PolyU's Experimentation with Guided Summarization

Renxian Zhang, You Ouyang, Wenjie Li
Department of Computing
The Hong Kong Polytechnic University
{csrzhang,csyouyang,cswjli}@comp.polyu.edu.hk}

1 Introduction

In addition to the familiar tasks of update summarization and automatic summary evaluation, the TAC 2010 summarization track includes a new task: guided summarization. It is designed to "encourage a deeper linguistic (semantic) analysis of the source documents" in order to generate a summary guided by a given set of aspects. The new task is an upgrade of query-focused summarization in that an automatic summary is to meet more specific and semantically structured user need. The upgrade demonstrates the growing inclination for semantically oriented summarization, whilst posing new challenges to traditional frameworks and techniques. The update summarization and the automatically evaluating summaries of peers (AESOP) tasks are similar to those included in the TAC 2009 summarization track.

The PolyU team has participated in both tasks of the TAC 2010 summarization track, including the guided summarization task and the AESOP. The guided summarization task is an unfamiliar one, with few success or failure stories. Our team takes up this new challenge, not widely studied in the summarization community, with techniques based on Information Extraction (IE). For the AESOP task, we also experiment with some new techniques. In the following, we report the system design for the tasks as well as findings and diagnoses from our experimentation.

2 Aspect-Guided Summarization

For this part, a 100-word summary is to be generated for a document set of a certain category that is expected to contain as many aspects as possible. The categories and aspects are preset and most of the aspects are specific to categories. Since a word frequency-based summarization scheme does not meet the task requirement (and is actually discouraged by NIST), we experimented with an aspect recognition-based pipeline scheme, consisting of: 1) aspect-bearing sentence recognition; 2) sentential aspect recognition; 3) aspect-based sentence ranking.

2.1 Aspect-bearing Sentence Recognition

We observe from annotated data (Table 1) that aspects are heterogeneous and distributed unevenly in a news article. It might be more difficult to locate individual aspects in a document than to recognize aspect-bearing sentences. Inspired by the architectural design of Patwardhan (2010), we adopt two-stage pipeline aspect recognition. In the first stage, aspect-bearing sentences are recognized, which constitute the candidate pool for summary-worthy sentences. In the second stage, individual aspect instances are recognized in all the aspect-bearing sentences, providing the necessary information for sentence ranking and selection.

The two-stage design has two perceivable advantages: 1) the sentence-level recognition can benefit from an aggregation of all aspect-level annotated data which leads to high recognition accuracy; 2) the pool of aspect-bearing sentences is a bottom-line guarantee for sentence selection because even if the subsequent aspect recognition fails, selecting sentences from this pool ensures that all summary sentences will contain aspects and the summary is still aspect-guided.

For our purpose, aspect recognition is a supervised learning task that requires aspect-annotated data. Because of the lack of such data, we manually constructed a training corpus for the TAC 2010 summarization task by selecting news articles belonging to one of the five target categories ("accidents and natural disasters", "attacks", "health and safety", "endangered resources", and "investigations and trials") from past DUC/TAC test data, all of which come from the AQUAINT and AQUAINT-2 collections to ensure domain and style consistency. For each of the articles of a certain category, we annotated the

target aspects in the format of NIST-provided samples. Table 1 gives the details of our corpus size.

Category	Number of Documents	Number of Aspect Instance
Accidents	160	3355
Attacks	171	4194
Health_safety	160	3639
Resources	160	2944
Investigations	292	5502

Table 1. Training Corpus Size

For each category, we trained an SVM classifier with a linear kernel using all frequent words (*Freq* > 10) as features. We did 10-fold cross validation on our training data and achieved relatively high accuracy with the five classifiers (average accuracy > 80%).

2.2 Sentential Aspect Recognition

We observe from our annotated data that aspect instances are all sentence-bound, spanning from one word to a whole sentence. Therefore the second-stage aspect recognition is treated as essentially an IE task – extracting aspect instances from the aspect-bearing sentences.

To facilitate aspect extraction from annotated aspect instances, we adopt a bottom-up scheme of regular expression pattern induction, which has been successfully applied to similar IE tasks (Califf and Mooney 2003).

