Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

PEAK: Pyramid Evaluation via Automated Knowledge Extraction

Qian Yang
IIIS, Tsinghua University
Beijing, China
laraqianyang@gmail.com

Rebecca J. Passonneau
CCLS, Columbia University
New York, NY, USA
becky@ccls.columbia.edu

Gerard de Melo
IIIS, Tsinghua University
Beijing, China
gdm@demelo.org

Abstract

Evaluating the selection of content in a summary is important both for human-written summaries, which can be a useful pedagogical tool for reading and writing skills, and machine-generated summaries, which are increasingly being deployed in information management. The pyramid method assesses a summary by aggregating content units from the summaries of a wise crowd (a form of crowdsourcing). It has proven highly reliable but has largely depended on manual annotation. We propose PEAK, the first method to automatically assess summary content using the pyramid method that also generates the pyramid content models. PEAK relies on open information extraction and graph algorithms. The resulting scores correlate well with manually derived pyramid scores on both human and machine summaries, opening up the possibility of wide-spread use in numerous applications.

1 Introduction

The capability to summarize one or more documents is a vital skill for people, and a significant research goal in natural language processing (NLP). Summarization is important in education, as it has been found to be one of the best instruction strategies to improve students' reading and writing skills (Graham and Perin 2007). The importance of automated summarization is reflected in the rate of papers on summarization at four major NLP conferences (ACL, EACL, EMNLP and NAACL) over the past six years, which has consistently been around 5% of the total. Reliable assessment of summaries is therefore beneficial both for people learning to summarize, and for progress in automated summarization methods. A defining characteristic of a summary is that it should condense the source text, retain important information, and avoid redundancy. Thus content assessment is critical. This paper presents a method to automate a pre-existing summary content assessment technique called the Pyramid method (Nenkova, Passonneau, and McKeown 2007), that was developed to evaluate abstractive summarization (where the source text is rewritten), that has often been used to evaluate extractive summarizers (where source sentences are selected verbatim), and that has been applied to student summaries.

ROUGE (Lin and Hovy 2004) is widely used to assess content selection of automated summarizers (Nenkova and McKeown 2011; Gupta and Lehal 2007) because it is fully automated, easy to use, and evaluation of sets of summaries correlates well with human scores. It is often supplemented with pyramid scores due to their greater reliability and better diagnostic properties (Nenkova and McKeown 2011). Since human-written summaries typically share some content, but are unlikely to have identical content, both methods compare each assessed summary to a set of multiple reference summaries written by humans. ROUGE compares n-grams from the target summaries to the reference summaries and computes a recall measure. It cannot capture similarity of meaning when there is little lexical overlap, and cannot recognize when lexical overlap does not correspond to similar meaning. The pyramid method not only avoids these problems, but also distinguishes levels of importance among content through a wisdom-of-the-crowd method that is inherently semantic and pragmatic. When assessing individual summaries, ROUGE scores are not accurate (Louis and Nenkova 2009), while pyramid scores provide accurate and diagnostically valuable scores. Application of the method has, however, largely depended on manual annotation.

The pyramid method relies on an emergent importance weighting. Annotators identify content that recurs across a set of human-written reference summaries and create clusters of similar phrases to represent Summary Content Units (SCU). Each SCU must have no more than one contribution from each reference summary.[1] Each SCU, which is given a mnemonic label by the annotators, then has a weight corresponding to the number of contributing reference summaries to represent the relative importance of the content. Note that assigning differential importance to source content has often been used in studies of student summaries as a reading or writing intervention (Brown and Day 1983). The method captures alternate forms that express the same meaning, which can be as various as a nominalization versus a clause, as in "the condensation of water vapor into droplets" versus "the water vapor condenses into water drops," as well as lexical variation, as in "Indian sovereignty allows them to run casinos" versus "Reservations can set up casinos." The

[1] Redundant information, which rarely appears in reference summaries, is ignored.

pyramid content model is then used to annotate the content in the target summaries, and the sum of the SCU weights serves as a raw score. The score can be normalized analogous to precision to measure how much of the content in a summary is as important as it could be, or analogous to recall to measure how much of the important content in an average reference summary is expressed in the target summary.

