Automatic Pyramid Evaluation Exploiting EDU-based Extractive Reference Summaries

Tsutomu Hirao[1] and **Hidetaka Kamigaito**[2] and **Masaaki Nagata**[1]
[1]NTT Communication Science Laboratories, NTT Corporation
[2]Institute of Innovative Research, Tokyo Institute of Technology
[1]{hirao.tsutomu,nagata.masaaki}@lab.ntt.co.jp
[2]kamigaito@lr.pi.titech.ac.jp

Abstract

This paper tackles automation of the pyramid method, a reliable manual evaluation framework. To construct a pyramid, we transform human-made reference summaries into extractive reference summaries that consist of Elementary Discourse Units (EDUs) obtained from source documents and then weight every EDU by counting the number of extractive reference summaries that contain the EDU. A summary is scored by the correspondences between EDUs in the summary and those in the pyramid. Experiments on DUC and TAC data sets show that our methods strongly correlate with various manual evaluations.

1 Introduction

To develop high quality summarization systems, we need accurate automatic content evaluation. Although, various evaluation measures have been proposed, ROUGE-N (Lin, 2004), Basic Elements (BE) (Hovy et al., 2006) remain the de facto standard measures since they strongly correlate with various manual evaluations and are easy to use. However, the evaluation scores computed by these automatic measures are not so useful for improving system performance because they merely confirm if the summary contains small textual fragments and so they do not address semantic correctness.

The pyramid method was proposed as a manual evaluation that well supports the improvement of summarization systems (Nenkova and Passonneau, 2004; Nenkova et al., 2007). First, the method identifies conceptual contents, Summary Content Units (SCUs), in reference summaries and then constructs a pyramid by collecting semantically equivalent SCUs. The weight of an SCU in the pyramid is defined as the number of reference summaries that contain the SCU. Thus, an SCU shared by many reference summaries is given higher weight. Second, a system summary is scored by the correspondences between SCUs in the summary and the pyramid. Its results are very useful for system improvement, *i.e.*, we can know which important SCUs the system could or could not include in the summary. Although the pyramid method is reliable, it requires considerable cost and effort.

To address the weaknesses, automatic pyramid evaluation, Pyramid Evaluation via Automated Knowledge Extraction (PEAK) was proposed (Yang et al., 2016). Since SCU is the conceptual content of the text, it is difficult to automatically extract them from reference summaries by systems. Thus, PEAK regards subject-predicate-object triples as alternatives to SCUs and constructs a pyramid by clustering semantically equivalent triples. However, the performance of subject-predicate-object triples extraction is not satisfying for the practical demands and semantic similarity utilized for clustering the triples does not correlate well with human judgment (see Section 2). As a result, the resultant pyramid is unreliable. Actually, PEAK is significantly inferior to ROUGE and BE (see Section 4.3) in terms of correlation.

To cope with the above problems, this paper proposes yet another automatic pyramid evaluation method. Its key feature is constructing a pyramid that consists of Elementary Discourse Units (EDUs), clause-like text units introduced in Rhetorical Structure Theory (Mann, William Charles and Thompson, Sandra Annear, 1988), in the source documents. In other words, we regard EDUs as alternatives to SCUs. To construct the pyramid, we transform human-made reference summaries into EDU-based extractive reference summaries and then weight every EDU by counting the number of the extractive reference summaries that contain the EDU. The rea-

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4177–4186
Brussels, Belgium, October 31 - November 4, 2018.

son why we derive extractive reference summaries whose SCUs are EDUs is as follows. First, Li et al. (2016) reported that EDUs are very similar to SCUs. Second, the performance of EDU segmenter is sufficient to satisfy practical requirements (see Section 2). Third, we do not need measure any semantic similarity to identify EDUs common to the extractive reference summaries. We also examine two types of extractive reference summary. One is based on the alignment between EDUs in reference summary and source documents. The other is based on the *extractive oracle summary* (Hirao et al., 2017). We conducted experiments on the Document Understanding Conference (DUC) 2003 to 2007 data sets and Text Analysis Conference (TAC) 2008 to 2011 data sets. The results showed that our methods exhibit strong correlation with manual evaluations.

2 Background and Related Work

The pyramid method (Nenkova and Passonneau, 2004; Nenkova et al., 2007), a manual evaluation framework, was developed to measure the content coverage of summaries. The pyramid method consists of two steps: (1) pyramid construction, and (2) summary scoring based on the pyramid. First, human annotators identify Summary Content Units (SCUs), conceptual content units in the reference summaries. They then construct a pyramid by clustering and weighting SCUs. The weight of an SCU is defined as the number of reference summaries that contain the SCU. As a result, if there are K reference summaries, the upper bound weight of an SCU in the pyramid is K and the lower bound is 1. Second, the score for a summary is determined by the correspondences between SCUs in the summary and those in the pyramid. Thus, the score is defined as the sum of weights of SCUs that correspond to those in the pyramid in the summary divided by the sum of SCU weight possible for an average-length reference summary. The pyramid method has two advantages over conventional manual evaluations: (1) the score is not intuitive but is systematically computed, *i.e.*, the score can be explained as the sum of weights of SCUs in the pyramid, (2) the correspondences between the SCUs in a summary and the pyramid tell us whether the summary contains important SCUs or not. Thus, the results explicitly tell us why a summary was given a good or bad score.