The main idea of bottom-up aspect extraction is to induce from training instances as many useful regular expression patterns as possible by doing string-level match and pattern generalization. The main algorithm, *pattern_induce*(*str1*, *str2*), is shown in figure 1.

```
pattern_induce(str1, str2):
  # recursively induce regular expression patterns from str1 and str2
  for each (m1, m2) in max_len_match(str1, str2):
    split str1 into: left_sub1 + m1 + right_sub1
    split str2 into: left_sub2 + m2 + right_sub2
    p_middle = pattern_generalize(m1, m2)
    p_left = pattern_induce(left_sub1, left_sub2)
    p_right = pattern_induce(right_sub1, right_sub2)
  return pattern_merge(p_left, p_middle, p_right)
```

Figure 1. The pattern_induce Algorithm

Pattern_induce( ) recursively induces patterns from the two aspect strings by each time focusing on aligned substrings corresponding to the longest possible match in the current string pair (max_len_match) and the remaining left and right substrings. After patterns are induced from the left, middle, and right substrings, they are merged into one pattern (pattern_merge) as the final result.

Pattern_generalize( ) is the core component of this algorithm, which is based on the regular expression pattern induction in (Califf and Mooney 2003). For two aspect strings, we generalize a three-component regular expression pattern according to their lexical, POS, and semantic attributes. In the following example, Sa and Sb are two matched (sub)strings. The tags "lex", "pos", and "sem" indicate their lexical, POS, and semantic attributes.

Sa: {lex: 'earthquake', pos: NN, sem: <natural phenomenon>}
Sb: {lex: 'floods, pos: NNS, sem: <natural phenomenon>}

For each attribute pair, pattern_generalize( ) generates one or two patterns. If the attributes are an exact match (e.g., Sa.sem and Sb.sem), one pattern is generated (e.g., <natural phenomenon>). If they do not match (e.g., Sa.lex and Sb.lex), two patterns are generated: a disjunction pattern (e.g., 'earthquake' | 'floods') that matches either one string and a wildcard pattern (e.g., *) that matches any string. Then the attribute-level patterns are combined to generate a number of three-attribute patterns, representing all possible combinations. For our example, four such patterns can be generalized:

P1: {lex: 'earthquake' | 'floods', pos: NN | NNS, sem: <natural phenomenon>}
P2: {lex: *, pos: NN | NNS, sem: <natural phenomenon>}

P3: {lex: ‘earthquake’ | ‘floods’, pos: *, sem: <natural phenomenon>}

P4: {lex: *, pos: *, sem: <natural phenomenon>}

The semantic classes are derived from WordNet Domains (Bentivogli et al. 2004). Obviously, each matched pair can generate up to $2^3 = 8$ generalized patterns. If each pair of aspect instances from our training corpus is compared, the resultant pattern collection will be huge. In fact, a fair proportion of such patterns are low-fidelity, i.e., patterns that cannot apply to most of the instances. In order to filter them, we follow (Califf and Mooney 2003) by checking the induced patterns against all the unused instances. All the patterns with recognition precision < threshold (0.75) are deleted.

In our implementation, the pattern generalization process is computationally expensive and it is impractical to generate all the possible patterns based on all aspect instance pairs. Therefore, we only sample a proportion of such pairs.

2.3 Sentence Ranking and Summary Generation

In our design, the focus of the TAC 2010 summarization task is on aspect recognition. After aspect-bearing sentence recognition and sentential aspect recognition are done, sentence ranking and selection are rather straightforward. Since aspects are realized as words or phrases bounded by sentences, they can be reformulated as the “words” in generic summarization. Using this reformulation, we can theoretically apply any popular word frequency-based summarizer (e.g., Carbonell and Goldstein 1998) to aspect-guided summarization. Different from the case of generic summarization, sentences are ranked according to their aspect number and diversity. For aspect a_i and its *j*th instance a_{ij}, we score a_{ij} according to the frequency of a_i (*freq*(a_i)) and the percentage of patterns that recognize a_{ij} (*support*(a_{ij})), thus preferring high-fidelity and rare aspect instances.

$$Score(a_{ij}) = support(a_{ij}) / freq(a_i)$$

The sentence score is the sum of all its aspect instance scores normalized by sentence length.

$$Score(S) = \sum_{a_i \in S} \max(Score(a_i)) / |S|$$

where *Score*(a_i) is the sum of all the *i*th aspect instance scores in *S*. We use max(*Score*(a_i)) because it is possible for a sentence fragment to be recognized as different aspect instances.

