Temporal Question Generation from History Text

Harsimran Bedi
TCS Research, India

Sangameshwar Patil
TCS Research, India

Girish K. Palshikar
TCS Research, India

{bedi.harsimran, sangameshwar.patil, gk.palshikar}
@tcs.com

Abstract

Temporal analysis of history text has always held special significance to students, historians and the Social Sciences community in general. We observe from experimental data that existing deep learning (DL) models of ProphetNet and UniLM for question generation (QG) task do not perform satisfactorily when used directly for temporal QG from history text. We propose linguistically motivated templates for generating temporal questions that probe different aspects of history text and show that finetuning the DL models using the temporal questions significantly improves their performance on temporal QG task. Using automated metrics as well as human expert evaluation, we show that performance of the DL models finetuned with the template-based questions is better than finetuning done with temporal questions from SQuAD.

1 Introduction

Major events in history have always held significance for the Social Sciences community. Understanding the history of a nation, a society, an era or historic personalities involves analysing the timelines of major events that happened, their locations, the actors, and the consequences that followed. Given a set of history documents (Wikipedia pages, books, papers), it is a challenging problem to automatically extract timelines from them and to use these timelines for downstream applications such as Q&A (Bauer and Teufel, 2016; Bedi et al., 2017; Palshikar et al., 2019a,b; Gottschalk and Demidova, 2019; Hingmire et al., 2020). Another important application is to generate temporal questions from historical narrative text, which can be used for testing and improving the students' understanding of the temporal aspects of history. While much research has focused on generation of general questions from text, generation of temporal questions has received less attention (Heilman and Smith, 2010; Du et al., 2017; Pan et al., 2020; Peng et al., 2020).

We experimented with two deep learning (DL) based language models, UniLM (Dong et al., 2019) and ProphetNet (Qi et al., 2020) finetuned on SQuAD (Rajpurkar et al., 2016) for QG from history documents. The percentage of temporal questions generated and the acceptability of the questions was quite low (details in Section 4). To improve the quality and quantity of temporal questions generated, we propose linguistic knowledge based methods (*templates*). Each template analyzes the given sentence to generate a temporal question having a specific structure, by looking at the relationships among nominal and verbal events, time expressions (timex), verbs and its arguments in the dependency parse tree. Our manual evaluations show that the templates generate temporal questions with high acceptability. Then we leveraged the generated temporal questions to finetune ProphetNet and UniLM for the temporal question generation task. We evaluated the performance of the finetuned models using both domain-expert evaluation and automated metrics of BLEU-4, METEOR, and ROUGE-L. The results show that temporal questions created through our templates significantly improve the performance of DL models on the task of temporal QG.

2 Related Work

Compared to general purpose question generation (QG) and question answering (QA), the temporal aspect of QG has been relatively less explored in the literature. TEQUILA (Jia et al., 2018) is a system for temporal QA over knowledge bases (KB). It identifies temporal questions, converts them into non-temporal sub-questions and temporal constraints and then uses the underlying KB-QA

Proceedings of the 18th International Conference on Natural Language Processing, pages 408–413
Silchar, India. December 16 - 19, 2021.

engine to extract the answers to the sub-questions. Sun et al. (2018) generate and rank answers to complex questions by creating Event Graphs from text using dependency parser mainly focusing on temporal and causal relations. Both of these methods focus on temporal QA whereas our work is focused on temporal QG.

Recent trends in deep learning (DL) based methods for QG are mainly driven by neural sequence-to-sequence modeling (Qi et al., 2020; Dong et al., 2019). However, acceptability of temporal questions generated using these methods is low. Compared to other literature, the work by Peng et al. (2020) is closer to the scope of this paper. They have used triples <subject, predicate, object> from WikiData, a structured knowledge-base and a rule-based method to generate temporal questions. However, our approach uses raw input text and does not need any external KB for generating temporal questions. Further, we use the template-based questions to finetune and improve the DL methods for temporal QG.