During the past few years, studies have focused on the automatic scoring of summaries based on manually generated pyramids. Harnly et al. (2005) proposed a scoring method that matches SCUs in the pyramid with possible textual fragments in the summary. They enumerate all possible textual fragments within a sentence in the summary and compute similarity scores between the fragments and the SCUs in the pyramid based on unigram overlap. Then, they find the optimal correspondences between SCUs and the fragments that maximize the sum of similarity scores. Passonneau et al. (2013) extended the method by introducing distributional semantics to compute the similarity scores between SCUs and the fragments.

Recently, Yang et al. (2016) proposed the first automatic pyramid method, Pyramid Evaluation via Automated Knowledge Extraction (PEAK). PEAK employs subject-predicate-object triples extracted by ClausIE (Del Corro and Gemulla, 2013) as SCUs, and constructs pyramids by cutting a graph whose vertices represent the triples and whose edges represent semantic similarity scores between the triples computed by Align, Disambiguate and Walk (ADW) (Pilehvar et al., 2013). When evaluating a summary, PEAK constructs a weighted bipartite graph whose vertices represent subject-predicate-object triples extracted from the pyramid and the summary, respectively; the edges represent the similarity scores between the triples as computed by ADW. It scores the summary by solving the Linear Assignment Problem which involves maximizing the sum of the similarity scores on the bipartite graph.

The major difference between PEAK and our method is that the former regards the reference summary as a set of subject-predicate-object triples while the latter regards a reference summary as a set of EDUs obtained from the source documents. Thus, to construct high quality pyramids, PEAK is required to not only accurately extract the triples but also measure the semantic similarity between them accurately. However, in general, both extracting the triples and measuring the semantic similarity are still challenging NLP tasks. The performances are not always achieved in practical use. Actually, the F-measure of ClausIE is around 0.6 (Del Corro and Gemulla, 2013) and the correlation coefficients between the semantic similarity obtained from ADW and human judgment lie in the range of 0.55 to 0.88

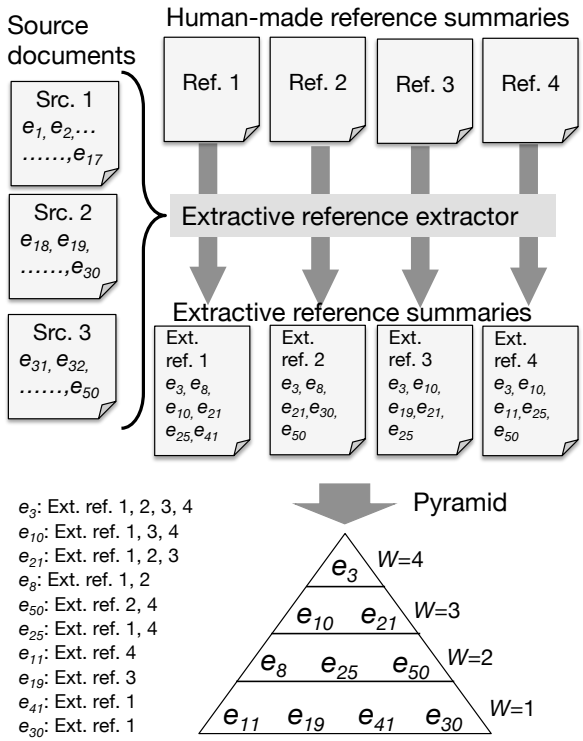

Figure 1: Overview of our pyramid construction.

(Pilehvar et al., 2013). As a result, the resultant pyramids have insufficient quality to be practical. Clearly, further improvement is necessary.

While our method is required to decompose a document into EDUs accurately, the EDU segmenter offers accurate decomposition performance; existing EDU boundary detection methods have F-measures over 0.9 (Fisher and Roark, 2007; Feng and Hirst, 2014). Moreover, since extractive reference summaries are set of EDUs from the source documents, we do not need semantic similarity to identify EDUs that have the same meaning. Thus, we can easily construct a pyramid by simply counting the number of extractive reference summaries that contains each EDU.

3 Automatic Pyramid Evaluation

First, we transform human-made reference summaries into extractive reference summaries; the EDUs in the source documents are used as the atomic units. Second, we construct a pyramid by weighting EDUs in the extractive reference summaries. EDU weights are defined as the number of reference summaries that contain each EDU (see Figure 1). In addition, we propose two techniques for deriving the extractive reference summaries.