In the spirit of MMR (Carbonell and Goldstein 1998), after the highest-ranking sentence is selected to generate the summary, all the *Score*(a_{ij}) are discounted with reference to the similarity between a_{ij} and any same-aspect instances contained in the selected sentence. We iterate the process until the summary word length is reached.

2.4 Update Summarization

According to our design, the update summarization task is rather simple. The two-stage aspect recognition is still central to this task. After that and before ranking and selecting sentences from document set B, we discard any sentence that is highly similar to any sentence in document set A of the same topic. In our implementation, sentence similarity is the cosine similarity between their term vectors and “highly similar” is translated to a value above threshold (0.75).

3 Automatically Evaluating Summaries of Peers (AESOP)

The AESOP tasks include ranking only for the system summaries and ranking for both the human summaries and the system summaries. Generally, most system summaries are composed by sentences extracted from the original documents. By contrast, human summaries are usually abstracts. As reported in previous studies, the evaluation results on extractive and abstractive summaries may also be different. As to the ROUGE evaluation, it is more efficient in ranking the extractive system summaries than differentiating the abstractive human summaries and the extractive system summaries. In our study, we consider a ROUGE-style method as our starting point and then try to improve it by several different strategies.

We submit a total of four runs of evaluations to the AESOP task, which are introduced below. The first one is a baseline run, followed by three extensive runs.

3.1 The Baseline System

In the baseline system, we use a typical matching-based evaluation method, which is indeed similar

to ROUGE. First of all, a simple hypothesis is made that the words appearing in more human summaries are better at differentiating the system summaries. Therefore, the actual importance of a word to the topic is estimated by its document frequency in the human summaries, i.e., *score*(*w*) = *DF*(*w*). Then the total score of a summary *S* is calculated by the sum of the scores of the words in it: $score(S) = \sum_{w_i \in S} DF(w)$. Since the summaries are limited by the fixed length, this score can be directly used to evaluate the content of the summary without length normalization.

3.2 Extending strategy 1: Filtering the non-indicative words

As a matter of fact, the free-style human abstracts may probably contain some words that may be hardly discovered by summarization systems, such as the words that never appear or are very rare in the documents. But system summaries may happen to cover these words. Therefore, we consider a filtering strategy that only uses a subset of the words in the human summaries for evaluation. The words in this subset tend to be discovered by the summarizing systems. We expect that this can more accurately gauge the ability of the summarizing methods in discovering the important content according to the given human summary.

We consider two types of words here, including insignificant words and general words that are identified by their high frequency in the document set and an entropy-based measure respectively. Assume that the frequency of the word *w* in a TAC topic *T* is *TF*(*w*,*T*), the entropy of the word over all the topics in the TAC corpus is calculated as:

$$En(w) = -\sum_{T_i} \frac{TF(w,T_i)}{TF(w)} * log \frac{TF(w,T_i)}{TF(w)} \quad ,$$

where *TF*(*w*) is the total frequency of the word in the corpus that equals $\sum_{T_i} TF(w,T_i)$. This measure indicates how likely a word belongs to a specific topic. Words that have too large entropy are considered as general words and are filtered. In practice, the words that satisfy either *TF*(*w*,*T*) < 4 or *En*(*w*) > 3 are ignored in our evaluation method.

3.3 Extending Strategy 2: Normalizing the word importance

In this run, we try to emphasize the core words in a different way. Usually, there are several core words in a given topic that should be included into the summary, such as the words "American", "Indian", "reservation" in a topic about American Indian reservation. These words may be more significant in differentiating the abilities of the summaries in discovering the important content of the given topic. Here we consider the document frequency in the human summaries as an indicator of the significance level of a word. We propose a normalization strategy that gives more importance to the words that appear in more human summaries. If we denote the set of words whose document frequency in human summaries is *i* as L_i, the score of a word in L_i is then changed from *i* to $1/|L_i|$. In fact, the total score of the words in L_i for each *i* is just 1 according to this strategy. Therefore, it can also be regarded as the evaluation of the summary quality by its ability in covering every set L_i.

3.4 Extending strategy 3: Graph-based ranking model

In the third extension, we consider the effect of the submitted summaries for improving the evaluation method. The basic idea of this strategy is that good summaries usually have more overlap with other summaries. We adopt a PageRank-style ranking model here to calculate the ranking scores of the summaries.

