be replaced by any textual units (paragraphs, compressed sentences, etc.). Concepts (indexed by i) are a general set of information units which will be word n-grams in our experiments. We want to maximize the number of concepts covered by a selection of sentences:

$$\text{maximize} \qquad \sum_i w_i c_i \tag{1}$$

where w_i is the weight of concept i and c_i is a binary variable indicating the presence of that concept in the summary. The score of a summary is the weighted sum of the concepts it contains. While this function gives a selection over concepts, we are actually interested in a selection over sentences. Thus, we introduce s_j, a binary variable representing the selection of sentence j for the summary. Next, we add a length constraint:

$$\text{subject to} \qquad \sum_j l_j s_j < L \tag{2}$$

where l_j is the length of sentence j and L is the maximum summary length. Now we need to tie sentences and concepts together to maintain consistency. A concept can be selected only if it is present in at least one selected sentence and a sentence can be selected only if all concepts it contains are selected. Formally, this can be represented by two types of constraints:

$$\sum_j s_j o_{ij} \geq c_i \forall i \tag{3}$$

$$s_j o_{ij} \leq c_i \forall i, j \tag{4}$$

where o_{ij} indicates the presence of concept i in sentence j. While this can lead to $O(n^2)$ constraints, in practice $o_{ij} = 0$ for most of the concept-sentence pairs, keeping the number of effective constraints quite low. Lastly, we formalize the variables introduced above, c_i and s_j:

$$c_i = 0 \text{ or } 1, \forall i \qquad s_j = 0 \text{ or } 1, \forall j \tag{5}$$

This formulation is an *integer linear program*, a single linear maximization term subject to a number of linear equality or inequality constraints on integer-valued variables. While the ILP problem is NP-hard in general (Karp, 1972), considerable optimization research has produced software for solving instances efficiently[1].

Note that there is no explicit redundancy term in this formulation. Instead, redundancy is limited implicitly by the fact that concept values are only counted once, combined with a length constraint that prefers sentences with high concept density. Moreover, the solver usually finds an exact solution to the problem very quickly, depending on the choice of concepts.

Assigning value to sub-sentence units is motivated by (Nenkova and Vanderwende, 2005), who choose sentences greedily according to the sum of their word values. This work is extended by (Yih et al., 2007), who use frequency and position features to learn word probabilities, and a stack decoder to maximize the total probability of a summary. The framework presented here is very similar, though the motivation is non-statistical, the implementation is simpler and purely heuristic, and the resulting ILP solutions are not approximate. In terms of exact solutions, (McDonald, 2007) adapts the MMR framework and gives an ILP formulation with explicit relevance and redundancy terms. While the resulting summaries represent a global maximum, the formulation is different from ours, and in practice, that work uses whole sentences rather than sub-sentence fragments as units of selection.

3 Building Systems

Our formulation thus far is quite general. To specify a system, we need to define our concepts, provide a function that maps the input documents to (concept, value) pairs, and pick a constraint on units of selection. Here, we describe three systems: our primary system, our secondary system (a modification

[1] We use the open source solver from gnu.org/software/glpk

of the primary system to include compressed sentences), and an oracle system.

3.1 ICSI-1 (sysid 13)

Concepts

For concepts, we use tokenized, stemmed[2] word n-grams. Certainly this choice tends to undermine the assumption that concepts are independent, atomic units of information, since bigrams like "president said" or "does not" seem less facts themselves than parts of facts. Still, we are interested to see how valuable such simple concepts can be.

Mapping Function

Each n-gram in the input document set is mapped to a real value according to the number of documents in which it occurs. Frequency is one of the best-known proxies for relevance in summarization (Nenkova and Vanderwende, 2005), and document frequency in particular appears to outperform other frequency measures in multi-document summarization (Schilder and Kondadadi, 2008). Specifically, we use bigrams, ignoring those appearing in fewer than three documents. To make use of a query, as is given in TAC problems (we form a query by concatenating the title and narrative description), we ignore sentences that do not share some non-stopword with the query when computing document frequency. Finally, bigrams consisting solely of stopwords are discarded. Table 1, below, and Table 2, in section 4.1, help motivate the choice of document frequency over bigrams.