3.1 Extractive Reference Summaries based on Alignment between EDUs

When similarity scores between EDUs in a reference summary and those in the source documents are available, we can regard extractive reference summary derivation as an optimal alignment problem with a length constraint, an extension of Linear Assignment Problem. We assume that a bipartite graph in which the vertices represent EDUs in the reference summary and source documents, and the edges represent similarity scores between the EDUs. The optimal alignment is obtained by solving following ILP problem:

$$\text{maximize} \sum_{j=1}^{|\mathcal{E}|} \sum_{k=1}^{|\mathcal{M}|} \phi(e_j, m_k) a_{j,k} \quad (1)$$

$$\text{s.t.} \sum_{j=1}^{|\mathcal{E}|} \sum_{k=1}^{|\mathcal{M}|} \ell(e_j) a_{j,k} \leq L_{\max} \quad (2)$$

$$\sum_{j=1}^{|\mathcal{E}|} a_{j,k} \leq 1 \qquad \forall k \quad (3)$$

$$\sum_{k=1}^{|\mathcal{M}|} a_{j,k} \leq 1 \qquad \forall j \quad (4)$$

$$a_{j,k} \in \{0, 1\} \qquad \forall j, k. \quad (5)$$

$\mathcal{E}$ is the set of all EDUs in the source documents and $\mathcal{M}$ is the set of all EDUs in the reference summary. $\ell()$ returns the length (the number of words) of a textual unit. $\phi(e_j, m_k)$ returns the similarity score between the j-th EDU in the source documents and the k-th EDU in the reference summary as follows:

$$\phi(e_j, m_k) = \frac{\ell(\text{LCS}(e_j, m_k))}{\ell(m_k)}. \quad (6)$$

$\text{LCS}(\cdot, \cdot)$ returns the Longest Common Subsequence between e_j and m_k. $a_{j,k}$ is a binary indicator, and $a_{j,k} = 1$ denotes that the j-th EDU e_j in the source documents is aligned to the k-th EDU in the reference summary, *i.e.*, e_j is included in the extractive reference summary. Equation (2) ensures the the length of the extractive reference summary is less than $L_{\max}$, the length of the human-made reference summary. After solving the ILP problem, we can obtain the extractive reference summaries by collecting EDUs according to $a_{j,k} = 1$.

3.2 Extractive Reference Summaries based on Extractive Oracle Summaries

As another extractive reference summary, we can utilize extractive oracle summary (Hirao et al., 2017). The extractive oracle summary is defined as the set of consequential textual fragments within a sentence obtained from the source documents that has the maximum automatic evaluation score. Since we regard EDUs as SCUs and employ ROUGE/BE as an automatic evaluation measure, an extractive reference summary is a summary that consists of EDUs in the source documents and has maximum ROUGE/BE score.

For a given reference summary R , the extractive oracle summary is defined as follows:

$$\begin{aligned} O = &\arg\max_{E \subseteq \mathcal{E}} f(R, S) \\ &\text{s.t.} \quad \ell(S) \leq L_{\max}. \end{aligned} \tag{7}$$

$f()$ denotes an automatic evaluation measure (ROUGE/BE) and is defined as follows:

$$f(R, S) = \frac{\sum_{i=1}^{|U|} \min\{N(u_i, R), N(u_i, S)\}}{\sum_{i=1}^{|U|} N(u_i, R)}. \tag{8}$$

S is a system summary and U is the set of all atomic units in the reference summary. N-grams are utilized as the units used in computing ROUGE and head-modifier-relation triples are utilized in computing BE. $N(u_i, R)$; $N(u_i, S)$ returns the number of occurrences of the units in the reference and system summary, respectively.

Since the extractive oracle summaries in Hirao et al. (2017) are based on sentences, we extend the method to obtain EDU-based extractive oracle summaries. The ILP formulation that returns an extractive oracle summary is defined as follows:

