Event-Centric Summary Generation

Lucy Vanderwende, Michele Banko and Arul Menezes
One Microsoft Way
Redmond, WA 98052 USA
{lucyv,mbanko,arulm}@microsoft.com

Abstract

The Natural Language Processing Group at Microsoft Research participated in Task 2 of the Document Understanding Conference for the first time in 2004. Our primary interest is two-fold: 1) to explore an event-centric approach to summarization, 2) to explore a generation approach to summary realization.

1 Introduction

Our primary interest in participating in the Document Understanding Conference is to explore an event-centric approach to summarization. We ask ourselves whether identifying important events, as opposed to entities, would change the nature of the obtained summary in a significant way. In addition, we were also interested in testing our generation component within a new application. This decision was motivated in part by our observation that human-authored multi-document summaries tend to rely less on sentence extraction, and contain a significant amount of novel text compare to traditional single-document summaries (Banko and Vanderwende, 2004). While our system has not yet reached the stage of introducing terms which are not present in the documents to be summarized, we developed our summarizer with this future goal in mind.

In order to approach the problem in an event-centric manner, we use a graph-scoring algorithm to identify highly weighted nodes and relations in the graph that we construct by producing a dependency-style analysis for the documents in the document cluster. We use this scoring to guide content selection which then is presented to our generation component for realization. We focused exclusively on the graph-scoring algorithm and content selection and generation; we have not yet used any of the other features known to be useful when creating summaries, such as position in the article, publication date, coreference chains, or summarization keywords.

In the following sections, we first present a system description, followed by a discussion of our DUC submission and experiments we subsequently conducted in order to better understand our results. We finish with a discussion of future work.

2 System Description

The MSR-NLP summarizer uses both the analysis components and the generation components of the NLP system under development in Microsoft Research, NLPwin (Heidorn, 2000). The analysis components consist of a rule-based syntactic analysis component, followed by a component which produces a logical form analysis for each sentence in the document cluster. The logical form we use in this system is a dependency-style graph, which normalizes certain syntactic surface variations and in which the nodes are labelled with words from the text; the logical form as a dependency-graph is computed from an intermediate level of representation, language-neutral syntax (Campbell and Suzuki, 2002). The generation component is a syntactic realization component that takes a logical form structure as its input, produces an intermediate syntactic tree and subsequently a surface string. The generation component used in this system is rule-based (Aikawa et al., 2001); an alternative, machine-learned, method for creating a generation component which can take a logical form structure as input is described in Gamon et al, 2002. The generation component was designed to be a standalone component, but to date has only been applied in the context of a machine translation system. One of our goals in participating in DUC was to test the application independence of the generation component.

2.1 Creating Document Representations

From the sentences contained within a given set of documents, we build one graph representative of the