Tests of the pyramid method indicate that the scores are reliable given four to five reference summaries (Nenkova, Passonneau, and McKeown 2007), and that the annotation is reliable based on interannotator agreement, and on evidence that independent application of the method by different annotators produces virtually the same ranking of systems (Passonneau 2010). After 2004, the pyramid method replaced an earlier method of content selection at the annual Document Understanding/Text Analysis Conferences (DUC and TAC) that have carried out large scale evaluations of summarizers since 2000 (Nenkova and McKeown 2011).

The lack of automation is the biggest barrier to more widespread use. An automated scoring procedure tested on student summaries was found to correlate well with a highly reliable main ideas score from a reading and writing intervention, to correlate well with manual pyramid scores, and to have adequate recall and precision of SCUs (Passonneau et al. 2013). This automated scoring has recently been used to evaluate an abstractive summarizer (Bing et al. 2015). Full automation would facilitate more widespread use for system development or student feedback. The method presented here is the first to automate both the construction of pyramid models and assignment of pyramid scores.

In this paper, we propose PEAK (Pyramid Evaluation via Automated Knowledge Extraction). Our approach to pyramid construction relies on open information extraction to identify subject-predicate-object triples, and on graphs constructed from the triples to identify and assign weights to salient triples. For scoring, we rely on the Munkres-Kuhn bipartite graph algorithm to find the optimal assignment of model SCUs to target summaries. Our results show that PEAK scores correlate very well with the manual Pyramid method. We even encounter examples where PEAK fares better than humans in assigning weights to SCUs.

2 Related Work

The traditional way to evaluate summaries is for human assessors to judge each summary individually for aspects such as readability and informativeness. To promote more consistent ratings, from 2001 to 2003 the Document Understanding Conference relied on a single human-written model summary as a yardstick, against which the human assessors would assess the automatic summaries. This process of individually scoring every summary is, of course, very time-consuming and does not scale well in the number of summaries for a given summarization task. It was also found to be overly sensitive to the choice of reference summary.

An alternative is to have humans produce a small set of model summaries for the source text(s), and to rely on automated methods to score all candidate target summaries. Using multiple model summaries leads to more objective and consistent content assessments. The well-known ROUGE metric (Lin 2004) automatically compares n-grams in each target with those of the model summaries, and can be applied to a large number of summaries for a given summarization task. Studies show that ROUGE fares reasonably well when averaging across 10 or more summaries, as e.g. in summarization system evaluation, but unfortunately tends not to be reliable enough to assess an individual summary (Gillick 2011), as would be required, for instance, in a classroom setting. Given the model summaries, one can easily create summaries that obtain very high ROUGE scores but are nonsensical (Gillick 2011). ROUGE scores of state-of-the-art summarization systems match those of human-written summaries, although it is quite apparent that the human ones are of significantly higher quality (Gillick 2011).

The pyramid method was first proposed in Nenkova and Passonneau (2004), with more detail in Nenkova, Passonneau, and McKeown (2007). The manual annotation captures underlying semantic similarity that ROUGE cannot capture. It has high reliability on both pyramid creation and target (peer) annotation (Passonneau 2010). A concurrent automated approach to make summary evaluation more semantic relied on latent semantic analysis (Steinberger and Ježek 2004). Unlike the pyramid method, it does not distinguish elements of content by importance.

Harnly et al. (2005) presented a method to automate pyramid content scoring, given a pre-existing reference pyramid. Specifically, they used similarity measures such as cosine similarity and edit distance of n-grams in target summaries compared with pyramid content units, and relied on Dynamic Programming to determine the overall solution. Their method produced good system rankings, but absolute scores were far lower than the scores from manual annotation, and the SCU identification was poor.

Passonneau et al. (2013) presented an improved automatic scoring method, again using dynamic programming but relying on distributional semantics, and applied it to students' summaries. Results were promising for providing relatively accurate content feedback on individual summaries. Both approaches to automated pyramid scores still first require humans to manually create the pyramids from the model summaries. Our method obviates the need for manually created pyramids. Additionally, our experiments show that our method obtains comparable results on the task of automatic assessment compared to the previous work.

Louis and Nenkova (2009) proposed an automatic method that attempts to entirely avoid any human involvement at all by directly comparing the summary to be evaluated with the original text. Their method works surprisingly well when evaluating a system by averaging over many summaries, but for individual input summaries, the authors often observed low correlations with human-based assessments.

