

BEwT-E: Basic Elements with Transformations for Evaluation

Stephen Tratz and Eduard Hovy

Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292 USA
{stratz,hovy}@isi.edu

Abstract

This paper describes BEwT-E (Basic Elements with Transformations for Evaluation), an automatic system for evaluating text summarization or machine translation tasks. BEwT-E is a new, more sophisticated implementation of the BE framework that uses transformations to match short, syntactically well-defined units called Basic Elements (BEs) that are lexically different yet semantically similar.

1 Introduction

Human evaluation of machine-produced text can be time consuming, costly, and prone to human variability (Teufel and van Halteren, 2004; Nenkova and Passonneau, 2004). To avoid these issues, automatic systems have been developed in order to more efficiently and objectively evaluate NLP systems. In recent years, a lexical n-gram method named BLEU has become the standard automatic evaluation metric in the field of machine translation (Papineni, 2002). A similar metric named ROUGE has been used in the field of text summarization to compare human-written summaries with computer-generated summaries (Lin and Hovy, 2003). Similar n-gram methods such as POURPRE have been successfully applied to question answering evaluation (Lin and Demner-Fushman, 2005).

A problem exhibited by many automated evaluation metrics methods including BLEU and ROUGE is their limited ability to match alternative phrasings. No automated system will match “a massive emerald-colored vehicle” to “a large green car” using lexical identity alone.

Some newer metrics such as METEOR, M-BLEU, and M-TER (Banerjee and Lavie, 2005; Agarwal and Lavie, 2008) have incorporated the ability to match using stemming as well as the synonym sets available in WordNet. CDER (Leusch et al., 2006) has experimented with using weighted word substitution based upon character-based edit distance.

Another problem with other methods is their reliance on surface-level formulation, and the absence of sensitivity to syntactic structure. Using bigrams, the phrase “large car” in a system summary, for example, would not match “large green car” in a gold standard summary, despite “large” and “green” independently modifying “car”. In an attempt to overcome this, ROUGE employed so-called skip n-grams, namely n-grams that can accommodate a small number of skipped items.

To overcome both of these common shortcomings, the Basic Element (BE) method for summarization evaluation was developed (Hovy et al., 2006). This method facilitates matching of expressive variants of syntactically coherent units called Basic Elements (BEs). The system was able to achieve fairly good correspondence with human evaluation of text summarization. However, it still only performed rudimentary matching of alternative phrasings using a list of paraphrases (Zhou et al., 2006).

This paper describes a new implementation of the BE method called BE with Transformations for Evaluation (BEwT-E) that includes a significantly improved matching capability using a variety of operations to transform and match BEs in various ways.

We first outline the BE method. Then we describe our new implementation of it along with

how the BEs are weighted and descriptions of the transformations used to enable flexible matching.

2 The Basic Element Method

The intuition behind BEs is to decompose the system-generated texts and the gold standard(s) each to a list of minimal-length syntactically well-defined units (Basic Elements) and then to compare the two lists. Five issues must be addressed:

- What is the nature of a minimal unit?
- How are BEs extracted?
- How should each BE be weighted?
- How should matches be determined?
- How should the matches be combined into an overall score?

As described in (Hovy et al., 2005), each BE is a syntactic unit (a single word or multi-word phrase or name; a modifier-head pair, etc.). In the new implementation, each BE consists of a list of one to three words and their associated parts-of-speech or NER type. These include:

- Unigram BEs: all nouns, verbs, and adjectives found in the text
- Bigram BEs: subject+verb, verb+object, verb+adverb, verb+adjective, verb+particle, adjective+noun, headnoun+headnoun_of_appositive, possessorOf+headnoun, pronominal_noun+headnoun, etc.
- Trigram BEs: two head words connected via a preposition or other functional word like 'because', 'since', 'while', or 'where'.

