Automated Summarization Evaluation with Basic Elements

Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto[1]

Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
{hovy, cyl, liangz}@isi.edu

Abstract

As part of evaluating a summary automatically, it is usual to determine how much of the contents of one or more human-produced 'ideal' summaries it contains. Previous automated methods such as ROUGE compare using fixed word ngrams, which are not ideal for a variety of reasons. In this paper we describe a framework in which summary evaluation measures can be instantiated and compared, and we implement a specific evaluation method using very small units of content, called Basic Elements, that address some of the shortcomings of ngrams. This method is tested on DUC 2003, 2004, and 2005 systems and produces very good correlations with human judgments.

1. Introduction

Experience in Machine Translation and automated speech recognition has shown the great value of an automated evaluation measure for rapid system growth and improvement (Papineni et al., 2001). The text summarization community has also searched for automatic summary evaluation methods that produce reliable scores that correlate well with human scoring. When measuring the content of a summary, current automated methods compare fragments of the summary to be scored against one or more reference summaries (typically produced by humans). The more desirable fragments the summary contains, the better it is considered.

Choosing an appropriate fragment length, and comparing it appropriately, are two problems that have not yet been satisfactorily solved. One can of course address both problems by having humans bracket the fragments to be evaluated and then manually compare fragments with the content of ideal summaries (Nenkova and Passonneau, 2004; Van Halteren and Teufel, 2003). But doing so introduces human variability and is typically prohibitively expensive in time and cost.

In this paper we describe framework in which various automated summary content evaluation methods can be situated, and we implement a specific variant that uses rather short fragments we call Basic Elements (BEs).

2. A Framework for Automated Summary Evaluation

This section describes an overall framework in which various implementations of automated summary content evaluation methods can be housed and compared. The framework, which we called the BE Package, is available without restriction at http://www.isi.edu/~cyl/BE/. The BE Package provides for three principal modules: **BE breakers** (that create individual BE units, given a text), **BE matchers** (that rate the similarity of any two BE units), and **BE scorers** (that assign a score to each BE unit individually).

2.1. The BE Procedure

The problem of evaluating the content of a given summary breaks into three distinct sub-problems, corresponding to the first three modules listed above. As input, the BE Package takes the summary to be scored as well as a set of ideal (reference) summaries. It applies the modules twice, in two phases: preparation and scoring. In the preparation phase, the first module breaks up the reference summaries into a list of reference BEs; the second module considers all reference BEs and merges semantically identical ones; and the third module assigns a score to each of the reference BEs. In the second (scoring) phase, the first module breaks up the summary to be scored into a separate list of BEs; the second compares each BE to the list of reference BEs; the third assigns a score to each BE to be rated and computes the final overall score of all the BEs contained in the summary to be rated. (Of course, the first phase is not repeated for multiple use of the same reference summaries.)

2.2. Prior Work on Matching and Defining Fragments

As stated previously, comparing sentences is too coarse-grained because they contain many individual pieces of information, which may not be used by humans for reference summaries. The question becomes what level of granularity is appropriate for automatic summary content comparison. ROUGE (Lin and Hovy 2003), the most frequently used automated summary evaluation package, is closely modeled after BLEU for MT evaluation (Papineni et al. 2001). It uses ngrams of various lengths, a total of 17 different parameterizations, as the fragments for comparison. Through shown to correlate well with human judgments, ROUGE considers fragments, of various lengths, to be equally important, a factor that rewards low-informativeness fragments, such

[1] This author was on sabbatical while this work was performed. Usual address: Department of Media Technology, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu-shi, Shiga 525-8577, Japan; email fukumoto@cs.ritsumei.ac.jp.

as "of the", unfairly to relative high-informativeness ones, such as person names. In addition, the best ROUGE parameterization varies with tasks, thus making it difficult to settle on a particular and consistent parameterization for all summarization tasks.

Since the introduction of ROUGE, several work have emerged to address these problems. Lin and Demner-Fushman (2005) recently developed POURPRE, in which fragments are given an intrinsic score based on their innate informativeness, computed by measuring the information content of individual words. They show an increased correlation with human summary evaluation scores, using a somewhat nonstandard measure. There is work where humans define what constitutes content fragments. These fragments are single coherent semantic units, such as "United States of America", "coffee mug", "the/a plane landed", "the landing was safe", etc. Van Halteren and Teufel (2003), the "factoids" work, were the first to take up the idea seriously, and showed that the amount of human variability in the method makes a fairly large number of references necessary to achieve score stability. The idea was taken further by Nenkova et al. (2005), who named fragments Summary Content Units (SCUs), and deployed them in the Pyramid method (see later). Here a single person delimited the reference SCUs, and one or more people then matched the summaries to be rated against them.

