Improving the Performance of the Reinforcement Learning Model for Answering Complex Questions

Yllias Chali
University of Lethbridge
4401 University Drive W,
Lethbridge, AB, Canada
chali@cs.uleth.ca

Sadid A. Hasan
University of Lethbridge
4401 University Drive W,
Lethbridge, AB, Canada
hasan@cs.uleth.ca

Kaisar Imam
University of Lethbridge
4401 University Drive W,
Lethbridge, AB, Canada
imam@uleth.ca

ABSTRACT

This paper addresses the task of answering complex questions using a multi-document summarization approach within a reinforcement learning setting. Given a set of complex questions, a list of relevant documents per question, and the corresponding human-generated summaries (i.e. answers to the questions) as training data, the reinforcement learning module iteratively learns a number of feature weights in order to facilitate the automatic generation of summaries i.e. answers to unseen complex questions. Previous works on this task have utilized a fully automatic reinforcement learning framework that selects the document sentences as the potential candidate (i.e. machine-generated) summary sentences by exploiting a relatedness measure with the available human-written summaries. In this paper, we propose an extension to this model that incorporates user interaction into the reinforcement learner to guide the candidate summary sentence selection process. Experimental results reveal the effectiveness of the user interaction component in the reinforcement learning framework.

Categories and Subject Descriptors

H.3.m [**Information Storage and Retrieval**]: Miscellaneous; I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms

Algorithms, Performance, Experimentation

Keywords

Complex question answering, multi-document summarization, reinforcement learning, user interaction

1. INTRODUCTION

Users often ask questions in the context of a wider information need, for instance when researching a specific topic. This poses the problem of complex Question Answering (QA) that often relates to multiple entities, events and complex relations between them. For example, a complex question like *"How was Haiti affected by the earthquake?"* often has a wider focus without a single or well-defined information need. Multi-document summarization techniques can be applied effectively to handle this type of questions [4]. In this paper, we consider the task of answering complex questions using an extractive multi-document summarization approach within a reinforcement learning setting. The major limitation of the current search engines is that they lack the way of measuring whether a user is satisfied with the information provided. They also cannot improve their policy dynamically in real time [15]. This is the main motivation of applying the reinforcement approach [12] to the complex question answering domain. Given a set of complex questions, a collection of relevant documents per question, and the corresponding human-generated summaries (i.e. answers to the questions), a reinforcement learning model can be trained to extract the most important sentences as system generated automatic summaries.

Previous works on this domain have used a reinforcement learning framework that verified the importance of an original document sentence by measuring its similarity with the abstract summary sentences using a reward function [1]. Their formulation was simplified with no user interaction as they took the assumption that the human-generated abstract summaries are the gold-standards and the users are satisfied with these summaries. Experiments in the complex interactive Question Answering (ciQA) task[1] at TREC-2007 demonstrate the significance of user interaction in this domain. Motivated by the effect of user interaction shown in the previous studies [7, 14, 13], in this paper, we propose an extension to this reinforcement learning model by incorporating user interaction into the learner and argue that the user interaction component can provide a positive impact in the candidate summary sentence selection process.

2. REINFORCEMENT LEARNING MODEL

In [1], the complex question answering problem is formulated by estimating an action-value function. Given a complex question q and a collection of relevant documents $D = \{d_1, d_2, d_3, \ldots, d_n\}$, the task is to extract a summary (i.e. answer) automatically. The state is defined by the current status of the answer space. In each iteration, one sentence is added from the document to the answer pool that

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

[1] http://www.umiacs.umd.edu/j̃immylin/ciqa/

in turn changes the state. In each state, there is a possible set of actions where "action" stands for *selecting/choosing a particular sentence* from the remaining document sentences that are not included so far in the candidate extract summary. The value of taking an action a in the state s is defined under a policy π , denoted as $Q^{\pi}(s,a)$:

$$\begin{aligned} Q^{\pi}(s,a) &= E_{\pi}\left\{R_t|s_t = s, a_t = a\right\} \\ &= E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma_k r_{t+k+1}|s_t = s, a_t = a\right\} \end{aligned} \quad (1)$$

Here, E_{π} denotes the expected value given that the agent follows policy π, R_t is the *expected return* that is defined as a function of the reward sequence, $r_{t+1}, r_{t+2}, \cdots$, where r_t is the numerical reward that the agent receives at time step, t. We call Q^{π} the action-value function for policy π. γ is the discount factor that determines the importance of future rewards. Once the optimal policy (π^*) is found, the agent chooses the actions using the Maximum Expected Utility Principle [9]. As the number of states and actions are infinite, the approximate action-value function is represented as a parameterized functional form with parameter vector, $\vec{\theta_t}$. Corresponding to every state-action pair (s,a) , there is a column vector of features, $\vec{\varphi_s} = (\varphi_s(1), \varphi_s(2), \ldots, \varphi_s(n))^T$ with the same number of components as $\vec{\theta_t}$. The approximate action-value function is given by:

$Q_t(s,a) = \vec{\theta}_t^T\,\vec{\varphi_s} = \sum_{i=1}^{n}\theta_t(i)\varphi_s(i)$

In the training step of this reinforcement learning model, for computing the rewards, a fully automatic approach is used to select the document sentences as the potential candidate (i.e. machine-generated) summary sentences by exploiting a relatedness measure with the available human-written summaries using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [6]. A modified linear, gradient-descent version of Watkins' $Q(\lambda)$ algorithm [12] is applied to estimate the parameters of the model [1]. In this paper, we propose an extension to this reinforcement learning model by incorporating user interaction into the learner that can improve the performance of the reinforcement learning model by enhancing the efficiency of the candidate summary sentence selection process.

3. USER INTERACTION MODELING

In our proposed model, for a certain number of iterations during the training stage of the reinforcement learning, the user is presented with the top five candidate sentences (based on the ROUGE similarity scores between the candidate sentences and the human summaries). The user can also see the complex question being considered and the current status (content) of the answer space (i.e. state). The task of the user at this point is to select the best candidate among the five to be added to the answer space. In the reinforcement learning model of [1], the first candidate was selected to be added automatically as it was having the highest similarity score. In this way, there was a chance that a potentially unimportant sentence could be chosen that is not of user's interest. However, in our extended reinforcement learning model, the user interaction component enables us to incorporate the human viewpoint and thus, the judgment for the best candidate sentence is supposed to be perfect. The outcome of the reinforcement learner is a set of weights that are updated through several iterations until the algorithm converges. The user selects a sentence to add to the answer space and the feature weights are updated based on this response. The similar process runs up to three iterations for each topic during training. In the rest number of the iterations, the algorithm selects the sentences automatically and continue updating the weights accordingly.

4. FEATURE SPACE

Each sentence of a document is represented as a vector of feature-values. Our feature set includes two types of features, where one declares the importance of a sentence in a document and the other measures the similarity between each sentence and the user query. To measure the importance of a sentence, we consider its position, length, and match with title, certain named entity and cue words. For query-related features, we consider n–gram overlap, LCS, WLCS, skip-bigram, exact-word, synonym, hypernym/hyponym, gloss and Basic Element (BE) overlap, and syntactic features. These features have been adopted from several related works in the problem domain [3, 8, 10].

5. EXPERIMENTS AND RESULTS

5.1 Task Overview

This paper deals with the complex question answering task defined in DUC[2]-2006. The task is as follows: *"Given a complex question (topic description) and a collection of relevant documents, the task is to synthesize a fluent, well-organized 250-word summary of the documents that answers the question(s) in the topic".* We use an interactive reinforcement learning approach to generate topic-oriented 250-word extract summaries. Each topic and its document cluster was given to 4 different NIST[3] assessors, including the developer of the topic. Each assessor created a 250-word summary of the document cluster that satisfies the information need expressed in the topic. In the reinforcement learning phase, these multiple reference summaries (also termed as human-generated abstracts) are compared with the original document sentences using ROUGE to rank the candidate document sentences in terms of similarity scores. For our experiments, we use the first 30 topics at most from the DUC-2006 data to learn the weights respective to each feature and then use these weights to produce extract summaries for the next 15 topics (test data).

5.2 System Description

The major objective of this research is to study the impact of the user interaction component in the reinforcement learning framework. To accomplish this purpose, we follow six different ways of learning the feature weights by varying the amount of user interaction incorporated and the size of the training data: **1) SYS_0_20**, **2) SYS_10_20**, **3) SYS_20_0**, **4) SYS_20_10**, **5) SYS_30_0**, and **6) SYS_30_30**. The numbers in the system titles indicate how many user-interaction and non-user-interaction topics each system included during training, respectively. For example, the first system is trained with 20 topics of the DUC-2006 data without user interaction. Among these systems, the sixth system is different as it is trained with the first 30 topics of the DUC-2006 data without user interaction. The learned

[2] http://duc.nist.gov/

[3] http://www.nist.gov/index.html

weights that are found from the **SYS_30_0** experiment are used as the initial weights of this system. This means that virtually the **SYS_30_30** system is trained with 60 topics (30 topics with interaction from **SYS_30_0** and 30 topics without interaction). The outcomes of all these systems are the sets of learned feature weights that are used to generate extract summaries (i.e. answers) for the last 15 topics (test data) of the DUC-2006 data set. So, after the six learning experiments, we get six sets of learned feature weights which are used to generate six different sets of summaries for the test data (15 topics). We evaluate these six versions of summaries for the same topics and analyze the effect of user interaction in the reinforcement learning framework.