3 Comparing Text Using BEwT-E

3.1 Extracting BEs

In order to extract the BEs, we first parse the texts using the Charniak parser (Charniak and Johnson, 2005), identify named entities using the LingPipe NER system (Baldwin and Carpenter), and then extract the BEs using a series of Tregex rules (Levy and Andrew, 2006). Tregex rules can be thought of as regular expressions over trees. Examples of the Tregex rules used by BEwT-E and the BEs they produce for a sample sentence are given Figure 1.

```
John's cat drank milk.
Charniak parse:
(S1 (S (NP (NP (NNP John) (POS 's)) (NN cat)) (VP
(VBD drank) (NP (NN milk))) (. .)))

Rule Name: Verb to NPHead
Tregex: VP [<# __=x & < (NP <# !POS=y)]
Tokens to Extract: xy
Extracted BEs: drank|VBD+milk|NN

Rule Name: Possessor of NPHead
Tregex: NP [< (NP <# (POS $- __=x)) & <# __=y]
Tokens to Extract: xy
Extracted BEs: John|Person+cat|NN
```

Figure 1. Example sentence, its Charniak parse, and the output from two separate BE extraction rules.

If any token identified for extraction by a BE extraction rule is contained within a named entity string recognized by a Named Entity Recognition (NER) system, the entire named entity string is extracted in place of the word.

During the extraction process, it is possible that several identical BEs may be extracted from the same document. Since duplicate BEs do not, by themselves, convey much additional information about the semantic content of a text, we experimented both with and without duplicates.

3.2 Weighting BEs

In weighting the BEs, a basic assumption to date has been that a fragment of content mentioned in several reference segments is more important, and should weigh more, than a fragment mentioned in only one. In manual text summarization studies, both Teufel and van Halteren (2004) and Nenkova and Passonneau (2005; the Pyramid Method) adopt the ‘popularity score’ rule: a fragment (called SCU or semantic content unit in the latter) is assigned points equal to the number of references containing it.

BEwT-E has three different weighting methods implemented that use the number of references in which a BE occurs in order to determine its weight. The three weighting schemes are *binary* (each matched reference BE is worth 1 regardless of the number of references containing it), *root* (the BE weight is equal to the square root of the number of references containing it), and *total* (the BE is worth 1 point for each reference that contains it).

3.3 Transformations Definition

The focus of our work is the matching and tallying of BEs from system and human texts. The original BE system matched primarily by lexical identity, expanded by paraphrase substitution using a large list of paraphrase alternatives extracted from a machine translation system (Zhou et al., 2006). However, it is usually possible to express similar information using many other types of different including variant choices of words and syntactic structures. Recognizing such matches typically requires humans. No automated system today can recognize all variants, and know which degrees of semantic similarity they express.

Nonetheless, one can make inroads into this problem automatically. BEwT-E uses a set of transformations to match BEs that convey similar semantic content yet are lexically different. What exactly constitutes acceptable similarity is captured by the transformations used by BEwT-E, which are listed below.

Lemmatization/Delemmatization: Words in BEs can be transformed so that they match regardless of plurality, tense, etc. For example, this transformation would enable “greenJJ+plantsINNS” to match “greenJJ+plantINN”.

Synonymy: WordNet (Miller et al., 1990) synonym sets were used to expand synonyms for nouns, verbs, and adjectives. For this, BEwT-E assumes that each word is an instance of its most frequent sense. For example, this transformation would enable “drinkIVB+milkINN” to match “imbibeIVB+milkINN”.

Generalization/Specialization: Hypernym and hyponym relationships from WordNet were used to generalize/specialize nouns and verbs so that BEs like “newspaperINN” and “pressINN” could be matched. This transformation treats person, organization, and location entities identified by the NER system as “personINNP”, “organizationINNP”, and “locationINNP”, respectively. For now, this transformation is not limited to just the immediate parent and child sense nodes in the WordNet hierarchy. As with the synonymy transformation, BEwT-E assumes each word is an instance of its most frequent sense.

Preposition Generalization: The Preposition Project has produced a sense inventory of English prepositions (Litkowski and Hargraves, 2005). This was used to create a list of all legal preposition mappings so that prepositions could be expanded. For example, this transformation would enable “manINN+fromIN+La_ManchalLocation” to match “manINN+ofIN+La_ManchalLocation”. If BEwT-E utilized a preposition sense disambiguation system, this transformation could be further restricted.