The idea of semantic fragments has also been pursued by other applications. Riezler et al. (2005) use them to evaluate the effectiveness of single-sentence condensation (compression). They condense by parsing sentences into LFG structures and then dropping selected portions of the LFG using rules trained using a Maximim Entropy model. To evaluate, they extract (relation–predicate–argument) triples from the LFG structures of the condensed and reference sentences and count the overlap. Similarly in MT, Mohanty et al. (2005) decompose sentences into small fragments and then translate the fragments individually, seeking to retain not only word equivalence but also syntactic relations.

With the development of the BE framework, we want to address the following questions: Can one automatically produce fragments of appropriate size? What is the most appropriate size? What are the criteria for bracketing fragments?

2.3. Basic Elements (BEs)

In this approach, we break down each reference sentence into a set of minimal semantic units, which we call **Basic Elements (BEs)**. After some experimentation, we have decided to define BEs as follows:

- the head of a major syntactic constituent (noun, verb, adjective or adverbial phrases), expressed as a single item, or
- a relation between a head-BE and a single dependent, expressed as a triple (head | modifier | relation).

Starting small like this allows one to automate the process of unit identification and, to some degree, facilitates the matching of different equivalent expressions.

2.4. Creating Units: The BE Breaker Module

We implemented and experimented with various implementations for creating BEs—BE Breakers:

- BE-L: Charniak parser (constituency tree) + CYL cutting rules
- BE-F: Minipar (dependency tree, with relations) + JF cutting rules
- Chunker: syntactic-unit chunker that includes cutting rules.
- Microsoft parser[2] (Heidorn, 2000) + cutting rules

Each breaker accepts a sentence as input and produces a list of BEs by decomposing parse trees using hand-built 'cutting rules'. These breakers produce slightly different results (the common overlap is approximately 40%). In particular, some breakers provide relations as part of the triples and others do not.

BE-F: BE-F extracts BEs from Minipar (Lin, 1995) dependency parse trees in which word-relations are labeled as *subj* (subject), *obj* (object), *comp1* (complement), *mod* (modifier), etc. Word pairs with their dependency relation are extracted to form a BE.

Processing Minipar parse information involves converting compound nouns and verbal idioms, such as 'turn over' and 'Secretary General', to single tree nodes. BE-F reifies embedded tentative nodes that express semantic subject or object with semantic nodes and performs extraction. For a propositional phrase, the head is related to its governing element by its preposition (e.g., 'sanction against Libya' produces a BE [sanction | Libya | against]). For embedded clauses, main verbs are related to the modifying verbs. If there is no subject, the semantic subject and the main verb form a BE with 'subject' as the relation.

2.5. Scoring Units

In the present implementation, each BE gets exactly 1 point for each reference summary it participates in. This score is weighted depending on the completeness of the match between the BE and the reference BEs, as described immediately below. We have not experimented with different weights based on words' information content, etc., although one can obviously do so.

2.6. Comparing and Matching Units

We categorize matching strategies into several classes (from easiest to most difficult):

- lexical identity: words must match exactly.
- lemma identity: the root forms of words must match
- synonym identity: words or any of their synonyms, identified by WordNet (Miller et al., 1990), must match.
- (an approximation) phrasal paraphrases must match.
- semantic generalization: words make up BEs are replaced by their semantic generalizations ("Mother Theresa" replaced by "human") and then matched at a variety of abstraction levels.

Sophisticated matching strategies should be able to recognize full or partial semantic equivalences ("approximately $20 million" and "19.8 million dollars"),

[2] We are indebted to Lucy Vanderwende and her group at Microsoft for parsing our test collection and allowing us to experiment with the results.

anaphoric coreferences ("he said" and "Joe said"), abbreviations, and metonymy ("Washington announced" and "The US government announced"). However, these methods demand a much higher level of language interpretation and understanding, and require many component technologies that are not current available.

We implemented the matching of lexical identity for BEs. The resulting matching algorithm is less difficult than one that matches unstructured phrases because the definition of BEs allows less chaos in phrase extraction.

3. Judgment Correlation with DUC 2005

A good automatic summarization evaluation procedure should be able to differentiate good systems from bad ones. In practice, the evaluation procedure is given a set of system-generated summaries and human-written reference summaries, and is required to provide a rank for the systems that created the summaries. To examine the validity of our method, we have tested the BE framework and its current implementation thoroughly using previous DUC evaluation results, namely DUC2002 and 2003, on single- and multi-document summarization tasks, and very short headline generation task. Another important property that a good automated evaluation procedure possesses is that it must show good and consistent correlation across evaluations of different summarization tasks. DUC2005 is the first time that query-based summarization has been performed on a large scale. 32 automatic summarization systems participated to create question-focused summaries by answering a list of complicated questions from sets of 25-50 texts. 50 sets of texts were used in the task. For each document set, 4 human-written summaries were provided as references.