$$\text{maximize} \sum_{i=1}^{|U|} z_i - \sum_{m=1}^{|S|} s_m \tag{9}$$

$$\text{s.t.} \sum_{j=1}^{|\mathcal{E}|} \ell(e_j) x_j \leq L_{\max} \tag{10}$$

$$N(u_i, R) \geq z_i \qquad \forall i \tag{11}$$

$$\sum_{n \in V_i} d_n \geq z_i \qquad \forall i \tag{12}$$

$$x_{\text{left}(n)} \geq d_n \qquad \forall n \tag{13}$$

$$x_{\text{right}(n)} \geq d_n \qquad \forall n \tag{14}$$

$$s_{\text{c}(j)} \geq x_j \qquad \forall j \tag{15}$$

$$d_n \in \{0, 1\} \qquad \forall n \tag{16}$$

$$x_j \in \{0, 1\} \qquad \forall j \tag{17}$$

$$z_i \in \mathbb{Z}_+ \qquad \forall i. \tag{18}$$

z_i is the count of the i-th unit in the oracle summary. x_j is a binary indicator, $x_j = 1$ denotes that the j-th EDU, e_j, is included in the oracle summary. s_m is a binary indicator, $s_m = 1$ denotes that EDU(s) in m-th sentence is included in the oracle summary. The value of $\sum_{m=1}^{|S|} s_m$ is equal to the number of sentence whose EDU(s) is used in oracle summary. Thus, an oracle summary that consist of fewer sentences tends to obtain a higher objective value. Therefore, we can avoid generating fragmented oracle summaries with low readability. This objective function is inspired by the work of compressive summarization method (Morita et al., 2013). V_i is the set of indices indicating the position of the i-th unit, and d_n is a binary indicator indicating whether the n-th unit is contained in the oracle summary or not. Function left$(\cdot)$ and right$(\cdot)$ return the index of EDU that contains a word on the left in the unit, and the index of EDU that contains a word on the right in the unit, respectively. Function c$(\cdot)$ returns the index of sentence that contains j-th EDU.

Figure 2 shows examples. Suppose that the 10-th triple in U is "<has,computer,rcmod>". From the figure, the indices of the triple corresponding to "<has,computer,rcmod>" are 6 and 21. Thus, $V_{10} = \{6, 21\}$. The word on the left in the triples is "computer" and the word on the right is "has". For the first triple, the index of the EDU that contains "computer" is 1 and the index of the EDU that contains "has" is 2. For the second triple, the index of the EDU that corresponds to "computer" is 4, while that of "has" is 5. Thus, $\text{left}(6) = 1$ and $\text{right}(6) = 2$, $\text{left}(21) = 4$, and $\text{right}(21) = 5$.

After solving the ILP problem, we construct the extractive oracle summary by collecting EDUs according to $x_j = 1$.

3.3 Pyramid Construction: EDU Weighting

By deriving extractive references, we can easily construct a pyramid. The weight of an EDU is defined as the number of extractive references that contain the EDU. Here, $\mathcal{P}$ is a complete set of all EDUs in K extractive reference summaries, *i.e.*, $\mathcal{P} = \bigcup_{i=1}^{K} E_i$. E_i is the set of EDUs obtained from the i-th extractive reference summary. The weight

Index of word/triple	word	Index of dependant	Relation	Index of EDU
1	**We**	**2**	**nsubj**	**1**
2	**need**	0	ROOT	**1**
3	a	4	det	1
4	**computer**	2	dobj	**1**
5	that	6	nsubj	2
6	**has**	**4**	**rcmod**	**2**
7	an	9	det	2
8	excellent	9	amod	2
9	**CPU**	6	dobj	**2**
10	to	11	aux	3
11	**implement**	**9**	**vmod**	**3**
12	the	13	det	3
13	algorithm.	11	dobj	3

Index of word/triple	word	Index of dependant	Relation	Index of EDU
14	**We**	**17**	**nsubj**	**4**
15	do	17	aux	4
16	not	17	neg	4
17	**need**	0	ROOT	**4**
18	a	18	det	4
19	**computer**	17	dobj	**4**
20	that	21	nsubj	5
21	**has**	**19**	**rcmod**	**5**
22	an	24	det	5
23	excellent	24	amod	5
24	GPU.	21	dobj	5

u_{10}=<has, computer, rcmod>
V_{10}={6, 21}
left(6)=1 left(21)=4
right(6)=2 right(21)=5

u_{15}=<We, need, nsubj>
V_{15}={1, 14}
left(1)=1 left(14)=4
right(1)=1 right(14)=4

u_{20}=<implement, CPU, vmod>
V_{20}={11}
left(11)=2
right(11)=3

Figure 2: Examples of head-modifier-relation triples.

of the j-th EDU in $\mathcal{P}$ is defined as follows:

$$w_j = C(p_j). \quad (19)$$

$C()$ returns the number of extractive reference summaries that contain p_j, *i.e.*, the maximum score of $C(p_j)$ is K and its minimum score is 1. Since all EDUs in the source documents are assigned an integer score in the range of 0[1] to K, the scoring can be regard as a variant of relative utility score (Radev and Tam, 2003).