Add/Drop Periods: Abbreviations can often occur with or without periods. To handle this, this transformation adds or drops periods. This transformation would enable BEs “U.S.A|Location” and “USA|Location” to match.

Abbreviations/Acronyms: BEwT-E has a transformation that enables matching abbreviations with their expanded form. This transformation consists of two parts. This first part is simply a lookup list of common abbreviations that includes lists of person titles, street names, states, provinces, measurements, and countries. The second part is a block of code capable of generating some of the most likely abbreviations for persons, organizations, and locations.

Name Shortener/Expander: This transformation transforms entity names so that BEs like “John_B_Smith|Person” can be matched against “Smith|Person”, “John|Person” or “John Smith|Person” and organization names like “Google|Organization” can be matched with “Google In|Organization”

Denominalization: It is not unusual for a one reference to an event to occur in the form of a verb and another in the form of a noun. To transform BEs from the nominalized form back to the verb form, this transformation utilizes the “derived from” relationship links in WordNet. For example, this transformation enables the BE “rejectionINN+ofIN+John|Person” to match either “John|Person+rejectIVB” or “rejectIVB+John|Person”.

Nominalization: This transformation is similar to the denominalization transformation except it operates in the opposite direction. For example, this transformation would allow “gerbilINN_hi-

bernated|VBD” to match “hibernation|NN+of|IN+gerbill|NN”.

“Role” Transform: In some sentences, the role a person plays appears as a prenominal noun next to his/her name while in other sentences the person may be seen performing the action associated with the role. This transformation was created to handle these situations. For example, this transformation enables BEs “Barry_Bonds|Person+hit|VBD” and “hitter|NN+Barry_Bonds|Person” to match. In order to do this, it uses the “derived from” relationship links in WordNet.

Prenominal Noun ↔ Prepositional Phrase: This transform converts BEs such as “Iraq|Location+invasion|NN” into similar trigram BEs such as “invasion|NN_of|IN_Iraq|Location”, or vice versa.

Noun Swapping for IS-A type rules: Some BE extraction rules, such as the rule for handling appositives, extract a pair of nouns that are expected to exhibit a IS-A relationship. Since the order of these nouns is unimportant, this transformation allows the BEs to match even if the nouns are in reverse order. For example, this transformation enables “Phelps|Person+swimmer|NN” to match “swimmer|NN+Phelps|Person”.

Pronoun Transform: Pronouns are commonly used in place of a more specific reference, presenting problems for NLP systems. This transform allows personal pronouns to match person names and the plural pronouns “they” and “them” to match organization names and plural nouns.

Pertainym Adjectives Transform: Using meronym, holonym, and derivational relationship links in WordNet, this transform enables BEwTE to match BEs like “China|Location+people|NNS” to “Chinese|JJ+people|NNS” and “biological|JJ+instruments|NNS” to “biology|NN+instruments|NNS”.

Many of these transformations can be applied more or less aggressively. For example, synonym lookups and generalization could be limited only to bigram and trigram BEs and/or could use all available WordNet senses instead of just the most frequent sense. Exploring the potential and

risks of such degrees is an interesting subject for future research.

3.4 Transformations Implementation

The application of the transformations occurs during a step between BE extraction and the overall score computation. Each segment is processed separately.

First, a reference BE pool containing all of the BEs extracted from the references for a particular segment is constructed. This pool is the complete set of BEs that other BEs may be mapped to.

Next, one by one, each the BEs for the segment are passed through the transformation pipeline, an ordered list of the available transformations. Both the BE that was passed into the transformation as well as any of the transformation's outputs are then passed into the subsequent transformation. The transformed versions of the original BEs that match any of the BEs in the reference set are then saved to disk along with the list of transformations used to produce them.

The transformation process is rather expensive and an exponential number of computations may be required for any single BE. However, in practice, many of the transformations will not produce any transformed versions of a BE. Actually, the BE transformation process tends to take less time than the parsing step.