3 The PEAK Method

Overview

In the pyramid method, model summaries are annotated to identify *summary content units (SCUs)*, sets of text fragments that express the same semantic content, typically a single proposition. An SCU has at most one contributor

SCU 49	Plaid Cymru wants full independence
C1	Plaid Cymru wants full independence
C2	Plaid Cymru...whose policy is to...go for an independent Wales within the EC
C3	calls by...(Plaid Cymru)...fully self-governing Wales within the EC
C4	Plaid Cymru...its campaign for equal rights to Welsh self-determination

Figure 1: Sample SCU from *Pyramid Annotation Guide: DUC 2006*. Four model summaries contribute to an SCU with the mnemonic label *Plaid Cymru wants full independence*. (Note that the label captures what the annotator finds in common across the contributors; it plays no role in use of the pyramid for assessment).

phrase from each model summary. The SCU weight is the number of contributor models, which ranges from 1 to N. Figure 1 shows an example SCU from a webpage of guidelines used in DUC 2006. It shows that contributor phrases to the same SCU can have distinct lexical and syntactic realizations of the same semantics. Here the weight is four. SCU weight induces a partition over the SCUs from a given set of reference summaries. With many models, it can be observed that the sizes of the equivalence classes in descending order of weight have a Zipfian distribution: a few SCUs occur in all models, many occur in most, and a long tail of SCUs occur in only one or two of the model summaries express (Nenkova, Passonneau, and McKeown 2007).

To score a target summary against a pyramid, annotators mark spans of text in the target that express an SCU, and the SCU weights increment the raw score for the target. If different model summaries of the same source text are used, the set of SCUs and their weights will be different. Three different methods to test how many model summaries are required for scores to be stable and reliable all provided evidence that four to five models are sufficient (Nenkova, Passonneau, and McKeown 2007). The raw pyramid scores have various normalizations. A precision analog (used in DUC 2005 (Passonneau et al. 2005)) normalizes the summed weights of a set of SCUs by the maximum sum that the same number of SCUs can have, based on the number of SCUs of each weight in the pyramid. A recall analog (used in DUC 2006 (Passonneau et al. 2006)) normalizes by the maximum sum for the average number of SCUs in the model summaries. Finally, the harmonic mean of these two scores is an f-measure analog (used in Passonneau et al. 2013).

In manual pyramid annotation, the annotators iterate over the process until they are satisfied with the semantic content for each unit, and the contributor assignments. PEAK is designed to produce SCUs with the same properties: a coherent semantics for each SCU expressed in each contributor, and only one contributor per model summary. Summaries typically have complex sentences, so one summary sentence often contributes to more than one SCU. To detect candidate propositions in the model sentences, we use open information extraction to identify relation triples. We assess semantic similarity of triples based on an analysis of a hypergraph where the nodes are the elements of triples, the three nodes of a triple are connected by a hyperedge, and nodes in different hyperedges can be connected by weighted edges that represent their semantic similarity. The next three subsections present the hypergraph, show how it is used to generate a pyramid, and explain how the resulting pyramid is used to score target summaries.

SCU Identification

Due to the condensed nature of human summaries, they often contain complex sentences. For instance, consider: "*The law of conservation of energy is the notion that energy can be transferred between objects but cannot be created or destroyed.*" It expresses two salient ideas: that energy can be transferred between objects, and that energy cannot be created. Open information extraction (Open IE) methods extract so-called subject-predicate-object triples, in which the subject, predicate, and object are natural language phrases extracted from the sentence, and which often correspond to syntactic subject, predicate and object. For example, "*These characteristics determine the properties of matter*", yields the triple ⟨*These characteristics*, *determine*, *the properties of matter*⟩. While Open IE extracts individual propositions from text, it can produce partial duplicates (see Figure 5) and occasional noise, such as ⟨*the matter*, *itself*, ∅⟩. Our current implementation relies on the ClausIE system (Del Corro and Gemulla 2013) for Open IE.