Though we experimented with a variety of modifications to this simpler mapper, nothing we tried gave significant ROUGE improvements. It is likely, however, that a more sophisticated mapping function would yield improved summaries.

Units of Selection

One of the benefits of assigning value to word-level concepts as opposed to sentences is that we are not necessarily forced to select full sentences for our summaries. In particular, we could allow compressed versions of sentences in the set $\{s_j\}$ without changing the framework. For now, we let $\{s_j\}$ be the set of all input sentences longer than four words.

Preprocessing

For sentence segmentation we use the NLTK[3] implementation of the unsupervised Punkt sentence segmenter (Kiss and Strunk, 2006), which gives results comparable with some of the best rule-based and supervised systems. While many competitive systems employ a long list of rules for trimming sentences, we have only a few simple rules for removing formatting markup. This is an obvious gap in our system, and such a hand-crafted list would likely improve both content and linguistic quality.

Post-processing

We employ the simplest possible ordering of selected sentences. Sentences are sorted first by source document date, and second by their order in the source documents.

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 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.

3.2 ICSI-2 (sysid 43)

This system uses the same components as ICSI-1, except that it includes a sentence rewriting module that generates alternative compressed sentences for use as extra candidates during the sentence selection stage, much like (Madnani et al., 2007). The alternative sentences can be included in the ILP formulation by adding extra constraints to ensure that only one of the sentences derived from an original candidate can be selected:

$$\sum_j s_j g_{jk} \leq 1 \quad \forall k \tag{6}$$

where g_{jk} signals that sentence j belongs to group k. This simple extension is effective at pruning affiliated candidates while searching for the global optimum. However, the objective function is not tied

[2] We use the rule-based Porter stemmer

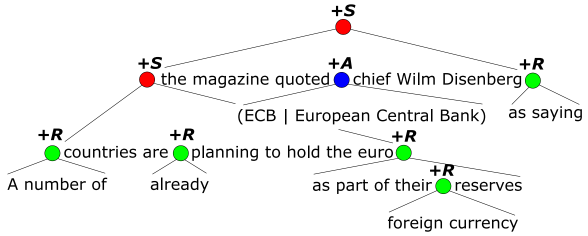
[3] Natural Language Toolkit (nltk.sourceforge.net)

to the quality of the alternative sentences, so much of the burden of linguistic quality falls on the compression module. Nevertheless, the constraints that enable sentence compression can be extended to allow for any kind of reformulation, such as the fusion of multiple sentences (which we did not implement).

We developed three types of sentence compression:

- Use of a subsentence clause as a new syntactic tree root (e.g. “he was here” in “he said he was here”)
- Removal of a syntactic subtree (e.g. a temporal clause or subordinate clauses)
- Alternative writing of a phrase (e.g. acronym definitions or co-references)

Sentence compression consists of two stages. First, a set of rules is applied to each node of a sentence’s syntactic tree in order to annotate it with subsentence (+S), removable (+R) and alternative groups (+A). The second step consists of recursively generating a list of candidates from this annotation. Figure 1 shows an example of a sentence annotated for compression and the resulting candidates. The rules for determining the +S, +R and +A annotation are described below.



** Countries are already planning to hold the euro as part of their reserves, the magazine quoted European Central Bank chief Wim Duisenberg as saying.*

** A number of countries are planning to hold the euro, the magazine quoted ECB chief Wim Duisenberg.*

** A number of countries are already planning to hold the euro as part of their foreign currency reserves.*

** A number of countries are planning to hold the euro.*

Figure 1: Compression candidates for a sentence from the DUC 2007 documents. A +S node can be used as a sentence root, +R nodes are independently removable, and sub-nodes in +A nodes can be used as alternatives.

Subsentence nodes (+S) consist of sentence clauses (S) formed from a noun phrase followed by a verb phrase excluding pronominal subjects. This filtering is designed to remove sentences without a verb and unresolved co-references. Short sentences such as “John said” are also pruned.

Removable nodes (+R) annotate temporal phrases, content in parentheses, subordinate clauses, adverbial clauses, and a subset of prepositional clauses. The rules are tuned to limit the risk of breaking semantic associations in the parse tree (such as structures containing comparatives or transitive verbs).