To further reduce the number of computations performed, a list of the transformed versions of a BE is maintained along with the list of set(s) of transformations used to produce each transformed version. If a transformed version of a BE is produced that is identical to a previous production and uses a super set of the transformations used in the previous production, the new production will be ignored and not passed into the next transformation.

Further enhancements to the transformation process may result in significant speedup. The interface currently used to access WordNet relies upon random file access and thus is relatively slow when compared to memory access or modern database systems. Also, some transformations have minimal or no interaction and could be combined into a single transformation step, thereby significantly decreasing the number of possibly pathways through the list of transformations.

3.5 Computing the Overall Score

After undergoing several transformations, a single BE may match several of the reference's BEs. These reference BEs may have been given different weights based upon their frequency in the references and, in future versions of BEwT-E, the matching may have a value less than 1.0 if a transformation was required to perform the match. This complicates the scoring process because, in computing the comparison score between two text segments, no BE is allowed to match or be matched multiple times.

The BE matching problem is essentially an instance of the weighted assignment problem and the unnormalized formula is expressed mathematically in Figure 1. The BE weighting function W determines the weight of the reference BE and is discussed in Section 3.2. The comparison function C returns a measure of how similar a pair of BEs are. In the current configuration of BEwT-E, C always returns 1.0 even though there are parameters for adjusting the similarity of the match based upon the set of transforms used to produce it. In the future, these parameters may be tuned.

BEwT-E implements a successive shortest paths (also known as shortest augmenting paths) algorithm to find the optimal BE matching. For more information regarding using successive shortest paths for solving assignment problems see (Enquist, 1982).

The total value of the matching is normalized by the total weight of the reference's BEs. To compute an overall score for a document or system, the average of the scores for the individual segments is calculated.

$$\begin{aligned}
 & \text{maximize } \sum_{i=0}^N \sum_{j=0}^M C(i,j) W(j) x_{ij} \\
 & \text{subject to} \\
 & \sum_{i=0}^N x_{ij} \in [0,1] \text{ for all } j \text{ where } 0 \leq j \leq M \\
 & \sum_{j=0}^M x_{ij} \in [0,1] \text{ for all } i \text{ where } 0 \leq i \leq N \\
 & x_{ij} \in [0,1]
 \end{aligned}$$

Figure 1. Problem of calculating the unnormalized comparison score between two sets of BEs with comparison and weighting functions C and W .

3.6 Multiple References

In order to calculate a BEwT-E score when multiple references are available, we compare the peer segment against each of the reference segments and consider the highest score to be the final multi-reference score. However, one may like to rank each reference in the same list as each peer system. To account for the fact that comparing a reference against itself would result in a perfect score and not comparing it against itself would mean that the reference gets compared to fewer references than the automatically generated peers, jackknifing was implemented. This involves creating N subsets of the N references, each of which is missing one reference. The score for each peer is then calculated by taking the average of the multi-reference scores produced by using these N different subsets. By default, BEwT-E has jackknifing enabled.

4 Evaluation

For MetricsMATR08, a development set was released to the participants that included data from the NIST Open MT 06 evaluation. BEwT-E scores both with and without transformations enabled are given in Table 1. Pearson and Spearman coefficients calculated against average human-produced segment adequacy judgments for the systems are provided in Table 2. Pearson correlation on a per segment basis ($N=1900$) is given in Table 3.

Translation	BEwT-E score	
	Transforms Off	Transforms On
reference02	0.686	0.729
reference03	0.670	0.713
reference01	0.638	0.686
reference04	0.581	0.630
system07	0.532	0.593
system05	0.546	0.592
system08	0.550	0.592
system04	0.525	0.579
system06	0.520	0.560
system01	0.434	0.491
system02	0.345	0.383
system03	0.295	0.331

Table 1. BEwT-E scores for NIST Open MT 06 development data with and without transformations enabled.