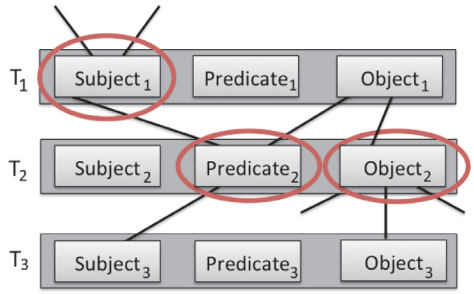

Figure 2: Hypergraph to capture similarites between elements of triples, with salient nodes circled in red

As illustrated in Figure 2, we create a hypergraph $G = (V, H, E)$ where the nodes V correspond to all the subject, predicate and object elements of the triples (inner boxes in Figure 2), every set of nodes from the same triple (e.g., T_1) is connected by a hyperedge $h \in H$ (shaded boxes), and nodes not connected by a hyperedge can be connected by edges $e \in E$. Edges e are weighted by similarity scores $\mathrm{sim}(u, v)$ between two nodes u,v. These are obtained from *Align, Disambiguate and Walk* (ADW) (Pilehvar, Jurgens, and Navigli 2013), a state-of-the-art approach to semantic similarity of text. A pair of nodes u and v will have an edge if and only if their similarity $sim(u, v) \geq t$. We picked the midpoint of 0.5 as the threshold t for two nodes to be more similar than not.

In Figure 2, salient nodes have been circled, based on the following definition.

WEIGHT: 4
ANCHOR: "Matter" "is" "all the objects and substances"

FROM SENTENCE1 of CONTRIBUTOR1: Matter is all the objects and substances that take up space around us.
SALIENT NODES: "Matter" "all the objects and substances"

CONTRIBUTOR1: "Matter" "is" "all the objects and substances"
CONTRIBUTOR2: "Matter" "is identified" "as being present everywhere and in all substances"
CONTRIBUTOR3: "The author of this passage titled What is Matter" "defines" "matter as the stuff that all objects and substances in the universe are made of"
CONTRIBUTOR4: "Matter" "is" "what makes up all objects or substances and contains both volume and mass"

Figure 3: SCU created by PEAK

Definition 3.1. The set of *salient nodes* V_{S} is defined as

$$V_{\mathrm{S}} = \{v \in V \mid \deg(v) \geq d_{\min}\}, \tag{1}$$

where $d_{\min}$ is a pre-defined threshold, representing nodes that have enough 1-degree neighbors to yield moderate- to high-weight SCUs. The defined maximum weight is 4 (the number of model summaries - 1), but in the experiments we find the maximum weight can be greater than 4, due to repetition in the model summaries. So we set $d_{\min}$ to 3, which is slightly bigger than the midpoint of the regular maximum weight, meaning that nodes with degree ≥ 3 are chosen as salient. We believe this reflects the way humans make such assessments.

As potential SCUs, we consider all triples where at least two of the three elements are in V_{S}. For the final set of SCUs, we merge near-equivalent SCUs extracted from the same sentence. This is because ClausIE's open information extraction method decomposes sentences into semantic triples where one contains the other, as in ⟨*Energy*, *is*, *the property of matter*⟩ and ⟨*Energy*, *is*, *the property*⟩ (see Figure 5). We also merge similar triples from different sentences, as described in Section 3.

Pyramid Induction

After the identification of salient triples, the next step is to align triples extracted from distinct model summaries into candidate SCUs. For this, we propose a matching algorithm based on the notion of a *similarity class*.

Consider the example in Figure 3. Here, we have an SCU induced from one of the salient triples: ⟨*Matter*, *is*, *all the objects and substances*⟩. We treat each next salient triple in turn as an anchor for an SCU, and identify contributors that are semantically similar to the anchor by creating a similarity class for each salient node of the triple.

Definition 3.2. The *Similarity Class* $E(v)$ of a node $v \in V_{\mathrm{S}}$ is defined as

$$E(v) = \{u \in V \mid (u, v) \in E\}, \tag{2}$$

i.e., the one-degree neighbors of v, or those nodes $u \in V$ such that $u \neq v$ and $\mathrm{sim}(u, v) \geq 0.5$).

We create a similarity class E_i for every salient node in an anchor triple. In our example, the subject and object nodes are salient, and we create E_1 for "*Matter*" and E_2 for "*all the objects and substance*", as shown in Figure 4.

