Answering Complex Questions Using Query-focused Summarization Technique

Yllias Chali and Shafiq R. Joty
University of Lethbridge
4401 University Drive
Lethbridge, Alberta, Canada, T1K 3M4
{chali,jotys}@cs.uleth.ca

Abstract

Unlike simple questions, complex questions cannot be answered by simply extracting named entities. These questions require inferencing and synthesizing information from multiple documents that can be seen as a kind of topic-oriented, informative multi-document summarization. In this paper, we have experimented with one empirical and two unsupervised statistical machine learning techniques: k-means and Expectation Maximization (EM), for computing relative importance of the sentences. The feature set includes different kinds of features: lexical, lexical semantic, cosine similarity, basic element, tree kernel based syntactic and shallow-semantic. A gradient descent local search technique is used to learn the optimal weights of the features. The effects of the different features are also shown for all the methods of generating summaries.

1 Introduction

After having made substantial headway in factoid and list questions, researchers have turned their attention to more complex information needs that cannot be answered by simply extracting named entities (persons, organizations, locations, dates, etc.) from documents. For example, the question: *"Describe the after-effects of cyclone Sidr-Nov 2007 in Bangladesh"* requires inferencing and synthesizing information from multiple documents. This information synthesis in NLP can be seen as a kind of topic-oriented, informative multi-document summarization, where the goal is to produce a single text as a compressed version of a set of documents with a minimum loss of relevant information.

We experimented with one empirical and two well-known unsupervised statistical machine learning techniques: k-means and EM and evaluated their performance in generating topic-oriented summaries. However, the performance of these approaches depends entirely on the feature set used and the weighting of these features. We extracted different kinds of features (i.e. lexical, lexical semantic, cosine similarity, basic element, tree kernel based syntactic and shallow-semantic) for each of the document sentences in order to measure its importance and relevancy to the user query. We have used a local search technique to learn the weights of the features. In more complex tasks such as computing the relatedness between the query sentences and the document sentences in order to generate query-focused summaries (or answers to complex questions), to our knowledge no study uses tree kernel functions to encode syntactic/semantic information. For all our methods of generating summaries (i.e. empirical, k-means and EM), we have shown the effects of syntactic and shallow-semantic features over the bag-of-words (BOW) features.

2 Feature Extraction

2.1 Lexical Features

2.1.1 N-gram Overlap

N-gram overlap measures the overlapping word sequences between the candidate sentence and the query sentence. With the view to measure the N-gram (N=1,2,3,4) overlap scores, a *query pool* and a *sentence pool* are created. In order to create the query (or sentence) pool, we took the query (or document) sentence and created a set of related sentences by replacing its important words[1] by their first-sense synonyms. We measured the recall based n-gram scores for a sentence P using the following formula:

n-gramScore(P) = $max_i(max_j$ N-gram$(s_i, q_j))$

N-gram(S,Q) = $\frac{\sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{gram_n \in S} Count(gram_n)}$

Where, n stands for the length of the n-gram $(n = 1, 2, 3, 4)$ and $Count_{match}\,(gram_n)$ is the number of n-grams co-occurring in the query and the candidate sentence, q_j is the

[1] hence forth important words are the nouns, verbs, adverbs and adjectives

j^{th} sentence in the query pool and s_i is the i^{th} sentence in the sentence pool of sentence P.

2.1.2 LCS, WLCS and Skip-Bigram Measure

Given two sequences, S_1 *and* S_2, the Longest Common Subsequence (LCS) of S_1 *and* S_2 is a common subsequence with maximum length. The longer the LCS of two sentences is, the more similar the two sentences are. The basic LCS has a problem that it does not differentiate LCSes of different spatial relations within their embedding sequences [4]. To improve the basic LCS method, we can remember the length of consecutive matches encountered so far to a regular two dimensional dynamic program table computing LCS. We call this weighted LCS (WLCS) and use k to indicate the length of the current consecutive matches ending at words x_i and y_j. Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bigram measures the overlap of skip-bigrams between a candidate sentence and a query sentence. We computed the LCS, WLCS, and skip-bigram-based F-measure following [4] using both the query pool and the sentence pool as in the previous section.