Alternatives (+A) are created by grouping noun phrases which occur frequently with the same head. For example, an apposition might be used instead of a name. Additionally, acronyms which are resolved in the source text are mapped to both their short and extended version at every occurrence (although they are handled differently in order to place the definition first once sentences are ordered). The generation of this mapping represents a primitive form of co-reference resolution.

Post-processing

For ICSI-2, we also implemented a dendrogram clustering approach to sentence ordering (Hickl et al., 2007). In general, sentences that share concepts are grouped together. The dendrogram is generated by iteratively grouping sentences (or clusters of sentences) which are most similar. As it does not give an ordering by itself, each pair of subtrees in the dendrogram is ordered so that the pair most similar to the average bag-of-concepts is picked first.

3.3 An Oracle System

Consider the following modification to ICSI-1: Change the mapping function so that the value of each n-gram is simply the number of different human-generated “gold” summaries in which it appears. Leaving the remainder of the system components unchanged, we have created a kind of oracle system. Specifically, this is a ROUGE oracle. Assuming we have such a set of gold summaries, the ROUGE-n score of a new summary is the average recall in word n-grams between this summary and those in the gold set. ROUGE, like machine translation’s BLEU, is subject to much criticism (nonsensical summaries can get high scores, for example), but when averaged over many problems, it tends to be very highly correlated with manual evaluation scores (Lin, 2004; Dang, 2006).

The oracle system is valuable because it suggests how close our best systems come to achieving the maximum ROUGE score. This subject is addressed by (Lin and Hovy, 2003), who find optimal ROUGE-1 summaries by exhaustive search for a single-document task, and (Conroy et al., 2006) who use a greedy search to find approximately optimal ROUGE-1 summaries for a multi-document task. Our oracle system, by virtue of the ILP framework, gives a tighter upper bound on ROUGE-n scores for extractive multi-document summarization. Experiments with the oracle can also help motivate system component choices, as shown in the next section.

Note that the official ROUGE toolkit[4] differs from our formulation in two ways:

1. Cross-sentence n-grams: These are unlikely to be matched and therefore an unnecessary complication to the ILP.

2. Non-linear weights for concepts: ROUGE penalizes n-grams appearing less frequently in a candidate summary than in a reference summary, but cuts the bonus for exceeding this frequency. This type of scoring is rather unjustified as it gives more points to redundant summaries. Moreover, this non-linearity would add significant complexity to the ILP formulation.

Because our model represents a simplification of ROUGE, the scores reported below are the output of the official ROUGE toolkit, as opposed to the ILP's objective function. In practice, however, the scores appear to be very similar.

4 Evaluation Results

4.1 Oracle Performance

Table 2 shows the performance of the oracle, using unigram, bigram, and SU4 gold concepts. SU4 includes both unigrams and "skip" n-grams – the endpoints of a word sequence spanning up to six words.

Perhaps the most interesting result here is the gap in ROUGE-2 score between the optimizations for gold unigrams and gold bigrams. Apparently, selecting sentences to maximize recall of important unigrams does not do a good job finding sentences with the important bigrams. And the reverse is not true: the bigram concepts appear to give results robust across all flavors of ROUGE.

Data	Concepts	R-1	R-2	R-SU4
A	unigrams	0.475	0.162	0.193
A	bigrams	0.463	0.199	0.212
A	SU4	0.470	0.191	0.219
B	unigrams	0.460	0.157	0.187
B	bigrams	0.453	0.200	0.212
B	SU4	0.461	0.192	0.217

Table 2: ROUGE-1, ROUGE-2, and ROUGE-SU4 results for the oracle system evaluated on the update (set B) and non-update (set A) portions of the TAC data set.

Also striking is the gap between system and oracle performance. Human summaries received ROUGE-2 scores between 0.110 and 0.132 on this data set, so humans are closer to the best systems than to the oracle, as measured by ROUGE.

Unfortunately, the oracle was not evaluated manually by human judges along with the other participants' systems. Perhaps future evaluations can include the results of such an oracle experiment.