Similarity Class E1 for "Matter"	Similarity Class E2 for "all the objects and substances"

E1 = {
"Matter",
"All matter",
"matter",
"the matter itself",
"a different matter",
"the matter itself systematically", ...
}
	E2 = {
	"all the objects and substances",
	"the substance",
	"all objects and substances in the universe",
	"what makes up all objects or substances and contains both volume and mass",
	"as being present everywhere and in all substances", ...
	}

Figure 4: Similarity Class

A triple T_i from a summary S_i when serving as an anchor triple is a candidate contributor to a potential SCU. The similarity classes of the nodes in T_i provide a mechanism to find additional contributor triples from model summaries S_j distinct from S_i. Any sentence from a model summary S_j that yields a triple T_j such that two nodes u, v in T_j are in distinct similarity classes E_i and E_j for anchor T_i will be a *potential contributor*. Any given model summary must contribute at most once to a given SCU. Therefore, for each model summary S_j distinct from S_i, we need to select the best contributor triple T_j from possibly multiple candidates extracted from S_j. We compute similarity scores for each node in an anchor T_i to each member of the node's similarity class, and choose an optimal assignment based on maximizing the similarity of a candidate T_j to the anchor T_i.

Given an SCU anchor triple T_i with subject s, predicate p, object o, the similarity classes for s, p and o are E_s, E_p and E_o. For every model summary, we only consider as *potential contributors* c_i those triples $\langle s_i, p_i, o_i \rangle$ where the majority, i.e., two or three, of $\{s_i, p_i, o_i\}$ are in at least two of E_s, E_p and E_o. From the set of *potential contributors* T_j for a given summary S_j, we find the highest ranking contributor c_{max}:

$$\begin{array}{lllll} \max\limits_{i} & \sum_i similarityScore(x_i, y_i) & & & \\ \text{s.t.} & x_i \in \{s_i, p_i, o_i\} & & & \\ & y_i = s & & if \quad x_i \in E_s & \\ & y_i = p & & if \quad x_i \in E_p & \\ & y_i = o & & if \quad x_i \in E_o & \end{array} \tag{3}$$

The total number of contributors for an SCU s provides the weight w_s of the SCU. For the example SCU in Figure 3, the weight of the candidate SCU is 4 because there are 4 contributors, including the anchor. For convenience, we represent an SCU as its anchor and its weight, omitting the list of contributors that would appear in the manual annotation. Note that each next contributor triple to a candidate SCU has a turn as an anchor. For a candidate SCU that has n contributors, there will be at least n variants of the same

SCU. We merge similar candidate SCUs into a single SCU using Algorithm 1. At this stage of pyramid construction, the goal is a high precision of hypothesized SCUs, both in the total number of SCUs and in their weights. The value of T_1 affects both outcomes, which are interdependent. We experimented with values of 0.7 and below and found that at 0.7, there were too few SCUs and at 0.9, the SCU weights were too low. So in our experiments, T_1 is fixed at 0.8.

Algorithm 1 Merge similar SCUs

procedure MERGE(SCU anchors, weights)
  set a graph G whose nodes are all SCU anchors
  set threshold T_1
  for each node $anchor_m$ **do**
    for each node $anchor_n$ **do**
      calculate $similarityScore_{m,n}$
      if $similarityScore_{m,n} \geq T_1$ **then**
        add edge between $anchor_m$ and $anchor_n$
  $mergedSCU \leftarrow$ the connected component in G
  $mergedWeight \leftarrow$ max. weight of connected component
  Return $mergedAnchor, mergedWeight$

Automated Scoring of Summaries

We can now use the pyramids created by PEAK to score target summaries written by humans (e.g., students) or machines. For this, we again use our semantic triple-based formalism. One advantage of such a formalism is that such an explicit representation can support the generation of assessment feedback, as well as any other downstream processes that may benefit from such information. For approaches based on distributional semantics, such as the matrix-based one of Passonneau et al. (2013), this may be more challenging.

All living things, man-made, or from mother nature itself are made out of matter. Also, because matter takes up space and certain amount of material, it can be detected and measured. Volume is what matter takes up as space; mass is what certain amount of material the object has. Because a cork is hard, it will break when you break it, and clay will flatten when bended. But when you drop both in water, the cork will float, and the clay will sink. These characteristics determine the properties of matter. Physical properties are what the object looks like without changing anything about it. Chemical properties describe matter when it is changed with a different kind of property. Energy is the property of matter. Energy can be transferred to one object to another, energy cannot be created or destroyed, this rule is called the law of conservation of energy. When an object has a motion or position, it is called mechanical energy. The energy an object has when it is moving is called kinetic energy. The energy an object has when it has a position or shape is called a potential energy.