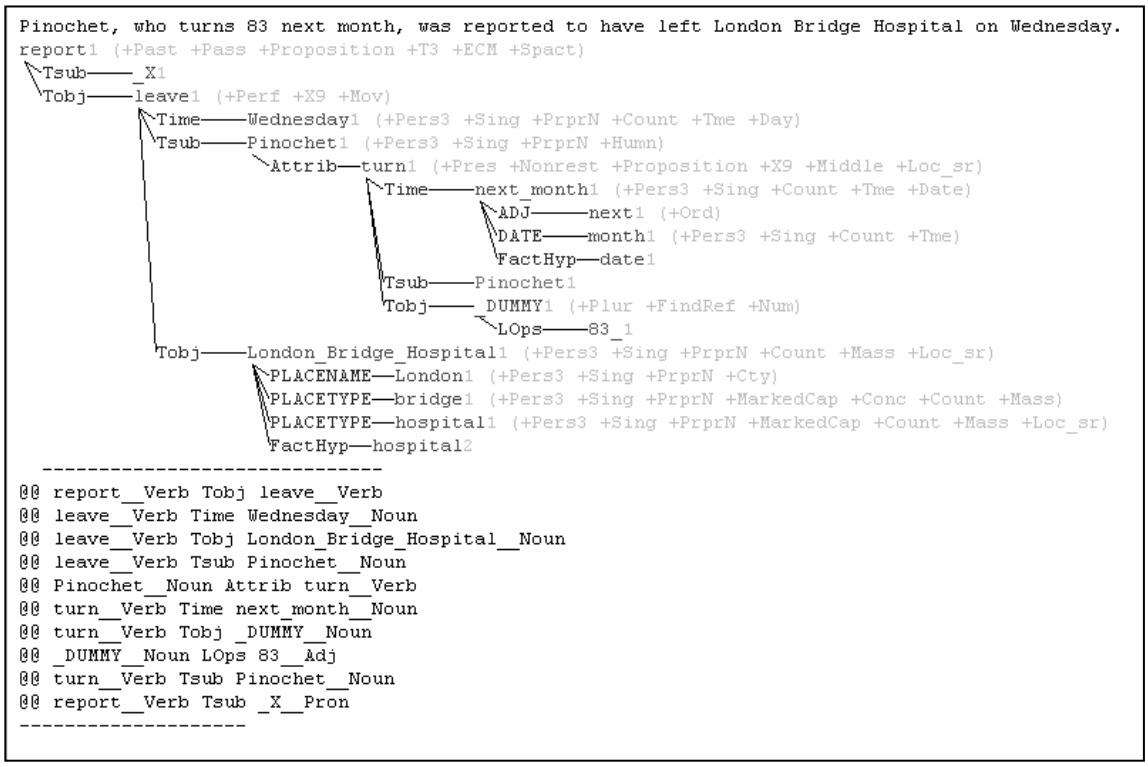

Figure 1: Gathering Logical Form Triples

entire cluster based on logical form analysis. Each sentence in the cluster is analyzed and its logical form obtained. During processing we remove all duplicate sentences and do not take their presence into account.

Our system produces a set of triples resulting from relationships detected between nodes in the logical form. These triples, examples of which can be seen in Figure 1, take the form, (*LFNode*$_i$, *rel*, *LFNode*$_j$). To form the document graph, we take these triples, and join nodes by way of their semantic relationships using a bidirectional link structure. Due to the nature of the algorithm we will use for scoring, we treat relationships as bidirectional links so that scoring mass can float freely between nodes. Additionally, when we observe a triple more than one time, we link between them at most once, but keep track of how many times we observe the relationship for when we later score fragments of the document graph. We do not include stopwords as part of our graph construction.

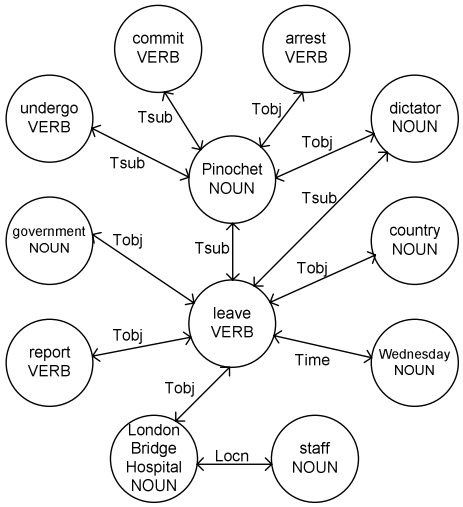

Figure 2: Document Graph Fragment

A piece of a graph built by our system for cluster d30003 (DUC 2003) is shown in Figure 2.

2.2 Node Scoring Using Pagerank

In order to compute scores for nodes in a document graph, our system uses the Pagerank algorithm (Brin and Page, 1998) commonly applied to a hyperlinked environment such as the worldwide web. The thrust of the pagerank algorithm is that when a node links to another node, it is casting a vote for that node. The more votes that are obtained by a node, the more importance it will receive. Additionally, the greater the importance of a node is, the more voting power it has. The formula for computing Pagerank of a node *n* is given as:

$$PR(n) = (1 - d) + d \sum_{li \in L} \frac{PR(l_i)}{C(l_i)}$$

where

L is the set of nodes linking into node n

$C(l)$ is the number of outgoing links coming from l

d is a dampening factor.