4.2 System Performance

We show results for ICSI-1 and ICSI-2 and their ranks among all evaluated systems (1 is best; 58 is worst; the baseline system is included), for the non-update (set A) and update (set B) parts separately in Table 3. Figure 2 compares results, averaged over all topics, for all participating systems (over sets A and B merged). Note that teams were invited to submit up to three systems, but only primary and secondary systems were evaluated manually. As a result, there are more results given for ROUGE-2 in this figure.

5 Discussion

ICSI-1, which uses only simple frequency features, and very basic preprocessing and post-processing, performs very well. In particular, two-tailed t-tests show that for set A (the non-update part), ICSI-1 is either the best performing system or not significantly

[4] Details at: haydn.isi.edu/ROUGE/latest.html

Set A (non-update)				Set B (update)			
Metric	**ICSI-1 (R)**	**ICSI-2 (R)**	**Best**	**Metric**	**ICSI-1 (R)**	**ICSI-2 (R)**	**Best**
Resp	2.689 (9)	2.238 (28)	2.792	Resp	2.167 (22)	2.146 (23)	2.604
Ling	2.479 (21)	2.021 (47)	3.250	Ling	2.479 (24)	1.979 (45)	3.417
Pyr	0.345 (2)	0.324 (10)	0.362	Pyr	0.255 (16)	0.254 (19)	0.344
R-1	0.379 (5)	0.383 (4)	0.391	R-1	0.359 (11)	0.362 (9)	0.375
R-2	0.110 (2)	0.111 (1)	–	R-2	0.088 (11)	0.096 (3)	0.101
R-3	0.049 (1)	0.044 (2)	–	R-3	0.031 (13)	0.035 (12)	0.047
R-4	0.023 (1)	0.022 (3)	–	R-4	0.013 (41)	0.015 (29)	0.027
R-SU4	0.134 (4)	0.143 (1)	–	R-SU4	0.125 (13)	0.130 (6)	0.137
BE	0.063 (2)	0.064 (1)	–	BE	0.059 (14)	0.061 (9)	0.075

Table 3: ICSI-1 and ICSI-2 performance at TAC 2008 for the non-update (set A) and the update (set B) parts. The values in parentheses are ranks in [1, 58]. The performance gap between sets A and B is explained by the lack of specific processing for the update part.

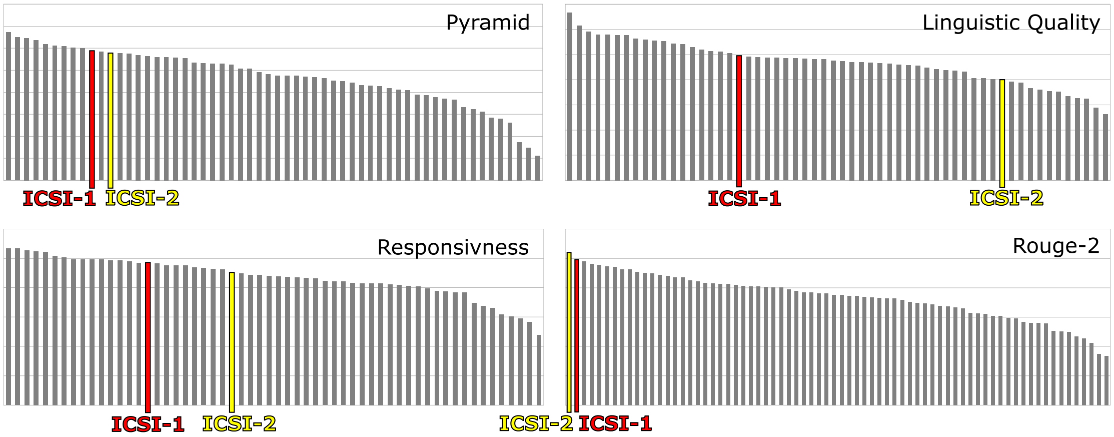


Figure 2: Ranks of ICSI-1 and ICSI-2 over all summaries (set AB). While compressed candidates in ICSI-2 boost ROUGE scores, linguistic quality and overall responsiveness decrease significantly.

different[5] from the best system in every category except linguistic quality. Our systems perform less well on the update set since we used the same system on both the standard and update parts, a result that demonstrates that specific processing to handle the update problem is valuable. A simple modification to ICSI-1 for dealing with the update task might involve adjusting the value of concepts based on their frequency in set A and set B.