2.1.3 Head and Head Related-words Overlap

The sentence (or query) is parsed by Minipar [2] and from the dependency tree we extract the heads which we call exact-head words. Again we take the synonyms, hyponyms and hypernyms[3] of both the query-head words and the sentence-head words and form a set of words which we call head-related words. We measured the exact-head score and the head-related score by counting the overlapping exact-head words and head-related words.

2.2 Lexical Semantic Features

We form a set of words which we call *QueryRelatedWords* by taking the important words from the query, their first-sense synonyms, the nouns' hypernyms/hyponyms and important words from the nouns' gloss definitions. *Synonym overlap (or hypernym/hyponym)* measure is the overlap between the list of synonyms (or hypernyms and hyponyms) of the important words extracted from the candidate sentence and the *QueryRelatedWords*, and *gloss overlap* measure is the overlap between the list of important words that are extracted from the gloss definitions of the nouns of the sentence and the *QueryRelatedWords*.

[2]http://www.cs.ualberta.ca/ lindek/minipar.htm

[3]hypernym and hyponym levels are restricted to 2 and 3 respectively

2.3 Graph-based Similarity Measure

In LexRank [2], the concept of graph-based centrality is used to rank a set of sentences, in producing generic multi-document summaries. A similarity graph is produced for the sentences in the document collection. In the graph, each node represents a sentence. The edges between the nodes measure the cosine similarity between the respective pair of sentences. The degree of a given node is an indication of how much important the sentence is. Once the similarity graph is constructed, the sentences are then ranked according to their eigenvector centrality. To apply LexRank to query-focused context, a topic-sensitive version of LexRank is proposed in [6]. We followed a similar approach in order to calculate this feature.

2.4 Syntactic and Semantic Features:

So far, we have included the features of type bag-of-words. The task like *query-based summarization* that requires the use of more complex syntactic and semantics, the approaches with only BOW are often inadequate to perform fine-level textual analysis. We extracted 3 features that incorporate syntactic/semantic information.

2.4.1 Basic Element (BE) Overlap Measure

The "head-modifier-relation" triples, extracted from the dependency trees are considered as BEs in our experiment. The triples encode some syntactic/semantic information and one can quite easily decide whether any two units match or not- considerably more easily than with longer units [7]. We computed this feature following [7].

2.4.2 Syntactic and Shallow-semantic Features

Encoding syntactic structure is easier and straight forward. Given a sentence (or query), we first parse it into a syntactic tree using a syntactic parser (i.e. Charniak parser) and then we calculate the similarity between the two trees using the *tree kernel* defined in [1].

Initiatives such as PropBank (PB) [3] have made possible the design of accurate automatic Semantic Role Labeling (SRL) systems like ASSERT[4]. For example, consider the PB annotation:

[ARG0 all][TARGET use][ARG1 the french franc][ARG2 as their currency]

Such annotation can be used to design a shallow semantic representation that can be matched against other semantically similar sentences, e.g.

[ARG0 the Vatican][TARGET use][ARG1 the Italian lira][ARG2 as their currency]

[4]http://cemantix.org/assert

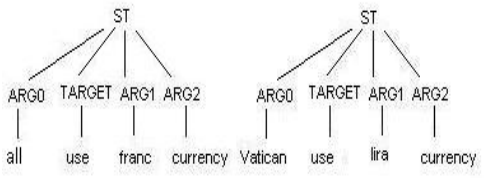

Figure 1. Example of semantic trees

In order to calculate the semantic similarity between the sentences, we first represent the annotated sentence using the tree structures like Figure 1 which we call Semantic Tree (ST) and then we compute the similarity using the Shallow Semantic Tree Kernel (SSTK) as defined in [5].