After initializing Pagerank scores uniformly, the algorithm is computed for several iterations, until little change in scores is observed. Typically the algorithm converges around 40 iterations.

In the context of our application, the "pages" of the Pagerank framework correspond to base forms of words found in the documents to be summarized, and the "hyperlinks" correspond to semantic relationships between words which have been detected at the logical form level.

The use of Pagerank enables us to identify entities participating in a variety of important events or relationships throughout the document cluster. For example, the more high-scoring events (verbs) an high-scoring entity (noun) is linked to, the more likely it is to be a focal point of a set of stories, and therefore a candidate to be discussed in a multi-document summary. In the partial example graph shown in Figure 2, *Pinochet* (100 inlinks) is assigned a high score by Pagerank (9.91), by way of its association with extremely high-ranking verbs *arrest* (30 inlinks, 3.15 Pagerank, most highly-ranked verb) and *commit* (24 inlinks, 3.12 Pagerank, third-most highly-ranked verb).

Using events to identify summary content is related to the use of verb specificity in Columbia's DEMS system (Schiffman et al., 2002). We think that there is an interesting difference, however, as verb specificity identifies verbs that are associated with only a few types of subjects, while Pagerank identifies interesting verbs based on all of its arguments, Also, verb specificity is based on a large corpus study, while the Pagerank score is computed for a particular document cluster.

2.3 Graph Scoring

Once we have obtained a score for each node in the document graph, we use these scores to assess the strength of links between nodes. As a linkweight between a node n and an inlinking node i is computed relative to other nodes which also point to n, these linkweights are not symmetrical with respect to direction.

A linkweight, $LW(i \rightarrow n)$ is computed as:

$$\frac{N(i \xrightarrow{rel} n) PR(i)}{\sum_{li \in L} N(l_i \xrightarrow{rel} n) * PR(l_i)}$$

where

L is the set of nodes linking into node n

$PR(i)$ is the Pagrank score of node i

$N(i \xrightarrow{rel} n)$ is the number of times we observe the relationship *rel* between i and n over the document set.

2.4 Summary Generation

Summaries are generated by extracting and merging portions of logical forms. The task of the summary generation component is to identify important and coherent logical form fragments, order and merge these fragments, and then generate summary sentences until the desired summary length has been reached.

We first identify important triples in the document cluster graph. These are defined as a highly weighted node (a node whose Pagerank score exceeds a threshold) together with the most highly weighted of its neighbors and the semantic relationship between them. For example, in Figure 2, if *leave* had a score above the threshold, and *London Bridge Hospital* was the highest scoring of its neighbors, then (*leave, Tobj, London_Bridge_Hospital*) would be an important triple, and furthermore (*leave, Tobj, government*) would *not* be marked as important.

We then scan the logical forms for each sentence in the document cluster to extract fragments. The starting point for each fragment is a triple that matches an important triple from the document cluster graph. We also include key arguments so as to make the fragment coherent, and we include nodes that indicate attribution (such as "said", "reported" etc).

We divide the extracted fragments into "event" and "entity" fragments. A subset of the event fragments are ordered and used to generate an initial summary. The entity fragments are then used to expand upon references to the same entity within the selected event fragments, as space permits. We chose to generate the initial summary using event fragments following our goal to explore an event-centric approach to summarization.

We cluster the event fragments by the event they refer to. For each event we select the most informative fragment. Currently we do this by choosing the fragment with the greatest number of argument nodes for that event.

Next, we order the selected event fragments. At present, ordering consists of two operations: the first is to group sentences referring to the same entity together, and the second is to order sentences which exhibit event-coreference such that the event precedes the referring expression; at present, this is limited to lemmas which have both noun and verb parts of speech, where we order the verb first.