	Transforms Off	Transforms On
Pearson	0.970	0.982
Spearman	0.786	0.929

Table 2. Correlation between BEwT-E system scores and average segment adequacy.

	Transforms Off	Transforms On
Pearson	0.621	0.631

Table 3. Pearson correlation coefficients from comparing BEwT-E scores and segment adequacy judgments on a per segment basis.

While the effect of the transforms upon the Pearson and Spearman coefficients is positive, it does appear to be statistically significant.

5 Conclusions and Future Work

Future work includes additional transformations and an anaphora resolution capability. In addition, replacing idioms with more literal text as a preprocessing step may prove helpful in some cases. It may also be possible to determine and utilize some quality estimate of the various references.

We would like to examine which BE extraction rules produce the most predictive BEs, possibly assigning different weights to the different rules.

It would be worthwhile examining the impact of using different preprocessing tools, including dependency parsers and other NER systems.

We are also interested in applying modified versions of BEwT-E to other domains and corpora.

6 Acknowledgments

Stephen Tratz is supported by a NDSEG fellowship.

References

- Agarwal, A. and A. Lavie. 2008. METEOR, M-BLEU and M-TER: Evaluation Metrics for High-Correlation with Human Rankings of Machine Translation Output. *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 115-118, Columbus, Ohio.
- Baldwin, B. and B. Carpenter. LingPipe. <http://www.alias-i.com/lingpipe/>.
- Banerjee S. and A. Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved

Correlation with Human Judgments. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65-72, Ann Arbor, Michigan.

- Charniak, E. and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173-180, Ann Arbor, MI.
- Conroy, J.M. and H. Trang Dang. 2008. Mind the Gap: Dangers of Divorcing Evaluations of Summary Content from Linguistic Quality. *Proceedings of the COLING conference*. Manchester, UK.
- Enquist, M. 1982. A Successive Shortest Path Algorithm for the Assignment Problem. *IFOR*, 20(4): 370-384.
- Giannakopoulos, G., V. Karkaletsis, G. Vouros, P. Stamatopoulos. 2008. Summarization System Evaluation Revisited: N-gram Graphs. *ACM Transactions on Speech and Language Processing - To Appear*
- Hovy, E.H., C.Y. Lin, and L. Zhou. 2005. Evaluating DUC 2005 using Basic Elements. *Proceedings of DUC-2005 workshop*.
- Hovy, E.H., C.Y. Lin, L. Zhou, and J. Fukumoto. 2006. Automated Summarization Evaluation with Basic Elements. Full paper. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*. Genoa, Italy.
- Leusch, G., N. Ueffing, and H. Ney. 2006. CDER: Efficient MT Evaluation Using Block Movements. In *Proceedings of the Thirteenth Conference of the European Chapter of the Association for Computational Linguistics*.
- Levy, R. and G. Andrew. 2006. Tregex and Tsurgeon: Tools for Querying and Manipulating Tree Data Structures. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*.
- Lin, C.Y. and E.H. Hovy. 2003. Automatic Evaluation of Summaries using n-Gram Co-occurrence Statistics. *Proceedings of the HLT-2003 conference*.
- Lin, J. and D. Demner-Fushman. 2005. Evaluating Summaries and Answers: Two Sides of the Same Coin? *Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*. Ann Arbor, MI. 41-48.
- Litkowski, K.C. and O. Hargraves. 2005. The Preposition Project. *Proceedings of ACL-SIGSEM Workshop on The Linguistic Dimensions of Prepositions*

and Their Use in Computational Linguistic Formalisms and Applications. University of Essex-Colchester, UK. 171–179.

Miller, G.A., R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 2(4): 235–245.

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

Papineni, K., S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311-318, Philadelphia, PA, July.

Teufel, S. and H. van Halteren. 2004. Evaluating Information Content by Factoid Analysis: Human Annotation and Stability. *Proceedings of the EMNLP conference*. Barcelona, Spain.

Zhou, L, C.Y. Lin, D.S. Munteanu, and E.H. Hovy. 2006. ParaEval: Using Paraphrases to Evaluate Summaries Automatically. *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*.