3.4 Automatic Scoring of Summaries

Based on the pyramid, we compute a score for a summary by aligning EDUs in the pyramid and EDUs in the system summary. By following PEAK, we find the optimal alignment by solving the Linear Assignment Problem. That is, we compute all similarity scores between EDUs in the summary and pyramid and then find the maximal score so that each EDU in the system summary is matched to at most one EDU in the pyramid. The ILP formulation of the problem is as follows:

$$\text{maximize} \quad \sum_{i=1}^{|\mathcal{C}|}\sum_{j=1}^{|\mathcal{P}|} g(c_i, p_j) w_j \alpha_{i,j} \quad (20)$$

$$\text{s.t.} \quad \sum_{i=1}^{|\mathcal{C}|} \alpha_{i,j} \leq 1 \qquad \forall i \quad (21)$$

$$\sum_{j=1}^{|\mathcal{P}|} \alpha_{i,j} \leq 1 \qquad \forall j \quad (22)$$

$$\alpha_{i,j} \in \{0, 1\} \qquad \forall i, j \quad (23)$$

$\mathcal{C}$ is the set of EDUs in the system summary. Function $g()$ indicates a binary function based on the similarity score between EDUs as follows:

$$g(c, p) = \begin{cases} 1 & \phi(c, p) \geq t \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

We set $t = 0.55$ in our experiments (Section 4).

$\alpha_{i,j}$ is a binary indicator, $\alpha_{i,j} = 1$ denotes that the i-th EDU in the system summary is aligned to j-th EDU in the pyramid.

The optimal solution of the objective function in the ILP problem (19)-(22) is not normalized. Since the unnormalized score is not suitable for comparing systems, we propose a normalization method. To normalize the score to satisfy the range of 0 to 1, we divide the score by the maximum score of sum of the EDU weights. Since every EDU in the pyramid has both length (the number of the word) and weight, the maximum score is derived by solving the *knapsack problem*:

$$\text{maximize} \quad \sum_{j=1}^{|\mathcal{P}|} w_j x_j \quad (25)$$

$$\text{s.t.} \quad \sum_{j=1}^{|\mathcal{P}|} \ell(p_j) \leq L_{\max} \quad (26)$$

[1]The EDUs that are not included in pyramid have scores of zero.

	2003	2004	2005	2006		2007	
Manual evaluation	Cov.	Cov.	Resp.	Resp.	Pyr.	Resp.	Pyr.
# of Topics	30	50	50	50	20	45	23
# of Systems	16	17	32	35	22	32	13
# of References	4	4	4,9	4		4	
Summary length	100	100	250	250		250	
Summary type	Generic	Generic	Query-focused	Query-focused		Query-focused	

Table 1: Statistics of the data sets (DUC-2003 to 2007).

	2008	2009	2010	2011
Manual evaluation	Resp. Pyr.	Resp. Pyr.	Resp. Pyr.	Resp Pyr.
# of Topics	48	44	46	44
# of Systems	58	55	43	50
# of References	4	4	4	4
Summary length	100	100	100	100
Summary type	Update Initial Update	Update Initial Update	Guided Initial Update	Guided Initial Update

Table 2: Statistics of the data sets (TAC-2008 to 2011).

$$x_j \in \{0, 1\} \qquad \forall j \qquad (27)$$

x_j is a binary indicator, and $x_j = 1$ denotes that the j-th EDU is included in the *knapsack*.

Finally, the score is defined as $\text{Pyramid}(\mathcal{P}, S) = \text{OTP}_{\text{LAP}}/\text{OPT}_{\text{KP}}$. OPT_{LAP} and OPT_{LAP} denote maximum score of Equation (20) and maximum score of Equation (25), respectively.

4 Experiments

To investigate the effectiveness of our automatic evaluation method, we compare the correlation coefficients yielded by our methods with those obtained from strong baselines, ROUGE-2, ROUGE-SU4 and BE. We employ ROUGE toolkit version 1.5.5 to compute ROUGE/BE scores and Stanford Parser (de Marneffe et al., 2006) to obtain head-modifier-relation triples. In addition, we examine two types of oracle summaries for our method. One is ROUGE-2-based, the other is BE-based.

We evaluate automatic evaluation measures by Pearson's correlation r, Spearman's rank correlation ρ and Kendall's rank correlation τ. Correlation coefficients are computed by average automatic score and average manual evaluation score for all topics.

4.1 Data Sets

We conducted experiments on the data sets developed for multi document summarization tasks in DUC-2003 to 2007 and TAC-2008 to 2011. Table 1 and Table 2 show the statistics of the data sets.

DUC-2003 and 2004 were used for a generic summarization task with 100 word limit; mean coverage was used in a manual evaluation. DUC-2005 to 2007 were used for a query-focused summarization task with 250 word limit; responsiveness was used in a manual evaluation. The number of topics varied from 30 to 50 and the participating systems from 16 to 35. Note that the pyramid method was applied to small sets of topics in DUC-2006 and 2007.

TAC-2008 and 2009 were used for an update summarization task while TAC-2010 and 2011 were employed for a guided summarization task. For both tasks, the participating systems required two types of summaries, *initial summary* and *update summary* with 100 word limit. Both pyramid method and responsiveness were used in manual evaluations. In particular, TAC-2008 to 2011 have large numbers of participating systems, from 44 to 48.

