# Evaluating DUC 2005 using Basic Elements

**Eduard Hovy, Chin-Yew Lin, and Liang Zhou**
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
{hovy, cyl, liangz}@isi.edu

## Abstract

In this paper we introduce Basic Elements, a new way of automating the evaluation of text summaries. We show that this method correlates better with human judgments than any other automated procedure to date, and overcomes the subjectivity/variability problems of manual methods that require humans to preprocess summaries to be evaluated. This is demonstrated on DUC 2005 peer systems and peer-produced summaries.

## 1 Introduction

When a language technology area develops a standard corpus on which to develop its techniques plus an automated method of evaluating results to drive development, progress tends to speed up dramatically.

It has long been a goal of the text summarization community to find automatic methods for summary evaluation that produce reliable and stable scores. All automated methods today work by comparing the system's summary to one of more reference summaries (ideally, produced by humans). But even leaving aside the problems of evaluating style and focusing just on summary content, evaluation has proven problematic. Experience has shown that measuring content at sentence granularity is not precise enough: generally sentences contain too many distinct pieces of information, some of which may be important to include in a summary and some of which not. However, comparing content at the word level is also not satisfactory: a word taken individually cannot be rated since it may play roles of various importance in various contexts with other words.

People have tried to overcome this problem by manually bracketing in the summary to be evaluated units of varying lengths that contain just the 'important' information, which a subsequent procedure (manual or automatic) can then rate. But this approach introduces human variability and typically incurs a significant effort.

In this paper we describe an automated method to produce useful chunks of varying size, and show that this method equals other automated methods of scoring summaries. We introduce Basic Elements (BEs) in Section 2. Section 3 provides details and Section 4 the results of evaluation by BEs on DUC 2005. Conclusions and some future work are given in Section 5.

## 2 Basic Elements

### 2.1 Definition

We address the problem of unit size by automatically producing a series of increasingly larger units, starting at the single word level. Unit importance can be determined by a variety of automated ways, described below. We address the problem of unit comparability by focusing on small-sized units, where paraphrase alternatives are rather limited.

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. Grouping smaller units into larger ones can be done automatically, even, we believe, to the larger-sized chunks typically used in the Pyramid Method.

As described below, we can produce BEs automatically in several ways Most of them involve a syntactic parser to produce a parse tree and a set of 'cutting rules' to extract just the valid BEs from the tree.

With units of minimal length, one can much more easily decide whether any two units match (express the same meaning) or not. For instance, “United Nations”, “UN”, and “UNO” can be matched at this level (but require work to isolate within a longer unit or a sentence), allowing any larger unit encompassing this to accept any of the three variants. Also, since the units are matched at the lowest levels, the danger of potentially double-counting segments that are contained in longer ones can also be avoided.

### 2.2 The Four Aspects of BEs

In order to implement Basic Elements as a method of evaluating summary content, four core questions must be addressed:

1. What or how large is a Basic Element? The answer to this is strongly conditioned by: How can BEs be created automatically?
2. How important is each BE? What basic score should each BE have?
3. When do two BEs match? What kinds of matches should be implemented, and how?
4. How should an overall summary score be derived from the individual matched BEs’ scores?

Different answers to each of these questions provide a different summary evaluation method. The Pyramid Method, for example, takes as BEs approximately clause-length semantic units shared by the reference summaries; gives each unit a score equal to the number of reference summaries containing it; allows two units to match when they express all or most of the same semantic content, as judged by the assessors; and derives the overall score by summing the scores of each unit of the candidate summary and normalized by the overall score of an ideal summary of equal size. In contrast, ROUGE uses as BEs various ngrams (for example, unigrams); scores each unigram by a function that depends on the number of reference summaries containing that unigram; allows unigrams to match under various parameterizable conditions (for example, exact match only, or root form match); and derives the overall summary score by some weighted combination function of unigram matches.

The BE Package is an overall framework in which various solutions to the four core aspects are implemented, and which therefore serves as a generalization over the particular methods and as an environment in which they can be compared.