3 Our Approach

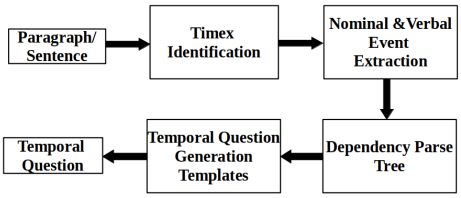

Figure 1: Temporal QG pipeline

Our method for temporal QG has two stages. In the first stage, we process the input text to extract and analyze the temporal as well as linguistic information. This information is then transformed syntactically as well as semantically using carefully designed templates to generate temporal questions. In the second stage, the generated temporal questions are used to finetune DL based models, ProphetNet and UniLM to improve their ability to generate acceptable temporal questions.

Extraction of TIMEX and Events: As shown in Figure 1, the processing pipeline of stage 1 starts with identification of tokens that indicate temporal expressions (i.e., TIMEX). We use HeidelTime (Strötgen and Gertz, 2015) to identify and normalize timex entities in the input sentence. To capture the verbal events in the input text, we make use of the past-tense propagation technique proposed by Palshikar et al. (2019a). To identify the nominal events, we make use the approach proposed by (Ramrakhiyani et al., 2021). They make use of NomBank (Meyers et al., 2004) and deverbal nouns (Gurevich et al., 2008) to identify the nominal events. Since, we specifically focus on history text in this paper, we also use a curated gazette of headwords indicating verbal and nominal events in history domain to augment the event extraction process.

Algorithm 1: Algorithm for template: ``When did SUB V N?''

Input: S = Sentence;
PT = Parse Tree of S;
TE_S = Timex Entities in S;
NE_S = Nominal Events in S;
Output: Temporal question of the form: ``When did SUB V N?''

Let w be a verb in past tense in S which is not modal OR auxiliary verb;
if *there is no such w* **then** return;
Let V be the present tense form of w;
Let SUB be the complete text connected to w using DR "nsubj" in PT;
if *there is no such SUB or SUB contains a pronoun* **then** return;
Let P1 be a preposition in S such that P1 is connected to w using DR "prep" in PT AND T is a timex in TE_S connected to P1 using DR "pobj" in PT;
if *there is no such T* **then** return;
if *there is N in NE_S such that N is connected to a preposition P2 using DR "pobj" in PT AND P2 is connected to w using DR "prep" in PT* **then**
  print ``When did SUB V N?''
else if *there is N in NE_S such that N is connected to w using DR "dobj" in PT* **then**
  print ``When did SUB V N?''
else if *there is verb U in present tense in S connected to w using DR "xcomp" in PT AND there is N in NE_S such that N is connected to U using DR "dobj" in PT* **then**
  print ``When did SUB U N?''

Temporal Question Generation Templates: The templates are designed to probe different aspects of history text such as spatio-temporal details of an event, key players involved in it, relative temporal order among events, consequences of an event etc. We note that the set of templates is open to further extension based on the interest of historians and analysts. The proposed approach is flexible such that the questions generated by the DL based methods can be adapted to the additional templates.

Table 1 provides an overview of the templates along with examples of generated temporal questions. Due to the space constraints and ease of exposition, we focus on the template #2 `When did <Subject> <Verb> <NominalEvent>?` ; but the overall approach is similar in case of other tem-

Sr	Template	Sentence	Question
1	When did <N> happen?	During the Jassy-Kishinev Offensive of August 1944, Romania switched sides on August 23, 1944.	When did the Jassy-Kishinev Offensive happen?
2	When did <SUB> <V> <N>?	In June 1941, Hitler ordered an invasion of the Soviet Union.	When did Hitler order an invasion of the Soviet Union?
3	What happened to <SUB> after <PR> <V> <N> <P> <T>?	Gandhi launched the Quit India Movement in August 1942, after which he was arrested with other Congress lieutenants like Nehru and Patel.	What happened to Gandhi after he launched the Quit India Movement in August 1942?
4	What happened to <SUB> during <T>?	During the 1980s, Cromwell's statue was relocated outside Wythenshawe Hall, which had been occupied by Cromwell's troops.	What happened to Cromwell's statue during the 1980s?
5	Which event happened first: <N1> or <N2>?	Russia was promised Constantinople in the Constantinople Agreement of 1915. The Jews were promised a homeland in Palestine in the Balfour Declaration of 1917, but the Arabs had already been promised a sovereign state in Turkish-controlled regions.	Which event happened first: the Constantinople Agreement or the Balfour Declaration?
6	What happened to <SUB> <TM> <N> <P> <T>?	India's Prime Minister, Shastri, suffered a fatal heart attack soon after the Tashkent Agreement on January 11, 1966.	What happened to India's Prime Minister after the Tashkent Agreement on January 11, 1966?
7	When did <SUB> <VE> <O>?	By the end of 1941, German forces and the European Axis powers occupied most of Europe and North Africa.	When did the European Axis powers occupy most of Europe and North Africa?