4.2 EDU Segmenter

We regard decomposing a sentence into EDUs as a sequential tagging problem and implement a neural EDU segmenter that classifies each word in a sentence as the boundary of EDU or not based on 3-layer bi-LSTM (Wang et al., 2015). The size of word embeddings and hidden layers of the LSTM were set to 100 and 256, respectively. To handle low-frequency words, all words are encoded to 40 dimension hidden state by using character-based bi-LSTM (Lample et al., 2016). To utilize entire words in a corpus, we integrated word dropout (Iyyer et al., 2015) into our models with smoothing rate, 1.0. Moreover, to avoid overfitting the training data, dropout layer was adopted to the input of the LSTMs with the ratio 0.3.

The segmenter was trained by utilizing the training data of RST Discourse Treebank corpus (Carlson et al., 2001). The macro-averaged F-measure of boundary detection on the test data of the corpus is 0.917. The source documents, system summaries and reference summaries utilized

	2003	2004	2005	2006		2007	
	Cov.	Cov.	Resp.	Resp.	Pyr.	Resp.	Pyr.
ROUGE-2	.906/.821/.617	.909/.838/.691	.932/.931/.792	.836/.767/.584	.905/.884/.740	.880/.873/.715	.979/.989/.949
ROUGE-SU4	.782/.774/.600	.854/.772/.559	.925/.893/.731	.849/.790/.601	.885/.850/.706	.835/.832/.650	.961/.973/.897
BE	.927/.862/.617	.936/.868/.721	.897/.867/.714	.834/.757/.584	.883/.837/.680	.891/.890/.732	.982/.973/.897
PEAK	–	–	–	.617/.640/–	.508/.538/–	–	–
Prop(BE)	.936/.909/.750	.929/.892/.750	.845/.819/.657	.786/.716/.516	.877/.833/.687	.885/.881/.715	.936/.967/.897
Prop(ROUGE)	.908/.874/.750	.938/.814/.676	.864/.809/.629	.740/.670/.465	.871/.818/.662	.853/.845/.679	.943/.951/.872
Prop(AL)	.831/.841/.633	.904/.855/.735	.821/.757/.567	.762/.667/.465	.801/.772/.584	.814/.793/.610	.958/.962/.872

Table 3: Evaluation results from DUC-2003 to 2007.

		Inital		Update	
		Pyr.	Resp.	Pyr.	Resp.
2008	ROUGE-2	.908/.909/.757	.830/.868/.677	.943/.942/.800	.910/.888/.728
	ROUGE-SU4	.888/.885/.733	.803/.834/.636	.926/.933/.783	.902/.895/.725
	BE	.913/.903/.732	.817/.818/.627	.944/.939/.799	.913/.880/.712
	Prop(BE)	.926/.905/.734	.867/.852/.663	.940/.918/.779	.922/.899/.736
	Prop(ROUGE)	.895/.891/.708	.851/.840/.648	.912/.871/.702	.901/.872/.699
	Prop(AL)	.833/.792/.598	.779/.794/.602	.929/.895/.746	.909/.905/.750
		Pyr.	Resp.	Pyr.	Resp.
2009	ROUGE-2	.911/.950/.823	.757/.844/.674	.939/.896/.742	.717/.755/.600
	ROUGE-SU4	.920/.925/.786	.767/.805/.631	.939/.857/.701	.729/.719/.568
	BE	.856/.931/.784	.692/.838/.670	.924/.929/.798	.695/.816/.670
	Prop(BE)	.867/.932/.782	.854/.848/.670	.855/.917/.782	.866/.810/.656
	Prop(ROUGE)	.886/.917/.770	.858/.819/.639	.864/.890/.741	.822/.735/.586
	Prop(AL)	.901/.872/.689	.881/.821/.666	.886/.857/.704	.830/.743/.594
		Pyr.	Resp.	Pyr.	Resp.
2010	ROUGE-2	.978/.917/.787	.967/.924/.801	.963/.911/.758	.953/.890/.742
	ROUGE-SU4	.968/.947/.830	.954/.952/.837	.910/.885/.727	.900/.878/.727
	BE	.965/.942/.817	.943/.907/.749	.953/.911/.775	.928/.872/.713
	Prop(BE)	.949/.872/.713	.953/.867/.720	.954/.912/.764	.957/.913/.774
	Prop(ROUGE)	.952/.854/.673	.959/.859/.702	.938/.873/.713	.936/.860/.711
	Prop(AL)	.928/.882/.697	.929/.891/.720	.898/.853/.676	.900/.845/.691
		Pyr.	Resp.	Pyr.	Resp.
2011	ROUGE-2	.955/.888/.734	.930/.776/.592	.862/.789/.616	.879/.831/.665
	ROUGE-SU4	.976/.888/.726	.943/.778/.585	.857/.824/.642	.892/.865/.689
	BE	.934/.900/.736	.903/.757/.554	.880/.828/.670	.842/.783/.610
	Prop(BE)	.905/.857/.690	.917/.832/.640	.891/.880/.693	.889/.868/.694
	Prop(ROUGE)	.925/.883/.708	.924/.847/.673	.864/.864/.689	.870/.862/.683
	Prop(AL)	.934/.891/.713	.920/.792/.618	.843/.787/.601	.865/.799/.607