There are multiple possible approaches to implementing each of these four points in software. Therefore, exploring the whole space in order to find the most stable and optimum evaluation configuration is not a trivial task. The current BE Package (Section 6) provides several parameterized modules as well as APIs for people wishing to build and test their own. Used as provided, the BE Package provides several implementations of the ideas of Van Halteren, Teufel, Nenkova, and Passonneau. We have performed a series of experiments to obtain reasonably good modules and parameter settings, but welcome additional studies and improvements.

## 3 The BE Method

The BE Package contains instances of four principal modules: the BE Breakers (that create individual BE units, given a text), the BE Scorers (that assign scores to each BE unit individually), the BE Matcher modules (that rate the similarity of any two BE units), and the BE Score Integrators (that produce a total score given a list of rated BE units). We have implemented and tested versions of each of these modules, and in some cases multiple versions. After outlining the procedure we discuss each module in turn.

### 3.1 The BE Procedure

The task is to provide a numeric score that reflects the quality of a given summary. Input to the BE Package is a set of reference (gold standard) summaries and the given summary to be rated. The Package first creates a set of BEs for each reference summary and integrates the sets by matching the individual BEs to obtain a single list of rated BEs, ranked from most valuable to least. (This first step needs of course not be repeated for subsequent use of the same reference summaries.) The summary to be evaluated is then also subjected to the BE breaker, and its BEs are matched against the ranked BE list. Matched BEs’ scores are integrated and the resulting score is returned.

### 3.2 Creating Units: The BE Breaker Module

The BE Breaker accepts a sentence as input and produces a list of BEs as output. Different BE Breakers produce different BEs. The intuition behind a BE is that it should be a single coherent semantic unit, such as “United States of America”, “coffee mug”, “the/a plane landed”, “the landing was safe”, etc. There is much room for discussion here, but the basic desiderata are: small size (to allow proper scoring of atomic bits of content), regularity/simplicity of definition, and automatic production.

As mentioned above, some experimentation led us to (for the present) define BEs as follows:

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

Some BE breaker modules provide the relations; others do not.

We presently produce BEs in several ways. Most of them involve a syntactic parser to produce a parse tree and a set of ‘cutting rules' to extract just the valid BEs from the tree. We have built and experimented with the following BE breakers:

- Charniak parser + CYL rules (abbrev. as BE-L)
- Collins parser + LZ rules (abbrev. as BE-Z)
- Minipar + JF rules (abbrev. as BE-F)
- Microsoft parser[1] (Heidorn, 2000) + cutting rules

**BE-F**: BE-F extracts BEs from parses generated by Minipar (Lin, 1995). Minipar produces a dependency parse of a sentence, in which each word is related to another word with labels such as *subj* (subject), *obj* (object), *comp1* (complement), *mod* (modifier), etc. BE-F extracts word pairs in some dependency relationship and generates a BE element.

During preprocessing for BE-F, compound nouns and verbal idioms such as 'turn over' and ‘Secretary General' are converted to single parse tree nodes using Minipar information. After parsing, BE-F reifies embedded tentative nodes that express semantic subject or object with semantic nodes. BE-F then extracts BEs from dependency relationships in the parse tree. For PPs, the PP head is related by its preposition to its governing element (e.g., for ‘against Libya’ modifying the word ‘sanction’, the extracted BE will be [sanction | Libya | against]). In an embedded clause such as ‘that clause’, the main verb of the embedded clause has a relationship with the modifying verb. If there is no subject word in such an embedded clause, its semantic subject and its main verb form a BE with the relationship ‘subject’ (and similarly for the case of ‘object’).

**BE-L**: BE-L is based on constituency parse trees generated by Charniak’s statistical parser (Charniak 2000). Converting a constituency parse tree into dependency triples is not new; Lin (1995) proposed a method based on Magerman’s (1994) head finding rules. A similar approach was used in Collins’s PCFG parser (1999). We followed the same approach, applying Magerman’s head finding rule to extract head-modifier dependencies. In order to identify the semantic relation between head and modifier, we trained a semantic role labeler using SVM based on PropBank data. Due to space limitations, we cannot provide details about the semantic labeler. However, its accuracy on the core argument identification (ARG0-5) was at about 93%. Examples from BE-L are shown below:

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

“Two Libyans were indicted for the Lockerbie bombing in 1991.”

```
<Libyans|two|CARDINAL>
<indicted|Libyans|ACCUSED>
<indicted|bombing|CRIME>
<indicted|1991|TIME>
```

Although BE-L achieves performance similar to BE-F and includes semantic labels for relations, it is much slower than BE-F. The speed bottleneck is mainly due to Charniak’s parser and the extra time needed to carry out semantic labeling of head-modifier relations.