	Baseline			System		
	Average	Min	Max	Average	Min	Max
ROUGE1	0.31748	0.29920	0.33330	0.34148	0.32832	0.35451
ROUGE2	0.06871	0.05221	0.07941	0.05849	0.04786	0.06699

Table 1: Baseline vs. System Results for DUC 2003

An initial summary is then generated from the ordered selection of event fragments. As each fragment is generated to realize a text string, the byte-length for each generated string is computed; generation is halted when the combined byte-length of the candidate summary sentences exceeds the desired summary length. Instead of truncating the last sentence, we try to utilize this space more creatively by expanding entities

We use the extracted entity fragments to expand upon entity references in the initial summary. This is done at the logical form level -- we expand an entity to produce a merged logical form, regenerate the whole sentence and recompute the summary length. This is repeated until we reach the summary size limit or have run out of possible expansions.

3 Experiments and Evaluation

3.1 Tools and Test Corpora

In order to test our system, we used ROUGE, (Lin and Hovy, 2003), an automatic method for evaluating automatic summaries against a set of human-authored summaries. This recall-based method, which measures n-gram co-occurrence between summary pairs, was found to correlate highly with human judgments at the unigram level, but weakly using n-grams with n larger than 1. As a result, we focused mostly on unigram ROUGE scores (referred to as ROUGE1), and to a lesser extent on bigram ROUGE scores (ROUGE2). We did not look closely at larger sized n-grams, even though those statistics were available to us.

As for corpora, we chose to work with test documents made available for Task 2 of DUC 2003. This data set is made up of 30 document clusters, each consisting of about 10 documents. For each cluster, 4 distinct human summaries have been provided, which we used as gold standards within our evaluation framework.

3.2 Experiments with DUC 2003

Table 1 shows our system results relative for lead baseline, which takes as a summary, the first *n* bytes of the most recent document in the cluster. Our system gained 7.6% in ROUGE-1 score relative to the baseline.

Anaphora Resolution

The NLPwin system contains a simple rule-based pronoun resolution method which achieves about 75% accuracy on a Wall Street Journal corpus (Ge et al. 1998). This method is a rule-based system which resolves pronominal anaphora based on a system of weighted preferences. After using heuristic rules to determine which personal pronouns are referential and which are pleonastic, the system attempts to find the most likely antecedent for referential pronouns from the current sentence and preceding sentences. As in Lappin and Leass (1994), each candidate antecedent is weighted according to a number of factors, including whether it precedes or follows the anaphor, distance between it and the anaphor, agreement, grammatical function, whether it and the anaphor have a parallel grammatical function, and whether it has been mentioned before and how often.

To assess the effect of using an automatic anaphora resolution component in our system, we ran our summarizer both with and without automatic resolution. We found that with anaphora resolution, 15 out of the 30 clusters saw an improvement in Average ROUGE-1 score, ranging from 0.009 to 0.106. We did not observe a change in ROUGE score for 11 clusters, and saw slight decreases for the remaining 4 clusters. On the whole, the use of automatic anaphora resolution improved our system by 3.3%. Aggregate results for 2003 data are shown in Table 2.

	No Anaphora Resolution	Anaphora Resolution	Relative Gain
ROUGE-1 Average	0.3305	0.3415	3.31%
ROUGE-2 Average	0.0522	0.0585	12.0%

Table 2: Effect of Anaphora Resolution

The Effect of Sentence Generation

In order to examine the effect of using sentence generation as opposed to sentence extraction during the formulation of summaries, we implemented a simple sentence-extraction component. The extractor uses Pagerank scores to rank sentences accoring to how much of the total Pagerank mass is contained within a given sentence. We select sentences containing the highest Pagerank density, normalizing for length so as to not automatically favor long sentences, until we fill up the total amount of space that has been alloted for a summary. While our system already elminates duplicate