5.3 Results and Discussion

We evaluate the system generated summaries using the automatic evaluation toolkit ROUGE [6]. We report the two official ROUGE metrics of DUC-2006 in the results: ROUGE-2 (bigram) and ROUGE-SU (skip bigram). In Table 1, we compare the ROUGE-F scores of all the systems. In our experiments, the only two systems that were trained with 20 topics are **SYS_0_20** and **SYS_20_0** (the one has 20 unsupervised, the other has 20 supervised). From the results, we see that there is essentially no difference between their performance in terms of ROUGE-2 scores. However, the **SYS_20_0** system improves the ROUGE-SU scores over the **SYS_0_20** system by 0.47%. Again, we see that the **SYS_20_10** system improves the ROUGE-2 and ROUGE-SU scores over the **SYS_10_20** system (both systems had 30 topics but where **SYS_20_10** had more human-supervised topics) by 0.96%, and 8.56%, respectively. We also find that the **SYS_30_0** system improves the ROUGE-2 and ROUGE-SU scores over the **SYS_20_10** system (both systems had 30 topics with **SYS_30_0** having more human supervision) by 0.25%, and 0.80%, respectively. So, the results show a clear trend of improvement when human interaction is incorporated. We can also see that the **SYS_30_30** system is performing the best since it starts learning from the learned weights that are generated from the outcome of the **SYS_30_0** setting. This denotes that the user interaction component has a positive impact on the reinforcement learning framework that further controls the automatic learning process efficiently (after a certain amount of interaction has been incorporated). In table 2 and table 3, we report the 95% confidence intervals for ROUGE-2 and ROUGE-SU to show the significance of our results.

Systems	ROUGE-2	ROUGE-SU
SYS_0_20	0.052252	0.118643
SYS_10_20	0.059835	0.114611
SYS_20_0	0.052551	0.119201
SYS_20_10	0.060409	0.124420
SYS_30_0	0.060560	0.125417
SYS_30_30	0.060599	0.125729

Table 1: Performance comparison: F-Scores

Systems	ROUGE-2
SYS_0_20	0.040795 - 0.063238
SYS_10_20	0.046216 - 0.073412
SYS_20_0	0.041366 - 0.063316
SYS_20_10	0.046718 - 0.073785
SYS_30_0	0.046364 - 0.074779
SYS_30_30	0.050021 - 0.075493

Table 2: 95% confidence intervals: ROUGE-2

Systems	ROUGE-SU
SYS_0_20	0.110603 - 0.126898
SYS_10_20	0.114425 - 0.134460
SYS_20_0	0.111324 - 0.127472
SYS_20_10	0.114423 - 0.134463
SYS_30_0	0.114820 - 0.134460
SYS_30_30	0.117726 - 0.134321

Table 3: 95% confidence intervals: ROUGE-SU

The automatic evaluation using ROUGE is not always reliable to all researchers [11]. So, we conduct an extensive manual evaluation of our systems. Two native English-speaking university graduate students judge the summaries for linguistic quality and overall responsiveness according to the DUC-2007 evaluation guidelines. They were blind to which system each output came from. The given linguistic quality score is an integer between 1 (very poor) and 5 (very good) and is guided by consideration of the following factors: 1. Grammaticality, 2. Non-redundancy, 3. Referential clarity, 4. Focus, and 5. Structure and Coherence. The responsiveness score is also an integer between 1 (very poor) and 5 (very good) and is based on the amount of information in the summary that helps to satisfy the information need. The inter-annotator agreement of Cohen's $\kappa = 0.55$ [2] was computed that denotes a moderate degree of agreement [5] between the raters. Table 4 presents the average linguistic quality and overall responsive scores of all our systems. Analyzing these results, we can clearly see the positive impact of the user interaction component in the reinforcement learning framework. The improvements in the results are statistically significant[4] ($p < 0.05$).

Systems	Linguistic Quality	Overall Responsiveness
SYS_0_20	2.92	3.20
SYS_10_20	3.45	3.40
SYS_20_0	3.12	3.39
SYS_20_10	3.50	3.72
SYS_30_0	3.68	3.84
SYS_30_30	3.96	4.10

Table 4: Linguistic quality and responsiveness scores

5.4 Effect of User Interaction

The main goal of the reinforcement learning phase is to learn the appropriate feature weights (See Section 4) that can be used in the testing phase. The effect of user feedback on the feature weights can be shown using a graph. We present the weights from different stages of the **SYS_20_0** experiment in figure 1. Note that the **SYS_20_0** system is trained with 20 topics of the DUC-2006 data with user interaction. The labels in the Y-axis refers to the features in the following order: 1) 1-gram overlap, 2) 2-gram overlap, 3) LCS, 4) WLCS, 5) exact word overlap, 6) synonym overlap, 7) hypernym/hyponym overlap, 8) sentence length, 9) title match, 10) named entity match, 11) cue word match,

[4] We tested statistical significance using Student's t-test.