**BE-Z**: BE-Z applies the Collins parser (Collins, 1999) and then a different set of cutting rules.

All three sets of cutting rules were developed independently, and hence the three engines produce somewhat different results. No single engine always produces the best results, since different parsers and rules operate differently well on different sentence constructions (especially with long sentences). We have experimented with combining the BEs from the different engines into a single list, but for simplicity of development and eventual distribution (different parsers have different license requirements) we decided to work with engines individually for now.

Example BEs for “two Libyans were indicted for the Lockerbie bombing in 1991” are as follows, written as (head | modifier | relation):

```
Libyans|NIL|NIL                 (BE-L)
Libyans|two|NIL                 (BE-L)
bombing|NIL|NIL         (BE-L)
bombing|Lockerbie|NIL           (BE-L)
indicted|NIL|NIL                (BE-L)
indicted|Libyans|ARG1           (BE-L)
indicted|1991|ARGM-TMP          (BE-L)
indicted|bombing|ARG2           (BE-L)

libyans|two|nn                  (BE-F)
indicted|libyans|obj    (BE-F)
bombing|lockerbie|nn    (BE-F)
indicted|bombing|for    (BE-F)
bombing|1991|in                 (BE-F)
```

A typical relation set may be {MOD} or {ARGn AUX MOD}. BE-L uses PropBank style semantic relations.

In the past, we have recommended BE-F. However, recently work has shown that BE-L does even better at corresponding to human judgments. We are pleased that the basic idea of BEs is borne out in different implementations. Other breakers will follow in later releases of the package. For research continuity, we continue to recommend BE-F.

Although we have experimented with multi-word proper name units (treating "United Nations" as a single

BE, which allows it to match "UN", for example), we cannot distribute this portion of the BE breakers and matchers, because it relies on a proper name identifier. We use BBN's IdentiFinder, which may be licensed from BBN (http://www.bbn.com/). An alternative is InXight's Thingfinder, which may be purchased from InXight (http://www.inxight.com/).

## 3.3 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.

## 3.4 Comparing and Matching Units

Matching BEs is less difficult than matching whole phrases, because less variation is possible. Nonetheless, we have identified a range of increasingly sophisticated matching strategies, some of which we do not know how to implement (arranged from strictest/easiest to most sophisticated):

- lexical identity: the words must match exactly, without alteration (implemented)
- lemma identity: the root forms of the words must match (implemented; root forms are obtained from WordNet)
- synonym identity: the words or any of their synonyms match. Synonyms may be obtained from WordNet, for example (not implemented yet)
- distributional similarity: words are similar according to the cosine distance on mutual information-based distributional similarity scores, obtained from the clustering package CBC (Lin and Pantel, 2002)
- (approximate) phrasal paraphrase matching (not implemented)
- semantic generalization match: BE words are replaced by semantic generalizations ("Mother Theresa" replaced by "human") and then matched, at a variety of levels of abstraction. This method has been implemented and tested but not distributed, since this relies on named entity identification and on WordNet

The default is exact lexical identity. The user can alternatively specify lemma identity.

Matching of triplet BEs can either include or ignore the BEs' relations, at the user's choice. In the current release, only strict lexical matching is provided. Depending on whether relation is being matched or not, all three (head, modifier, and relation) or only the first two fields have to match completely.

Future more sophisticated matcher modules should be able to recognize the equivalence (or partial-score equivalence) or such pairs as "approximately $20 million" and "19.8 million dollars"; anaphoric coreference "he said" and "Joe said"; abbreviations; and metonymy "Washington announced" and "The US government announced".