Table 1: Overview of templates; SUB=subject, N=Nominal Event, V=Verb, P=preposition, T=Timex, N1=Nominal Event 1, N2=Nominal Event 2, TM=Temporal marker, VE=Verbal Event, O=Object

plates. We generate the dependency parse tree of the sentence using spacy (Honnibal et al., 2020) and apply the template patterns to generate temporal questions. We traverse through the part-of-speech (POS) tags and dependency relations (DR) in the parse tree of the sentence to extract phrases and tokens required to fill the relevant parameters of the templates. For instance, for template #2, we need the `Subject (SUB), Verb (V), Nominal Event (N)` with the constraint that a temporal expression (T) is appropriately associated. Algorithm 1 gives the details of how we verify the constraints and extract the parameters of the template #2. As an example, consider the sentence: `In June 1941, Hitler ordered an invasion of the Soviet Union.` Figure 2 shows its dependency parse tree and the POS-tags of tokens. From this sentence, the Algorithm 1 extracts T = `June 1941`, P = `In`, SUB = `Hitler`, V = `ordered`, N = `invasion of the Soviet Union`. Finally, after verifying the appropriate constraints, Algorithm 1 generates the question `When did Hitler order an invasion of the Soviet Union?`.

Fine-tuning DL Models for Temporal QG: The second stage of our approach overcomes the limitations of existing deep learning (DL) models for temporal QG. We use two different DL models to emphasize the flexibility and robustness of this approach. ProphetNet uses future n-gram prediction and n-stream self attention mechanism to achieve state-of-the-art performance on many NLP tasks. Unified pre-trained Language Model (UniLM) employs a shared Transformer network and specific self-attention masks. Both ProphetNet and UniLM have shown superior performance on general purpose QG task for SQuAD dataset (Qi et al., 2020; Dong et al., 2019). We observe that off-the-shelf QG models of ProphetNet and UniLM do not perform satisfactorily when used directly for temporal QG from history text (Table 4). Hence, we finetune both the DL models using the temporal questions generated by the templates in the first stage. The DL models tackle the temporal QG task as a sequence-to-sequence learning problem. The source text comprises of the input sentence and the candidate answer and the target text is the reference question. The candidate answers are generated by rule-based methods, details of which are beyond the scope of this paper. As discussed in Section 4, finetuning the models using the questions generated by our approach performs better than using questions from SQuAD dataset.

4 Experimental Evaluation

Datasets: We evaluate the proposed approach for temporal QG using history text intended for diverse audience and focusing on different topics such as historical accounts of famous personalities (e.g. Napoleon), important phenomenon (e.g., Fas-

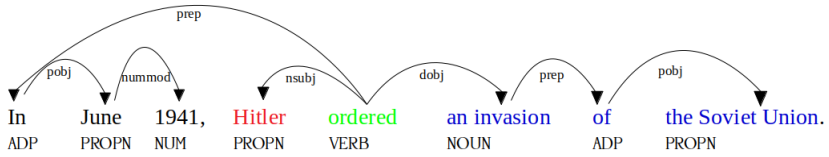

Figure 2: Dependency parse tree using Spacy for Template #2 example sentence

cism), battles, wars, and global conflicts. We use 3 chapters from history text books as well as 186 Wikipedia articles on historical topics. The pointers to the book chapters and Wikipedia articles are provided as a part of supplementary data.