Table 4: Evaluation results from TAC-2008 to 2011.

in our experiments were decomposed into EDUs by the segmenter.

4.3 Results and Discussion

Table 3 and 4 list the correlation coefficients between manual evaluation and automatic evaluation for DUC-2003 to 2007 and TAC-2008 to 2011, respectively. In the tables, the coefficients are written in the order "Pearson's r/ Spearman's ρ/ Kendall's τ". The rows of Prop(BE), Prop(ROUGE) and Prop(AL) denote our method with BE-based oracle summaries as extractive reference summaries, with ROUGE-2-based oracle summaries, and with extractive reference summaries based on alignment, respectively. "Cov.", "Resp." and "Pyr." denote mean coverage, responsiveness and manual pyramid, respectively.

With regard to mean coverage on DUC-2003 to 2004, Prop(BE) achieved the best correlation coefficients. The correlation coefficients indicate very strong correlation with the manual evaluation. Prop(ROUGE) and Prop(AL) attained comparable correlation coefficients with the baseline methods. The correlation coefficients still indicate strong correlation.

With regard to responsiveness, our methods achieved lower correlation coefficients on DUC-2005 to 2007 than on DUC-2003 to 2004. Although our methods are outperformed by the baseline methods, both r and ρ of Prop(BE) still ex-

ceed 0.8 except for responsiveness on DUC-2006. Since our methods mimic manual pyramid evaluation, correlation coefficients against manual pyramid on DUC-2006 to 2007 are better than those against responsiveness and the scores are comparable to those of the baselines.

Moreover, we compare our methods with PEAK on the DUC-2006 data set. For manual pyramid, r and ρ are 0.508 and 0.538, respectively, while for responsiveness they are 0.617 and 0.640, respectively. These scores are significantly lower than those attained by our methods and baselines. Note that these results are obtained by running the code from the author's web page `http://www.larayang.com/peak/`. The results demonstrated that our methods are superior to PEAK.

For manual pyramid on TAC-2008 to 2011, all methods attained quite strong correlation. The scores achieved were around 0.9 and better than those on DUC-2003 to 2007. In particular, Prop(BE) achieved the best scores in some cases. Although, responsiveness yielded lower correlation coefficients than manual pyramid, Prop(BE) still retains strong correlation , e.g., ρ are in the range of 0.857 to 0.932 against manual pyramid, 0.810 to 0.913 against responsiveness. The average correlation coefficients across all data sets on TAC are shown in Table 5. The average correlation coefficients of Prop(BE) slightly lower than those of ROUGE-2 and BE against manual pyramid. On the ohter hand, Prop(BE) achieved the best correlation coefficients against responsiveness. The results imply that Prop(BE) achieves comparable performance to baseline methods.

In a comparison of our methods, Prop(BE) attained the best results while Prop(ROUGE) showed better results than Prop(AL) in many cases. These results imply that extractive oracle summaries are helpful as extractive reference summaries and BE is better objective function to generate them.

In addition, we show SCUs and corresponding EDUs obtained from a human-made pyramid and Prop(BE) in Figure 3. They are obtained from topic "Earthquake Sichuan (ID:D1110B)" from TAC-2011 Guided Summarization Task, the topic type is categorized as "Accidents and Natural Disasters". Summarizers are required to generate a summary that includes following aspects: (1) WHAT: what happend, (2) WHEN: date, time, other temporal placement makers, (3) WHERE: physical location, (4) WHY: reasons for accident/disaster, (5) WHO_AFFECTED: casualties (death, injury), or individuals otherwise negatively affected by the accident/disaster, (6) DAMAGES: damages caused by the accident/disaster, (7) COUNTERMEASURES: countermeasures, rescue efforts, prevention efforts, other reactions to the accident/disaster. From the figure, we can see that the EDUs are not always identical to human-generated SCUs at word-level but are identical at concept-level.