1 "All living things" "itself are made out" "of matter"
1 "itself" "All living things are made out" "of matter"
.....
9 "Energy" "is" "the property of matter"
9 "Energy" "is" "the property"
.....

Figure 5: Open Information Extraction from target summaries

We again rely on open information extraction using ClausIE to obtain triples from the target summaries. Fig. 5 shows a target summary, and a list of triples along with the sentence number for each triple. The three parts of the triple (subject, predicate, and object) are concatenated into a single string, called a *label*. A target summary will have one label for each triple.

Recall that our pyramid model consists of s SCUs (anchors) with associated weights w_s. Every target summary label t is compared with every SCU s in the automatically generated pyramid. We again use ADW for the similarity computation. An SCU s here may be a merged one, so it may contain several triples, possibly with distinct weights. We compare t with all the triples in s, storing their similarity scores. Finding the maximal score for a target summary and ensuring that every t is matched to at most one s amounts to solving a maximal matching problem. We use the Munkres-Kuhn algorithm, as described in Algorithm 2. In our experiments, T is fixed to 0.6.

Algorithm 2 Computing scores for target summaries

procedure SCORE(target summary sum)
  for each sentence s in sum **do**
    $T_s \leftarrow$ triples extracted from s
  for each triple $t \in \bigcup T_s$ **do**
    for each SCU s with weight w **do**
      $m \leftarrow$ similarity score between t and s
      if $m \geq T$ **then**
        $W[t][s] \leftarrow w$ ▷ store weight
  S $\leftarrow$ Munkres-Kuhn (Hungarian) Algorithm(W)
  Return S

4 Experiments

Student Summaries

Our experiments focus on a student summary dataset from Perin et al. (2013) with twenty target summaries written by students. For this data, the study by Passonneau et al. (2013) had produced five reference model summaries, written by proficient native speakers of English, and two manually created pyramids, each from a different annotator. We use the reference summaries as input to PEAK in order to have it automatically generate a pyramid P. Subsequently, this pyramid is used to score the twenty student summaries. We compare the automatically generated scores with the original scores for those summaries produced by humans, as well as with previous automatic algorithms. For the score comparison, we use the raw (non-normalized) scores.

Table 1 gives the correlations between scores based on P and scores based on one of the two manual pyramids P1 and P2, which were created by different annotators who worked independently with the same five reference summaries.

We see that PEAK produces very strong results using an entirely automatically generated pyramid. The study by Passonneau et al. (2013) evaluated a series of algorithms with different parameter settings, obtaining Pearson's correlations between 0.71 and 0.93. However, their method starts off with the manually created pyramids and only performs the scoring automatically. For comparison, we re-ran PEAK

using the manual pyramids. In Table 1, we also list correlation scores based on P2 (on P1 the results are similar, but slightly lower, as our algorithm performs best with a somewhat larger number of SCUs).

	P1 + M. Scoring	P2 + M. Scoring
P + A.Scoring	0.8263	0.7769
P2 +A. Scoring	0.8538	0.8112
P1 + M. Scoring	1	0.8857

Table 1: Pearson's correlations between scores based on PEAK's pyramid P as well as the two human pyramids P1, P2, with either manual or automatic scoring.

Analysis. For further analysis, we compared the different pyramids themselves. Note that even different human experts may create quite distinct pyramids, but the final scores and ranks can be consistent (Passonneau 2010). When comparing the two manual pyramids P1 and P2, we find that they are indeed rather dissimilar. P1 has 34 SCUs but P2 has 60 SCUs. Still, the Pearson's correlation score between the manual scores based on P1 vs. P2 is 0.8857.