Input articles	Sentences	Sentences with Time expressions	Templates	Questions	Avg. Q_{acc}
189	31538	12460	7	2480	84.07

Table 2: Template-based approach details

We use two different datasets of temporal questions for the second stage of our approach i.e. finetuning experiments with ProphetNet and UniLM. First dataset consists of 868 question generated by our template based approach for training and 217 for validation of the DL model finetuning. The second dataset consists of temporal questions (i.e., questions with explicit date-time expressions as well as implicit expressions such as *before, after, during* etc.) extracted from SQuAD dataset. The resulting subset of SQuAD dataset consists of 35794 questions as training set and 2492 questions as the validation set. The dataset used for experimentation is available for research purposes upon email request.

Finetuning Details: To evaluate the effectiveness of finetuning the DL models for temporal QG task, we use two different models, ProphetNet and UniLM.

ProphetNet: The input provided for finetuning ProphetNet is in the form of *answer [SEP] narrative* and output is the generated question. We use the hyperparameter values as suggested by the authors on their github page[1] for finetuning task. We use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.00001. We set the dropout value as 0.1 and train for 10 epochs.

UniLM: The input provided for finetuning UniLM is in the form of *narrative [SEP] answer* and output is the generated question. Here also, we use the hyperparameter values as suggested by the authors on their github page[2] for finetuning task. We use BERT Adam optimizer (BERT version of the Adam algorithm with weight decay fix) with a learning rate of 0.00002. We train UniLM for 10 epochs.

Evaluation methodology: We employ both automated and human expert evaluation. For human evaluation, we asked experts (people well versed with English language and Global history) to mark the generated question as acceptable or non-acceptable following the human evaluation in Heilman and Smith (2010). A question is marked as acceptable if it is grammatically correct, readable, sensible and not too vague. More details can be found in the guidelines proposed by Heilman and Smith (2010). For human evaluation, we use a random sample of 100 generated questions for each experimental setting.

Q_{acc} metric is used for the percentage of generated temporal questions which are found acceptable by a human expert. $Q_{temporal}$ measures the fraction of temporal questions generated with respect to total number of generated questions. For automated evaluation, we use BLEU-4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE-L (Lin, 2004) which are standard metrics for natural language generation tasks. The questions generated by the template based approach are used as reference questions for calculating these automated metrics.

Results and Discussion: The evaluation details of our template based approach (i.e., the stage 1 of our approach) are given in Table 2. In the corpus of 31538 sentences, there are 12460 sentences that contain a temporal expression (timex). The template based approach (i.e., the stage 1 of our approach) generates 2480 questions from this corpus with an acceptability rate of 84.07%. In this work, we have not considered the entity coreference resolution. A large number of sentences do not get considered by the templates if a slot/parameter (e.g., SUB) of a template contains a pronoun. We plan to consider entity coreference resolution (Patil et al.,

[1] https://github.com/microsoft/ProphetNet

[2] https://github.com/microsoft/unilm/tree/master/unilm-v1

Template	Q_{acc}
template #1	94.71
template #2	82.45
template #3	78.24
template #4	73.33
template #5	60.76
template #6	89.79
template #7	80.08

Table 3: Template-wise acceptability scores (in %)

2018; Gupta et al., 2018) as part of future work.

Template wise acceptability scores are given in Table 3. Since no reference questions are available for this dataset, we evaluate the performance of template based approach through human experts. We keep aside 1000 sentences with timex that generated acceptable questions as the test set for evaluation of the second stage of finetuning DL models.

Pre-trained models of ProphetNet (denoted by P_{pre}) and UniLM (U_{pre}) for general purpose QG task are used as baselines. Let $P_{ft}^{T}, P_{ft}^{S}, P_{ft}^{TS}$ denote ProphetNet base models finetuned for temporal QG using the template based questions, SQuAD temporal questions, and the combined set of template-based as well as SQuAD temporal questions respectively. Similar notation is used for UniLM (U).