To apply the graph-based ranking model, we need to define the similarity measure between two summaries. For two summaries S_1 and S_2, the similarity $Sim(S_1,S_2)$ is calculated by the overlapping words, i.e., $|S_1 \cap S_2|$. The self-similarity of a summary is set to 0. By calculating the pair-wise similarity between any two summaries, we can obtain the similarity matrix *M*.

Using the similarity matrix, the score of a summary is calculated by the sum of the scores passed from all other summaries, i.e.,

$$score(S_i) = (1-d)\, InitialScore(S_i) + d * \sum_{S_j} score(S_j) * \frac{Sim(S_j,S_i)}{\sum_{S_k} Sim(S_j,S_k)}$$

where $InitialScore(S_i)$ is the prior importance and here is set as 1/4 for reference summaries and 0 for non-reference summaries. *d* is set to 0.85 as in most PageRank-style ranking models.

The ranking process is first setting the initial scores of the summaries and then iteratively calculating the scores using the above formula. The matrix form of the process can be written as

$$V_{Score}^{n+1} = (1-d)V_{InitialScore} + d * M * V_{Score}^{n}$$

Following the power iteration process, the final scores are used as the scores for the summaries in the evaluation scheme.

For the NoModel track, all the four human summaries are used as reference summaries for evaluating the system summaries. For the AllPeers track, the Jackknifing technique is adopted. Each time three human summaries are used as reference summaries for evaluating the other human summary and all the system summaries. Finally, the evaluating results of four evaluating rounds are averaged to get the final scores of all the summaries.

4 TAC 2010 Evaluation and Diagnoses

4.1 Guided Summarization

For this task, the PolyU team submitted two runs. PolyU1 has been described in section 2. PolyU2 is a reordered version of PolyU1. For each summary, we reordered the sentences based on aspect relations (e.g., WHAT, WHEN, WHERE should precede WHY or COUNTERMEASURES) according to some heuristically designed rules. The reordered summaries are expected to be more coherent than the original ones.

To our disappointment, our two runs (37 and 33) perform rather poorly among all the submitted peers. We did follow-up experiments to diagnose problems, which helps us to identify two sources of the poor performance: technical and algorithmic.

Technically, our runs suffered from some bugs, one of which leads to the collapse of five different per-category classifiers to one. The other major problem is with the summary length. For our submitted runs, if a summary exceeds the 100-word length limit, the last sentence is discarded, not truncated so that the whole summary has exactly 100 words. As a result, the average word length of our summaries is about 75-80 words, leading to "unfair" comparison with other 100-word peers, especially in terms of recall measures.

Algorithmically, the aspect-bearing sentence recognition performance is not as good as we expected. We manually inspected the recognized aspect-bearing sentences after debugging and found that the recognition precision is only about 60%, despite the fact that on our training corpus the recognition precision can reach above 80%. This discrepancy is attributed to the difference between our training corpus and the TAC 2010 test set. Second, sentential aspect recognition is incomplete due to the reason described in 2.2, resulting in small subsets of all patterns generalizable from the training data. Last but not least, sentence extraction is IE-driven. Compared with the robust IR-driven techniques widely adopted by the summarization community, our method is more sensitive to the quality of specialized information (aspects in the TAC 2010 task) recognition.

In our post-evaluation study, we debugged the code, resized summaries to exactly 100 words, manually filtered aspect-bearing sentences and experimented with an IR-driven extraction technique. Table 2 lists the result of our experimentation measured by the popular ROUGE metrics.

PolyU1 is our summited run (peer 37), with all the problems described above (buggy, <100 words, low-quality aspect-bearing sentences). Baseline1 and Baseline2 are provided by NIST, and Baseline3 is a simple but robust IR-driven system from SumFocus (Vanderwende et al. 2007), implementing high-frequency word weighting, position weighting, redundancy control, and title word (provided by NIST) overlap. We observe obvious performance gain of our system, from debugging, using higher-quality aspect-bearing sentences, and extending the summary length to 100 words. The resultant PolyU1.h is significantly better than PolyU1. A little unexpectedly, it is still inferior to the simpler Baseline3. We reckon that the PolyU1.* summaries are plagued by their IE-based design. With low-quality aspect-bearing sentence sets (note that manual checking only improves precision, not recall) and lower-quality phrase-level aspect patterns, our system is easily outperformed by a robust IR-based system.