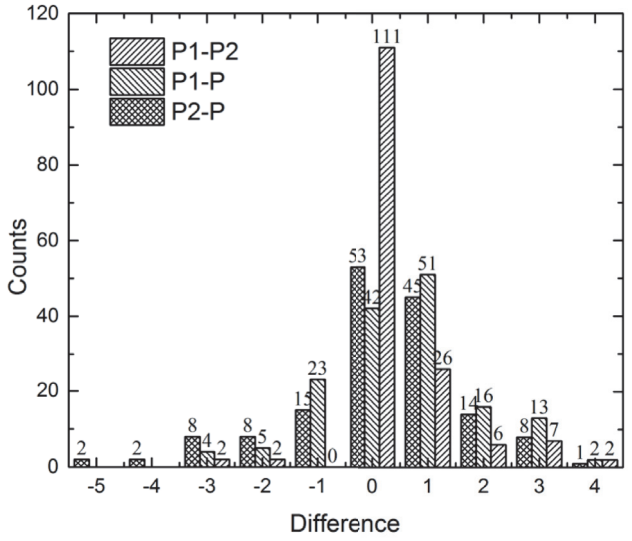

Figure 6: Histogram of weight differences between P1 and P2, P1 and P, P2 and P for every anchor

PEAK's pyramid P consists of 80 SCUs with 156 anchors. The counts of weight differences between P1 and P2, P1 and P, P2 and P for every anchor is depicted in Fig. 6. We find that PEAK identifies surprisingly many of P1's SCUs. Among the 34 SCUs in P1, 32 are matched, for a recall of 94.12%. For P2, the overall recall is just 56.7%. However, for the SCUs with weight 5, 4, or 3 in that pyramid, recall is 91.30%. We mainly miss SCUs with weights of 2 or 1. Fortunately, these lower-weight SCUs are less significant for scoring and ranking. Additionally, PEAK's pyramid contains other SCUs with weight 2 and 1 not considered noteworthy in the manual pyramids. While this results in low precision scores, a detailed analysis of the weight differences reveals that the pyramids are not overly dissimilar. Most of the extra SCUs have a weight of just 1 and hence do not affect the overall ranking. Given the profound differences between the two manual pyramids, we see that direct comparisons between pyramids are not necessarily significant. Instead, the correlation scores reported above appear more meaningful.

Studying PEAK's output in more detail, we observed further benefits of our approach. Relying on Open IE for extraction enables us to cope with multi-faceted sentences, for which we may obtain multiple extractions that constitute separate SCUs. Consider, for instance, "*The law of conservation of energy is the notion that energy can be transferred between objects but cannot be created or destroyed.*" From this sentence, we obtain both ⟨*energy*, *can not be*, *created*⟩ as well as ⟨*energy*, *can be transferred*, *between objects*⟩ as SCUs.

In a few cases, the weights obtained by our approach turn out to be even more accurate than those from humans. For instance, PEAK chooses an SCU ⟨*Matter*, *is*, *all the objects and substances*⟩, which matches SCU "*Matter is what makes up all objects or substances*" in a human pyramid. Comparing the two, PEAK's SCU lacks one contributor from the sentence "*Matter is anything that has mass and takes up space (volume)*". However, PEAK instead adds the corresponding triple to another SCU ⟨*Matter*, *can be measured*, *because it contains volume and mass*⟩. The latter appears to be a much closer match.

Machine-Generated Summaries

For further validation, we also conducted an additional experiment on data from the 2006 Document Understanding Conference (DUC) administered by NIST ("DUC06"). The original data consists of twenty parts, each of which contain four reference summaries and 22 machine-generated summaries with manual scores. Unfortunately, this data contains numerous inaccuracies, requiring manual cleaning. To create an evaluation dataset, we randomly chose one of the twenty parts and asked annotators to follow a set of guidelines to correct the original annotations.[2]

We evaluate PEAK on this data by generating a pyramid based on the four reference summaries, which is then used to score the twenty-two machine-generated summaries. These scores from PEAK are then compared with the manual ones.

The Pearson's correlation score between PEAK's scores and the manual ones is 0.7094.

5 Conclusion

In this paper, we have proposed the first fully automatic version of the pyramid method. Our method not only assesses target summaries but also generates the pyramids automatically. We rely on open information extraction to obtain a more accurate picture of the semantics of sentences, score similarities between nodes in a graph to determine salient SCUs, and develop an approach based on similarity classes to assign the weights for SCUs. Experiments show that our

[2]This data is available from http://www.larayang.com/peak/.