3.1. Correlation: BE vs. Responsiveness

NIST computed the average scaled responsiveness score (from human assessors) of each summarizer across all topics. To validate BE, we computed the Spearman rank coefficient and Pearson coefficient between BE and responsiveness scores. Two variations of BE are experimented. HM is set for (head-word | modifier). HMR is set for (head-word | modifier | relation). A high correlation is found, as shown in Table 1.

	HM	HMR
Spearman	0.928	0.926
Pearson	0.975	0.976

Table 1. Correlation b/w BE and responsiveness.

3.2. Correlation: BE vs. ROUGE

Table 2 shows the correlation between BE and ROUGE, a widely used and recognized automated summarization evaluation method (Lin and Hovy, 2003). The ROUGE scores are macro-averaged by NIST.

3.3. Correlation Overview

Figure 1 shows the overall correlation between ROUGE, BE, responsiveness, and the Pyramid method computed on those human- and system-generated summaries included in the Pyramid annotation effort (only 20 doc sets and 25 automatic systems were included). The label on each link indicates the parameterization, including the Spearman rank coefficient and the Pearson coefficient between the systems connected by the link respectively. The correlations between the Pyramid method and NIST responsiveness and ROUGE respectively are taken from (Passonneau et al., 2005).

	BE.HM		BE.HMR	
	Spearman	Pearson	Spearman	Pearson
rouge.1	0.953	0.924	0.944	0.923
rouge.2	0.965	0.970	0.964	0.971
rouge.3	0.928	0.956	0.933	0.958
rouge.4	0.882	0.897	0.889	0.900
rouge.L	0.943	0.918	0.936	0.917
rouge.SU4	0.959	0.951	0.954	0.951
rouge.W-1.2	0.947	0.927	0.940	0.926

Table 2. Correlation b/w BE and macroavg ROUGE.

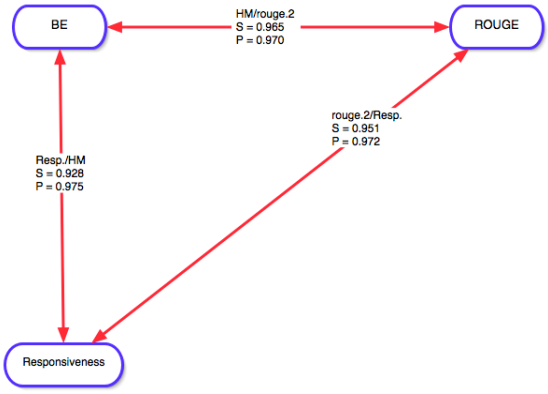

Figure 1. Correlation overview of ROUGE, BE, responsiveness, and Pyramid.

4. Conclusions and Future Work

The most pressing problem remaining is developing powerful BE matching routines; if one can match minimal BEs (and paraphrases) accurately then building matchers for compound BEs should be an interesting but not impossibly difficult exercise. Similarly, determining optimal weighting functions for individual BEs and for their combination to maximize correlations with human judgments requires careful but not impossibly hard work, and resembles the work recently done by Lin on ROUGE.

Finally, it is of particular interest to see whether one can reconstitute within the BE framework an exact automated version of the factoid work of Van Halteren and Teufel and the pyramid method of Nenkova and Passonneau.

5. References

DUC. 2001–2005. The series of Document Understanding Conference proceedings.

Lin, C.-Y. and E.H. Hovy. 2002. Manual and Automatic Evaluation of Summaries. *Proceedings of the Document Understanding Conference Workshop at Conference of the ACL (DUC-02).* Philadelphia, PA.

Lin, C.-Y. and E.H. Hovy. 2003. Automatic Evaluation of Summaries using n-gram Co-occurrence Statistics. *Proceedings of the HLT-NAACL conference.* Edmonton, Canada.

Lin, D. 1995. A Dependency-based Method for Evaluating Broad-Coverage Parsers. *Proceedings of IJCAI-95*.

Nenkova, A. and R. Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. *Proceedings of the HLT-NAACL conference*. Boston, MA.

Papineni, K., S. Roukos, T. Ward, and W. Zhu. 2001. Bleu: A Method for Automatic Evaluation of Machine Translation. *Proceedings of the conference of the Association for Computational Linguistics (ACL)*, 311–318, Philadelphia, PA.

Passonneau, R.J., A. Nenkova, K. McKeown, S. Sigelman. Applying the Pyramid Method in DUC 2005. In *DUC 2005 workshop*.

Van Halteren, H. and S. Teufel. 2003. Examining the Consensus between Human Summaries: Initial Experiments with Factoid Analysis. *Proceedings of the HLT-NAACL Workshop on Automatic Summarization*. Edmonton, Canada.