## 3.5 Combining Scores and Ranking Units

The implemented score adder module simply adds the point values of each BE in the summary to be evaluated. Optimizing the score integration function is a research task left for later, when a larger corpus of human judgments is available.

## 3.6 BEs and ROUGE

Note that ROUGE itself is also an instance of the BE framework, in which the BEs are unigrams (or ngrams of various types, depending on the parameter choice), the scoring function is simple unit points, and the simplest matching criterion is lexical identity. ROUGE scores can be requested as p art of the output of the BE Package.

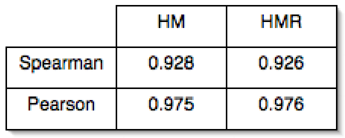
## 4 Judging DUC 2005

## 4.1 System Ranking

The BE package can be run in two separate parameterizations, HM (head-modifier) and HMR (head-modifier-relation). Table 1 shows the system ranking produced by running with HMR.

## 4.2 Correlation: BE vs. Responsiveness

NIST computed the average scaled responsiveness score of each summarizer across all topics. To validate BE, we computed the Spearman rank coefficient and Pearson coefficient between BE and responsiveness scores. A high correlation is found, as shown in Table 2.



| | HM | HMR |
|---|---|---|
| Spearman | 0.928 | 0.926 |
| Pearson | 0.975 | 0.976 |

Table 2. Correlation b/w BE and responsive-

## 4.3 Correlation: BE vs. ROUGE

Table 3 shows the correlation between BE and ROUGE. The ROUGE scores are macro-averaged by NIST.

| | 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 3. Correlation b/w BE and macroavg ROUGE.

```
 C 0.05646 (95%-conf.int. 0.04717 - 0.06738)
 I 0.05643 (95%-conf.int. 0.04363 - 0.07019)
 A 0.04963 (95%-conf.int. 0.04128 - 0.05784)
 E 0.04886 (95%-conf.int. 0.03948 - 0.06056)
 J 0.04653 (95%-conf.int. 0.03933 - 0.05512)
 D 0.04607 (95%-conf.int. 0.03680 - 0.05658)
 B 0.04467 (95%-conf.int. 0.03502 - 0.05652)
 G 0.04309 (95%-conf.int. 0.03468 - 0.05163)
 F 0.04248 (95%-conf.int. 0.03514 - 0.04988)
 H 0.04247 (95%-conf.int. 0.03390 - 0.05169)
17 0.02780 (95%-conf.int. 0.02603 - 0.02956)
 5 0.02773 (95%-conf.int. 0.02565 - 0.03005)
14 0.02678 (95%-conf.int. 0.02431 - 0.02939)
15 0.02675 (95%-conf.int. 0.02537 - 0.02824)
10 0.02654 (95%-conf.int. 0.02494 - 0.02821)
11 0.02346 (95%-conf.int. 0.02173 - 0.02523)
 4 0.02318 (95%-conf.int. 0.02184 - 0.02451)
 8 0.02245 (95%-conf.int. 0.02114 - 0.02392)
16 0.02219 (95%-conf.int. 0.02037 - 0.02419)
 6 0.02177 (95%-conf.int. 0.01961 - 0.02390)
25 0.02141 (95%-conf.int. 0.01940 - 0.02364)
 9 0.02139 (95%-conf.int. 0.01967 - 0.02331)
29 0.02097 (95%-conf.int. 0.01950 - 0.02254)
19 0.02091 (95%-conf.int. 0.01940 - 0.02237)
 3 0.02032 (95%-conf.int. 0.01898 - 0.02169)
24 0.02021 (95%-conf.int. 0.01872 - 0.02172)
 7 0.02004 (95%-conf.int. 0.01834 - 0.02165)
12 0.01981 (95%-conf.int. 0.01844 - 0.02118)
18 0.01967 (95%-conf.int. 0.01815 - 0.02132)
21 0.01943 (95%-conf.int. 0.01804 - 0.02082)
30 0.01699 (95%-conf.int. 0.01552 - 0.01842)
20 0.01696 (95%-conf.int. 0.01524 - 0.01891)
27 0.01651 (95%-conf.int. 0.01528 - 0.01784)
13 0.01551 (95%-conf.int. 0.01431 - 0.01676)
32 0.01523 (95%-conf.int. 0.01403 - 0.01655)
22 0.01452 (95%-conf.int. 0.01337 - 0.01570)
28 0.01416 (95%-conf.int. 0.01314 - 0.01524)
31 0.01345 (95%-conf.int. 0.01210 - 0.01479)
 2 0.01294 (95%-conf.int. 0.01163 - 0.01436)
26 0.01279 (95%-conf.int. 0.01087 - 0.01477)
 1 0.01196 (95%-conf.int. 0.01072 - 0.01344)
23 0.00647 (95%-conf.int. 0.00560 - 0.00729)
```