Nevertheless, we cannot rush to the conclusion that IE-based systems are inferior to IR-based systems for the TAC 2010 system because we have not unleashed the full power of IE.

A more valuable finding is that the IE element and the IR element can complement, instead of competing with, each other. In our experimentation, we simply rewarded aspect-bearing sentences with a multiplicative coefficient (1.3) on top of Baseline3 and produced an IR/IE hybrid system (Baseline3 + aspect sentences (100 words)). Evaluation result shows that although combined on such a shallow level, the hybrid system outperforms all the other runs we tried in our experimentation, with ROUGE scores much closer to the best submission (peer 22) than our original submission. Encouraged by this result, we will continue to explore aspect-guided summarization by improving the IE element and designing better IR/IE hybridization.

4.2 AESOP

The performance of our submissions to the AESOP task is shown in Table 3. From the results we have some observations.

(1) Generally, the performance of the proposed systems is among the better-performing half of all the submitted systems.

(2) In the NoModel track, the baseline ROUGE-2 can even perform as well as the best system, which means that ROUGE is indeed a good method in evaluating extractive summarization systems. Our systems have comparable performance but are still worse than ROUGE-2.

(3) In the AllPeers track, our systems are not much different from ROUGE-2, but much worse than the best systems. This may mean that the evaluation methods based on pure word-matching are not capable of evaluating abstractive summaries.

(4) Among all the four submitted systems, the baseline system Run1 is the worst in the NoModel track but the best in the AllPeers track. It may imply that we should adopt simpler evaluation methods when evaluating different types of summary.

	ROUGE-1 recall	ROUGE-2 recall	ROUGE-SU4 recall
Best (peer 22)	0.36832	0.0959	0.12893
Baseline1	0.27784	0.05428	0.08519
Baseline2	0.28686	0.05862	0.0894
Baseline3	0.33752	0.08006	0.11177
PolyU1 (peer 37, <100 words)	0.27988	0.04773	0.08151
PolyU1.d (Debugged, <100 words)	0.28642	0.05353	0.08758
PolyU1.m (manual filtering of aspect sentences, <100 words)	0.31211	0.06631	0.0988
PolyU1.h (manual filtering of aspect sentences, 100 words)	0.33654	0.07042	0.10626
Baseline3 + aspect sentences (100 words)	**0.34603**	**0.08265**	**0.11598**

Table 2. Diagnostic Result

	NoM A	No M B	All P A	All P B
Run1	0.919	0.865	0.911	0.821
Run2	0.917	0.897	0.846	0.717
Run3	0.916	0.906	0.904	0.814
Run4	0.923	0.889	0.848	0.732
Best	0.978	0.964	0.969	0.958
R-2	0.978	0.963	0.895	0.861

Table 3 AESOP Result

5 Conclusion

The PolyU team has experimented with an IE-based technology to tackle the new challenge of aspect-guided summarization. The follow-up study results show that the IE element is certainly helpful and can be integrated with IR-based technology. We also hypothesize that the less salient aspects are in the document, the worse the IR-based baselines will be.

Several new approaches are implemented in the AESOP task and the evaluation results show that more work needs to be done, especially in designing a more effective way to handle abstractive and extractive summaries.

References

Bentivogli, L., Forner, P., Magnini, B., and Pianta, E. 2004. Revising the WordNet domains hierarchy: semantics, coverage and balancing. In *Proceedings of the COLING-2004 Workshop on Multilingual Linguistic Resources*, 364–370, Geneva, Switzerland.

Califf, M., and Mooney, R. 2003. Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction. *Journal of Machine Learning Research* 4:177–210.

Carbonell, J. and Goldstein, J. 1998. *The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries*. In Proceedings of ACM SIGIR 1998, pp 335-336.

Patwardhan, S. 2010. *Widening the field of view of information extraction through sentential event recognition*. PhD Dissertation, The University of Utah.D.

Vanderwende, L., Suzuki, H., Brockett, C., and Nenkova, A. 2007. *Beyond SumBasic: Task-Focused Summarization with Sentence Simplification and Lexical Expansion*. Information Processing and Management 43(6):1606–1618.