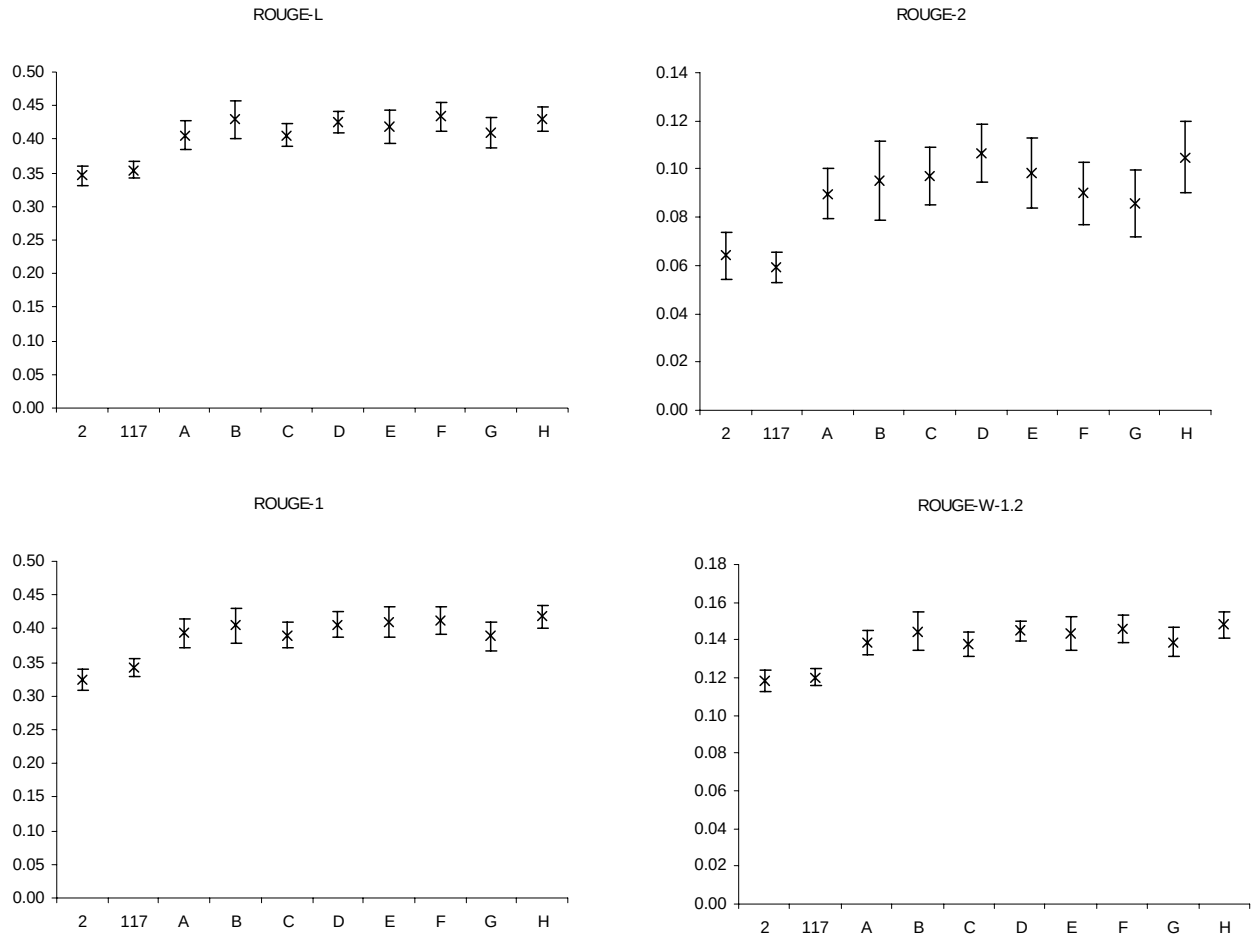

Figure 3: DUC 2004 Results

sentences from consideration, our extraction component did not handle the possibility of choosing highly-redundant sentences in any special way. However upon manual inspection of the output, we did not detect this to be a significant problem.

As shown in Table 3, sentence generation performs better than sentence extraction at the unigram level, and slightly less well at the bigram level. This may be due to the potential to introduce disfluent text during sentence generation, something which cannot happen using purely sentence extraction.