	Pyr.	Resp.
ROUGE-2	.932/.900/.752	.868/.847/.685
ROUGE-SU4	.923/.893/.741	.861/.841/.675
BE	.921/.910/.764	.842/.834/.663
Prop(BE)	.911/.899/.742	.903/.861/.694
Prop(ROUGE)	.905/.880/.713	.890/.837/.668
Prop(AL)	.894/.856/.678	.877/.824/.656

Table 5: Average correlation coefficients across data sets (TAC-2008 to TAC-2011)

In short, these results imply that our methods have at least comparable performance to the baselines. Although our methods are outperformed by the baselines in some cases, the correlation coefficients are high enough against manual evaluation. Moreover, our methods have a significant advantage over the baselines methods because our methods clearly indicate whether the output of the text summarization system failed to include important SCUs. Thus, our automatic pyramid method enhanced with extractive oracle summaries is helpful for further improvement of summarization systems.

5 Conclusion

This paper proposed an automatic pyramid evaluation method that allows us to scrutinize the failure analysis of systems. To construct a pyramid, we transform human-made reference summaries into extractive reference summaries whose atomic units are EDUs obtained from the source documents. Then, we weight every EDU by counting the number of extractive reference summaries that contain the EDU. When evaluating a summary, we determine the correspondences between EDUs in the pyramid to those in the summary by solving Linear Assignment Problem and give a score to the summary based on the correspondences. We also proposed two types of extractive reference summaries. The first is the alignment-based extractive reference summary. The second is the extractive

SCUs obtained from human-made pyramid

$w = 4$ The 7.8-magnitude earthquake struck
$w = 4$ Sichuan Province of China
$w = 4$ No warning signs detected
$w = 4$ Over 8,500 killed
$w = 4$ China allocated 200 million yuan ($29 Million) disaster relief
$w = 1$ Rain is forecast, could hamper relief efforts
$w = 1$ Quake also affected Gansu, Shaanxi provinces, and Chongqing municipality

EDUs obtained from pyramid of Prop(BE)

$w = 1$ The 7.8-magnitude earthquake struck Sichuan province shortly before 2:30 pm
$w = 2$ in aid for earthquake victims in Sichuan Province of China
$w = 4$ Chinese authorities did not detect any warning signs ahead of Monday's earthquake
$w = 1$ leaving at least 12,000 people died
$w = 2$ China has allocated 200 million yuan
$w = 1$ Rain in the coming days in Sichuan is expected to hamper earthquake relief efforts, as well as increase risks of landslides
$w = 1$ 50 in the municipality of Chongqing, 61 in Shaanxi province, and one in southwestern Yunnan

Figure 3: Examples of SCUs obtained from pyramids.

oracle summary.

To demonstrate the effectiveness of our methods, we conducted experiments on DUC-2003 to 2007 and TAC-2008 to 2011 data sets. The results demonstrated that our method yielded results that well correlated with various manual evaluations. The correlation coefficients are at least comparable to those obtained from strong baselines, ROUGE-2, ROUGE-SU and BE and significantly higher than those obtained from previous automatic pyramid evaluation, PEAK.

References

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proc. of the Second SIGdial Workshop on Dialogue and Discourse*, pages 1–10.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366.

Vanessa Wei Feng and Graeme Hirst. 2014. Two-pass discourse segmentation with pairing and global features. *CoRR*, abs/1407.8215.

Seeger Fisher and Brian Roark. 2007. The utility of parse-derived features for automatic discourse segmentation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 488–495.

Aaron Harnly, Ani Nenkova, Rebecca Passonneau, and Owen Rambow. 2005. Automation of summary evaluation by the pyramid method. In *Proc. of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, pages 226–232.

Tsutomu Hirao, Masaaki Nishino, Jun Suzuki, and Masaaki Nagata. 2017. Enumeration of extractive oracle summaries. In *Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 386–396.

Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. 2006. Automated summarization evaluation with basic elements. In *Proc. of the 5th International Conference Language Resource and Evaluation (LREC06)*, pages 899–902.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1681–1691.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Junyi Jessy Li, Kapil Thadani, and Amanda Stent. 2016. The role of discourse units in near-extractive summarization. In *Proc. of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 137–147.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proc. of Workshop on Text Summarization Branches Out*, pages 74–81.

Mann, William Charles and Thompson, Sandra Annear. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *In Proceedings of International Conference on Language Resources and Evaluation (LREC)*, pages 449–454.

Hajime Morita, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2013. Subtree extractive summarization via submodular maximization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1023–1032.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proc. of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics:HLT-NAACL 2004*, pages 145–152.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2).

Rebecca J. Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. 2013. Automated pyramid scoring of summaries using distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 143–147.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1341–1351.

Dragomir R. Radev and Daniel Tam. 2003. Summarization evaluation using relative utility. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 508–511.

Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. 2015. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *CoRR*, abs/1510.06168.

Qian Yang, Rebecca Passonneau, and Gerard de Melo. 2016. Peak: Pyramid evaluation via automated knowledge extraction. In *Proc. of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 2673–2679.