Table 1. System ranking by BE-HMR.

### 4.4 Correlation Overview

In Figure 1 we show the overall correlation between ROUGE, BE, responsiveness, and the Pyramid method. The label on each link indicates the parameterization, the Spearman rank coefficient, and the Pearson coefficient between the metrics connected by the link. Correlations shown here are computed by comparing against the primary NIST human responsiveness assessment scores. Only 20 topics, those used in the Pyramid annotations, were used to compute the scores for all four methods. The Pyramid scores were computed using the model/reference summaries involved in the annotation process. And the responsiveness, ROUGE, and BE scores were computed against all 9 model/reference summaries. Correlation figures published elsewhere may have used different number of model summaries or number of topics.

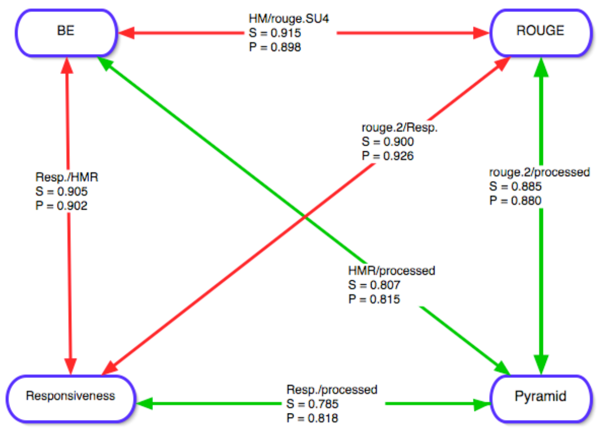


Figure 1. Correlation overview.

## 5 Conclusions and Future Work

Given the number of potential alternatives embodied in the BE Package, the amount of work required to determine the optimal combination of parameters and techniques is enormous. We are however very encouraged by the fact that the first and major bottleneck, the automated creation of minimal BEs, seems to have been addressed adequately for now, given by different BE breaker modules (different parsers and different BE chunking rules) independently.

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.

## References

Amigo, E., V. Peinado, J. Gonzalo, A. Peñas, and F. Verdejo. 2004. An Empirical Study of the Information Synthesis Task. *Proceedings of the conference of the Association for Computational Linguistics (ACL)*. Barcelona, Spain.

Charniak, E. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of NAACL 2000*.

Collins, M. Head-Driven Statistical Models for Natural Language Parsing. *PhD Dissertation*, University of Pennsylvania, 1999.

DUC. 2001–2004. The series of Document Understanding Conference proceedings.

Heidorn, G. 2000. Intelligent Writing Assistance. In R. Dale, H. Moisl and H. Somers (eds.), *A handbook of natural language processing: Techniques and applications for the processing of language as text*, New York: Marcel Dekker.

Hori, C., T. Hori, and S. Furui. 2003. Evaluation Method for Automatic Speech Summarization. *Proceedings of Eurospeec conference*. Geneva, Switzerland.

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*.

Lin, D. and P. Pantel. 2002. Concept Discovery from Text. *Proceedings of Conference on Computational Linguistics (COLING-02)*. 577–583. Taipei, Taiwan.

Magerman, D. 1994. Natural Language Parsing as Statistical Pattern Recognition. *Ph.D. Thesis*, Stanford University.

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.

Radev, D. and D. Tam. 2003. Summarization Evaluation via Relative Utility. *Proceedings of the CIKM conference.* New Orleans, LA.

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.