SCUs are very similar to those created by human annotators. Additionally, we present a method for assessing target summaries automatically, again obtaining a high Pearson correlation with human assessors. A distributable code package is available at http://www.larayang.com/peak/.

In terms of future work, we intend to refine the pyramid induction process by handling additional phenomena. For instance, coreference resolution and ideas from semantic parsing (Tandon et al. 2015) could expose further connections between sentences during the information extraction and merging stages.

Overall, our research shows great promise for automated scoring and assessment of manual or automated summaries, opening up the possibility of wide-spread use in the education domain and in information management.

6 Acknowledgments

This work is supported by the National Basic Research Program of China Grants 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grants 61033001, 61361136003, 61450110088.

References

Bing, L.; Li, P.; Liao, Y.; Lam, W.; Guo, W.; and Passonneau, R. J. 2015. Abstractive multi-document summarization via phrase selection and merging. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. Beijing, China: Association for Computational Linguistics.

Brown, A. L., and Day, J. D. 1983. Macrorules for summarizing texts: The development of expertise. *Journal of verbal learning and verbal behavior* 22(1):1–14.

Del Corro, L., and Gemulla, R. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, 355–366.

Gillick, D. J. 2011. *The Elements of Automatic Summarization*. Ph.D. Dissertation, EECS Department, University of California, Berkeley.

Graham, S., and Perin, D. 2007. A meta-analysis of writing instruction for adolescent students. *Journal of educational psychology* 99(3):445.

Gupta, V., and Lehal, G. S. 2007. A survey on automatic text summarization. Literature Survey for the Language and Statistics II course at CMU.

Harnly, A.; Nenkova, A.; Passonneau, R.; and Rambow, O. 2005. Automation of summary evaluation by the pyramid method. In *Proceedings of the Conference of Recent Advances in Natural Language Processing (RANLP*, 226.

Lin, C.-Y., and Hovy, E. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the ACL Text Summarization Branches Out Workshop*, 74–81.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, S. S., ed., *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Louis, A., and Nenkova, A. 2009. Automatically evaluating content selection in summarization without human models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, 306–314. Stroudsburg, PA, USA: Association for Computational Linguistics.

Nenkova, A., and McKeown, K. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval* 5(2-3):102–233.

Nenkova, A., and Passonneau, R. 2004. Evaluating content selection in summarization: The pyramid method. In Susan Dumais, D. M., and Roukos, S., eds., *HLT-NAACL 2004: Main Proceedings*, 145–152. Boston, Massachusetts, USA: Association for Computational Linguistics.

Nenkova, A.; Passonneau, R.; and McKeown, K. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing* 4(2).

Passonneau, R. J.; Nenova, A.; McKeown, K.; and Sigelman, S. 2005. Applying the pyramid method in duc 2005. In *Proceedings of the 2005 Document Understanding Conference (DUC 2005)*.

Passonneau, R. J.; McKeown, K.; Sigelman, S.; and Goodkind, A. 2006. Applying the pyramid method in the 2006 document understanding conference. In *Proceedings of the 2006 Document Understanding Conference (DUC 2006)*.

Passonneau, R. J.; Chen, E.; Guo, W.; and Perin, D. 2013. Automated pyramid scoring of summaries using distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 143–147. Sofia, Bulgaria: Association for Computational Linguistics.

Passonneau, R. J. 2010. Formal and functional assessment of the pyramid method for summary content evaluation. *Natural Language Engineering* 16(02):107–131.

Perin, D.; Bork, R. H.; Peverly, S. T.; and Mason, L. H. 2013. A contextualized curricular supplement for developmental reading and writing. *Journal of College Reading and Learning* 43(2):8–38.

Pilehvar, M. T.; Jurgens, D.; and Navigli, R. 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 (Volume 1: Long Papers)*, 1341–1351. Sofia, Bulgaria: Association for Computational Linguistics.

Steinberger, J., and Ježek, K. 2004. Using latent semantic analysis in text summarization and summary evaluation. In *Proceedings of the 7th International Conference on Information System Implementation and Modeling (ISIM 04)*, 93–100.

Tandon, N.; de Melo, G.; De, A.; and Weikum, G. 2015. Knowlywood: Mining activity knowledge from hollywood narratives. In *Proceedings of CIKM 2015*.