	Sentence Extraction	Sentence Generation
ROUGE-1 Average	0.3317	0.3415
ROUGE-2 Average	0.0676	0.0585

Table 3: Sentence Extraction vs. Generation

3.3 DUC 2004 Assessment

We participated for the first time in Task 2 of the 2004 Document Understanding Conference. The focus of this year's evaluation of short multi-document summaries was on the ROUGE metric, as opposed to previous years, where the primary assessments were peformed by human evaluators who judged summaries for coherence, cohesion, and grammaticality.[1]

In addition to reporting ROUGE-n scores for n=1 through 4, the 2004 evaluation also includes ROUGE-L, which measures longest common subsequence shared between reference and candidate summaries, and ROUGE-W, which similarly looks at weighted longest common subsequences (Lin, 2004).

Figure 3 depicts the performance of our system (id 117) for ROUGE-1, ROUGE-2, ROUGE-L and

[1] Human evaluation was later performed in 2004 for Task 2 in order to provide more data for assessing the power of human vs. automatic evaluation. However, full manual results were not available at the time of publication.

ROUGE-W metrics. This is shown relative to baseline (id 2) and human performance (ids A-H) on the task.

4 Directions and Future Work

We are pleased with the results of our first system submitted to the Document Understanding Conference, but of course, we have many areas that we wish to explore further.

We intend to further explore whether we can develop a system to produce more human-like generated summaries with respect to the introduction of novel text (Banko and Vanderwende, 2004). In order to do so, we will continue to study the nature of human-generated summaries. We thank NIST for making the summaries available in DUC. We also intend to study other types of summaries, though such material is difficult to come by.

We also intend to further study the impact of anaphora resolution. We are in the process of manually annotating the 30 DUC 2003 clusters, which we intend to use to emulate a system with perfect anaphora. Initial results indicate that we will need to change our graph representation to better reflect entities as units rather than complex structures.

So far, we have seen that the generation component improves our results over a simple extractive method. We also intend to examine whether the concept scoring that PageRank provides shows similar improvements over a scoring based on term frequency, and whether a different graph-ranking algorithm, such as used in Mani and Bloedorn, 1997, might be more appropppriate.

Finally, while the generation component itself proves to be application neutral, as we had hoped, the content selection and the nascent planning component warrants further exploration. Currently, our system uses a completely event-centric approach to content selection, and NP-expansion is only used to fill the space allotted for task 2. We will continue to experiment with different trade-offs between the event and entity logical form fragments that are given to the generation component. In addition, while ordering groups event fragments mentioning the same entity, we have not yet implemented a system to combine these fragments into larger logical form constructions.

Acknowledgments

We would like to thank Rich Campbell and Hisami Suzuki for providing an enviroment in which we could experiment with anaphora resolution.

References

Takako Aikawa, Maite Melero, Lee Schwartz, and Andi Wu. 2001. Multilingual Sentence Generation. In Proceedings of *8th European Workshop on Natural Language Generation*, Toulouse.

Michele Banko and Lucy Vanderwende. 2004. Using n-grams to understand the nature of summaries. In *Proceedings of the North American Association for Computational Linguistics*.

Sergey Brin and Larry Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*. Volume 30:1-7. pp. 107-117.

Richard Campbell, and Hisami Suzuki. 2002. Language-Neutral Syntax: An Overview. Microsoft Research Technical Report: MSR-TR-2002-76.

Michael Gamon, Eric Ringger, and Simon Corston-Oliver. 2002. Amalgam: A machine-learned generation module. Microsoft Research Technical Report: MSR-TR-2002-57.

Niyu Ge, John Hale and Eugene Charniak. 1998. A statistic approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4), 535-561.

Chin-Yew Lin and E.H. Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of 2003 Language Technology Conference*, Edmonton, Canada.

Chin-Yew Lin. 2004. ROUGE Working Note.

George E Heidorn, Intelligent Writing Assistance, in *Handbook of Natural Language Processing*, Eds. Robert Dale, Herman Moisl, and Harold Somers, Marcel Dekker, 2000.

Inderjeet Mani and Eric Bloedorn, 1997. Multi-document Summarization by Graph Search and Matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence* , 622-628.

Barry Schiffman, Ani Nenkova, and Kathleen McKeown, Experiments in Multidocument Summarization. In *Proceedings of HLT 2002 Human Language Technology Conference*, San Diego, CA, 2002.