ICSI-2 is a more experimental system. Intuition suggests that sentence compression ought to increase capacity for summary content, while decreasing linguistic quality. Table 4 compares performance of ICSI-1 and ICSI-2 across different evaluation metrics and Figure 3 shows a more detailed analysis of specific linguistic error types appearing in summaries that received low linguistic quality scores (of 1 or 2).

Pyramid	44.8%
SCUs	36.5%
Repetitions	22.9%
Linguistic Quality	16.7%
Overall Responsiveness	24.0%

Table 4: Percentage of topics for which ICSI-2 has higher scores than ICSI-1. Linguistic quality is much lower for ICSI-2 but the two systems have similar average Pyramid scores.

According to Figure 3, “ill-formed sentences”, the result of improper sentence compression, appear in nearly 65% of ICSI-2 summaries. As ICSI-1 uses

[5] with a 95% confidence interval

only full sentences from the source documents, this is the most striking difference between the two systems. “Bad ordering” refers to problems with sentence ordering. ICSI-2 summaries contain more sentences, which increases the chance for ordering inconsistencies. “Bad quotes” occur when a quote extending over multiple sentences is cut in the middle; “Garbage” refers to improperly handled formatting; “Unanswered questions” are questions asked in the summary that have no follow-up; “Dateline” refers to article headers that were not properly removed. All of these issues could be addressed with some careful preprocessing. “Unclear references” are typically pronouns but could be other noun phrases that are never introduced in a meaningful way. These are a major problem for both systems and full coreference resolution is probably needed to correct such mistakes. “No verb” refers to sentences without a main verb,“relative date” refers to clauses like “on Tuesday” that lack meaning in the context of a summary, and “bad discourse connector” refers to sentences that begin with “And” or “However” that do not make sense given the previous sentence. All of these errors are partially addressed by syntactic processing in ICSI-2. Note also that sentence segmentation failures appear in 20% of ICSI-1 summaries, suggesting that better segmentation is necessary. It is interesting to note the nearly 50% decrease in redundancy between ICSI-1 and ICSI-2, likely attributable to compressed sentences that convey more independent sets of concepts. Lastly, “other nonsense” is a catch-all category, indicating how much extra work will be required to manage a summary with short sentences.

Table 4 gives examples of grammatical nonsense sentences that are produced by the compression algorithm. This shows that sentence compression cannot rely only on syntactic information. For instance, if some fact is introduced and then negated later after some discussion, a compression module that removes the context will result in two adjacent sentences with contradictory information. More elaborate semantic models are required to deal with such situations.

Figure 5 shows summaries produced by ICSI-1 and ICSI-2 for one topic. In this example, compression leads to improved scores in all categories, though often we are not so lucky.

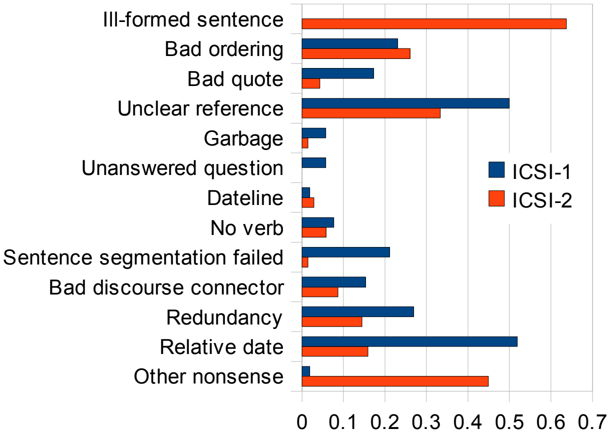


Figure 3: Repartition of error categories for summaries of a linguistic quality score of one or two.