3 Ranking Sentences

3.1 Learning Feature-weights: A Local Search Strategy

In order to fine-tune the weights of the features, we have used a local search technique. Initially, we set all the feature-weights, $w_1, \cdots, w_n$, as equal values (i.e. 0.5). Based on the current weights we score the sentences and generate summaries accordingly. We evaluate the summaries using the automatic evaluation tool ROUGE [4] (described in Section 4) and the ROUGE value works as the feedback to our learning loop. Our learning system tries to maximize the ROUGE score in every step by changing the weights individually by a specific step size (i.e. 0.01). Once we have learned the feature-weights, our *empirical* method computes the final scores for the sentences using the formula:

$$score_i = \vec{x_i}.\vec{w} \tag{1}$$

Where, $\vec{x_i}$ is the feature vector for i-th sentence, $\vec{w}$ is the weight vector and $score_i$ is the score of i-th sentence.

3.2 K-means Learning

We start with a set of initial cluster centers and go through several iterations of assigning each object to the cluster whose center is closest. After all objects have been assigned, we recompute the center of each cluster as the centroid or mean (μ) of its members.

Once we have learned the means of the clusters using the k-means algorithm, our next task is to rank the sentences according to a probability model. We have used Bayesian model in order to do so. Bayes' law says:

$$P(q_k|\vec{x}, \Theta) = \frac{p(\vec{x}|q_k, \Theta)P(q_k|\Theta)}{\sum_{k=1}^{K} p(\vec{x}|q_k, \Theta)p(q_k|\Theta)} \tag{2}$$

where q_k is a class, $\vec{x}$ is a feature vector representing a sentence and Θ is the parameter set of all class models. We set the weights of the clusters as equiprobable (i.e. $P(q_k|\Theta) = 1/K$).

3.3 EM Learning

EM is an iterative two step procedure: 1. Expectation-step and 2. Maximization-step. In the expectation step, we compute expected values for the hidden variables $h_{i,j}$ which are cluster membership probabilities. Given the current parameters, we compute how likely an object belongs to any of the clusters. The maximization step computes the most likely parameters of the model given the cluster membership probabilities.

Once the sentences are clustered by EM algorithm, we filter out the sentences which are not query-relevant by checking their probabilities, $P(q_r|x_i, \Theta)$ where, q_r denotes the cluster "query-relevant".

Our next task is to rank the query-relevant sentences in order to include them in the summary. This can be done easily by multiplying the feature vector $\vec{x_i}$ with the weight vector $\vec{w}$ that we learned by the local search technique (eq:1).

4 Experimental Evaluation

4.1 Evaluation Setup

We used the main task of DUC-2007 for evaluation. Given a complex question and a collection of relevant documents, the task was to synthesize a fluent, well-organized 250-word summary of the documents that answers the question(s) in the topic. We carried out automatic evaluation of our summaries using ROUGE [4] toolkit, which has been widely adopted by DUC for automatic summarization evaluation. It measures summary quality by counting overlapping units such as the n-grams (ROUGE-N), word sequences (ROUGE-L and ROUGE-W) and word pairs (ROUGE-S and ROUGE-SU) between the candidate summary and the reference summary. One purpose of our experiments is to study the impact of different features for complex question answering task. To accomplish this, we generated summaries for the topics of DUC 2007 by each of our seven systems defined as below:

The **LEX** system generates summaries based on only lexical features. The **LSEM** system considers only lexical semantic features. The **COS** system generates summary based on the graph-based method. The **SYS1** system considers all the features except the BE, syntactic and semantic

features. The **SYS2** system considers all the features except the syntactic and semantic features. The **SYS3** considers all the features except the semantic and the **ALL** system generates summaries taking all the features into account.