12) syntactic feature, and 13) BE overlap. Analyzing the graph, we find that at the end of the learning phase (end of topic-20), all the feature weights converge to zero except for 2-gram overlap and BE overlap. The zero weight values suggest that the associated features can be eliminated because they do not contribute any relevant information to *action* (i.e. candidate sentence) selection. The graph verifies that the proposed reinforcement system is responsive to user interests and actions.

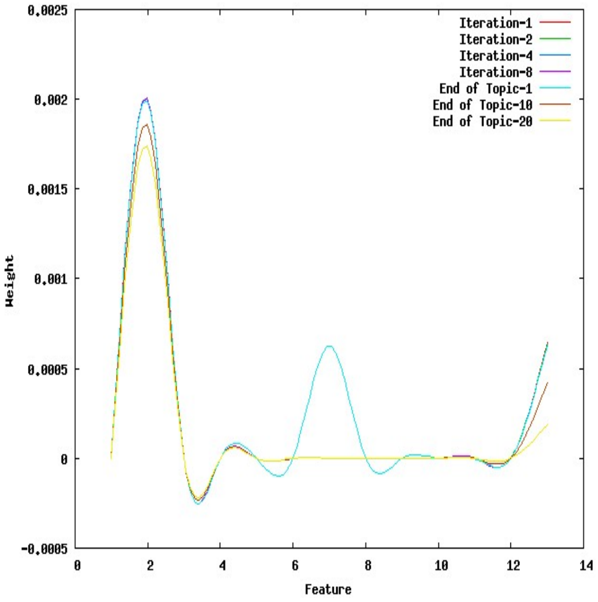

Figure 1: Effect of user feedback on feature weights

6. CONCLUSION

We proposed an extension to the reinforcement learning model of answering complex questions by incorporating a user interaction component. Experiments revealed that the systems trained with user interaction perform considerably better and this trend continues with the increase of the training data even using no interaction. This suggests that the system is capable to learn automatically (i.e. without interaction) and effectively after a sufficient amount of user interaction is provided as the guide to candidate answer sentence selection. A thorough automatic and manual evaluation of our systems proved this claim.

Acknowledgments

The research reported in this paper was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada–discovery grant and the University of Lethbridge.

7. REFERENCES

[1] Y. Chali, S. A. Hasan, and K. Imam. A Reinforcement Learning Framework for Answering Complex Questions. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 307–310. ACM, 2011.

[2] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[3] H. P. Edmundson. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery (ACM)*, 16(2):264–285, 1969.

[4] S. Harabagiu, F. Lacatusu, and A. Hickl. Answering Complex Questions with Random Walk Models. In *Proceedings of the 29th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 220 – 227, 2006.

[5] J. R. Landis and G. G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, 1977.

[6] C. 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.

[7] J. Lin, N. Madnani, and B. J. Dorr. Putting the user in the loop: interactive maximal marginal relevance for query-focused summarization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 305–308. ACL, 2010.

[8] M. Litvak, M. Last, and M. Friedman. A New Approach to Improving Multilingual Summarization using a Genetic Algorithm. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 927–936. ACL, 2010.

[9] S. Russel and P. Norvig. *Artificial Intelligence A Modern Approach, 2nd Edition.* Prentice Hall, 2003.

[10] F. Schilder and R. Kondadadi. FastSum: Fast and Accurate Query-based Multi-document Summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 205–208. ACL, 2008.

[11] J. Sjöbergh. Older Versions of the ROUGEeval Summarization Evaluation System Were Easier to Fool. *Information Processing and Management*, 43:1500–1505, 2007.

[12] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction* . The MIT Press, Cambridge, Massachusetts, London, England, 1998.

[13] M. Wu, F. Scholer, and A. Turpin. User preference choices for complex question answering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 717–718. ACM, 2008.

[14] R. Yan, J. Nie, and X. Li. Summarize what you are interested in: An optimization framework for interactive personalized summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1342–1351, Edinburgh, Scotland, UK., 2011. ACL.

[15] H. Zaragoza, B. B. Cambazoglu, and R. Baeza-Yates. Web search solved?: all result rankings the same? In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 529–538. ACM, 2010.