* *A judge rejected the state Republican Party’s attempt to stop King County.*

* *The Democratic Party was about $100000.*

* *Miers said in a letter to Bush that she had withdrawn her nomination to protect the independence.*

* *The Baishuijiang State Nature Reserve is home.*

* *The plant has been a problem in areas.*

* *Earlier Thursday, a senior official told AFP.*

* *FARC is the largest insurgent group in Colombia, 000 combatants across the country.*

* *The Airbus official said he had not seen any sign.*

* *The plane’s engineers will begin to find out.*

Figure 4: Examples of grammatical but unclear or meaningless sentences that result from sentence compression.

5.1 Simplicity and Speed

ICSI-1, despite excellent performance, can be seen as a kind of baseline, one that is likely to give better results once many other important ideas from the summarization field are incorporated. The simplicity, however, makes it easy to understand and modify. It is also very fast. After sentence segmentation, the 96 TAC 2008 summaries were generated in 40 seconds (the ILP solver spent around 0.38 seconds per problem). By way of comparison, the 45 DUC 2007 summaries, with 2.5 times more input data and 250-word summaries, took 67 seconds (1.39 seconds per ILP problem).[6] If scalability becomes an issue, we note that a greedy search gives approximately 10% lower ROUGE scores than the ILP solu-

[6] Using a 2.1Ghz AMD Athlon

ICSI-1

In an interview, Yonadem Kana, the leader of the Assyrian Democratic Movement in Iraq and a member of the Iraqi National Council, said the fighters have been deployed in Baghdida near the northern city of Mosul. Christians make up just 3 percent of Iraq's population of about 25 million. Officials estimate that as manyas 15,000 of Iraq's nearly one million Christians have left the country since August, when four churches in Baghdad and one in Mosul were attacked in a coordinated series of car bombings. Most of Christians in Iraq are in Baghdad and northern cities.

ICSI-2

Officials estimate that as many as 15000 of Iraq's nearly one million Christians have left the country since August, when four churches in Baghdad and one in Mosul were attacked in a coordinated series of car bombings. Most of Christians in Iraq are in Baghdad and northern cities. Christians make up just 3 percent of Iraq's population of about 25 million. Armed men kidnapped a Catholic archbishop in Iraq's main northern city of Mosul Monday. In an interview, Yonadem Kana, the leader of the Assyrian Democratic Movement in Iraq and a member, said the fighters have been deployed in Baghdida.

Figure 5: Summaries generated for topic D0808-A: "Describe the events related to Christian minorities in Iraq and their current status". ICSI-1 scores: resp=3; ling=3; pyr=0.489. ICSI-2 scores: resp=4; ling=4; pyr=0.517.

tion. Approximate local search is an option as well. ICSI-2 requires running a parser during preprocessing, but the ILP phase is not significantly slower.

6 Conclusion

These systems mark ICSI's first participation in an organized evaluation for summarization. While there is much work to be done, our initial systems perform quite well. ICSI-1, in particular represents a solid starting point for anyone interested in building a new system. As a result, we offer a distribution of our code[7], an implementation of ICSI-1, complete with preprocessing and ROUGE evaluation for TAC and previous DUC data sets.

References

J. Carbonell and J. Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. *Research and Development in Information Retrieval*, pages 335–336.

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. 2006. Overview of DUC 2006. In *Proceedings of DUC'06 workshop*.

Andrew Hickl, Kirk Roberts, and Finley Lacatusu. 2007. Lcc's gistexter at duc 2007: Machine reading for update summarization. In *Proceedings of DUC'07 workshop*.

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.

Chin-Yew Lin and Eduard Hovy. 2003. The potential and limitations of automatic sentence extraction for summarization. In *HLT-NAACL Workshop on Text Summarization DUC'03*.

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.

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 2007 Document Understanding Conference (DUC-2007) at NLT/NAACL*.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of European Conference on Information Retrieval (ECIR 2006)*.

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.

Frank Schilder and Ravikumar Kondadadi. 2008. Fastsum: Fast and accurate query-based multi-document summarization. In *Proceedings of ACL-08: HLT, Short Papers*.

Wen-tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. 2007. Multi-document summarization by maximizing informative content-words. In *Proceedings of IJCAI*.

[7] www.dgillick.com/summarize