4.2 Evaluation Results

Table 1 shows the ROUGE (F-measure) measures for k-means. It shows, SYS2 gets 9%, SYS3 and ALL gets 15% improvement in *ROUGE-2* scores over the SYS1 system. We get best *ROUGE-W* scores for SYS2 (i.e. including BE) but SYS3 and ALL do not perform well in this case. SYS2 improves the ROUGE-W score by 1% over SYS1. We do not get any improvement in *ROUGE-S* scores when we include any kind of syntactic/semantic structures.

The case is different for EM and empirical approaches. Here, in every case we get a significant amount of improvement when we include the syntactic and/or semantic features. For EM (Table 2), the ratio of improvement over SYS1 is: 1-3% for SYS2, 3-15% for SYS3 and 2-24% for ALL. In our empirical approach (Table 3), SYS2, SYS3 and ALL improve the scores by 3-11%, 7-15% and 8-19% over SYS1 respectively.

Score	LEX	LSEM	COS	SYS1	SYS2	SYS3	ALL
RG-2	0.078	0.080	0.089	0.078	0.085	0.090	0.090
RG-W	0.130	0.129	0.134	0.140	0.141	0.139	0.138
RG-S	0.142	0.139	0.150	0.153	0.151	0.152	0.152

Table 1. ROUGE measures in k-means learning

Score	LEX	LSEM	COS	SYS1	SYS2	SYS3	ALL
RG-2	0.092	0.083	0.090	0.088	0.090	0.101	0.109
RG-W	0.137	0.128	0.134	0.136	0.138	0.139	0.139
RG-S	0.157	0.140	0.149	0.154	0.159	0.163	0.162

Table 2. ROUGE measures in EM learning

Score	LEX	LSEM	COS	SYS1	SYS2	SYS3	ALL
RG-2	0.089	0.083	0.090	0.090	0.100	0.104	0.107
RG-W	0.135	0.128	0.134	0.137	0.143	0.147	0.148
RG-S	0.155	0.140	0.150	0.157	0.162	0.169	0.170

Table 3. ROUGE measures in empirical approach

Table 4 shows the F-scores of the ROUGE measures for one baseline system, and our three scoring techniques considering all features. The baseline system generates summaries by returning all the leading sentences in the $\langle TEXT \rangle$ field of the most recent document(s). It shows that the empirical approach outperforms the other two learning techniques and EM performs better than k-means algorithm. EM improves the scores over k-means by 0.7-22.5%. Empirical approach improves the F-scores over k-means and EM by 5.9-20.2% and 3.5-6.5% respectively.

System	ROUGE-1	ROUGE-2	ROUGE-W	ROUGE-S
Baseline	0.335	0.065	0.114	0.113
k-means	0.390	0.890	0.138	0.152
EM	0.399	0.109	0.139	0.162
Empirical	0.413	0.107	0.148	0.170

Table 4. F-measures for different systems

5 Conclusion

Our experiments show the following: (a) our approaches achieve promising results, (b) empirical approach outperforms the other two learning and EM performs better than the k-means algorithm for this particular task, and (c) our systems achieve better results when we include BE, syntactic and semantic features.

References

[1] M. Collins and N. Duffy. Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver, Canada, 2001.

[2] G. Erkan and D. R. Radev. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[3] P. Kingsbury and M. Palmer. From Treebank to PropBank. In *Proceedings of the international conference on Language Resources and Evaluation*, Las Palmas, Spain, 2002.

[4] C. Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, Barcelona, Spain, 2004.

[5] A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classificaion. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 776–783, Prague, Czech Republic, 2007. ACL.

[6] J. Otterbacher, G. Erkan, and D. R. Radev. Using Random Walks for Question-focused Sentence Retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 915–922, Vancouver, Canada, 2005.

[7] L. Zhou, C. Y. Lin, and E. Hovy. A BE-based Multi-dccument Summarizer with Query Interpretation. In *Proceedings of Document Understanding Conference*, Vancouver, B.C., Canada, 2005.