Model	B-4	M	R-L	$Q_{temporal}$	Q_{acc}
P_{pre}	0.37	0.29	0.59	53.85	80.00
P_{ft}^{T}	0.84	**0.61**	**0.93**	97.49	**99.00**
P_{ft}^{S}	0.36	0.29	0.60	70.00	91.00
P_{ft}^{TS}	0.73	0.50	0.85	98.50	**99.00**
U_{pre}	0.39	0.30	0.63	65.00	69.00
U_{ft}^{T}	**0.85**	0.60	**0.93**	**99.20**	97.00
U_{ft}^{S}	0.36	0.30	0.61	71.60	76.00
U_{ft}^{TS}	0.73	0.50	0.85	97.50	94.00

Table 4: Experimental comparison of DL models with pre-training vs. different finetuning settings (Abbr.: B-4 = BLEU-4, M = METEOR, R-L = ROUGE-L)

From Table 4, we observe that ability to generate temporal questions ($Q_{temporal}$) as well as acceptability of the generated questions (Q_{acc}) is low for both the pre-trained DL models, P_{pre} and U_{pre}. Finetuning the base models of ProphetNet as well as UniLM certainly helps to improve their performance on the temporal QG task. We note that SQuAD dataset of temporal questions is significantly larger than the set of template-based questions. Still, for both automated metrics as well as human expert evaluation, the performance of the DL models finetuned with only the template-based questions is significantly better than models that use SQuAD temporal questions.

5 Conclusion

We proposed a two-staged method for temporal QG from history text. First, we use templates motivated by linguistics and domain knowledge to carry out syntactic and semantic transformations to generate temporal questions. Then, the generated temporal questions are used to finetune DL models for QG. We experimentally validated the approach with two different DL models to demonstrate improvement due to finetuning as well as flexibility and robustness of this approach for temporal QG.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Sandro Bauer and Simone Teufel. 2016. Unsupervised timeline generation for wikipedia history articles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2343–2349.

Harsimran Bedi, Sangameshwar Patil, Swapnil Hingmire, and Girish Palshikar. 2017. Event timeline generation from history textbooks. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 69–77.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.

Simon Gottschalk and Elena Demidova. 2019. Eventkg–the hub of event knowledge on the web–and biographical timeline generation. *Semantic Web*, 10(6):1039–1070.

Ajay Gupta, Devendra Verma, Sachin Pawar, Sangameshwar Patil, Swapnil Hingmire, Girish K Palshikar, and Pushpak Bhattacharyya. 2018. Identifying participant mentions and resolving their coreferences in legal court judgements. In *International Conference on Text, Speech, and Dialogue*, pages 153–162. Springer.

Olga Gurevich, Richard Crouch, Tracy Holloway King, and Valeria de Paiva. 2008. Deverbal nouns in

knowledge representation. *J. Log. and Comput.*, 18(3):385–404.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617.

Swapnil Hingmire, Nitin Ramrakhiyani, Avinash Kumar Singh, Sangameshwar Patil, Girish Palshikar, Pushpak Bhattacharyya, and Vasudeva Varma. 2020. Extracting message sequence charts from hindi narrative text. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 87–96.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. Tequila: Temporal question answering over knowledge bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1807–1810, New York, NY, USA. Association for Computing Machinery.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The nombank project: An interim report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Girish Palshikar, Sachin Pawar, Sangameshwar Patil, Swapnil Hingmire, Nitin Ramrakhiyani, Harsimran Bedi, Pushpak Bhattacharyya, and Vasudeva Varma. 2019a. Extraction of message sequence charts from narrative history text. In *Proceedings of the First Workshop on Narrative Understanding*, pages 28–36.

Girish Palshikar, Nitin Ramrakhiyani, Sangameshwar Patil, Sachin Pawar, Swapnil Hingmire, Vasudeva Varma, and Pushpak Bhattacharyya. 2019b. Extraction of message sequence charts from software use-case descriptions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 130–137.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sangameshwar Patil, Sachin Pawar, Swapnil Hingmire, Girish Palshikar, Vasudeva Varma, and Pushpak Bhattacharyya. 2018. Identification of alias links among participants in narratives. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 63–68.

Lilan Peng, Zhen Jia, Qi Dai, and Herwig Unger. 2020. A novel method of complex temporal question generation. In *Developments of Artificial Intelligence Technologies in Computation and Robotics: Proceedings of the 14th International FLINS Conference (FLINS 2020)*, pages 109–116. World Scientific.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Nitin Ramrakhiyani, Swapnil Hingmire, Sangameshwar Patil, Alok Kumar, and Girish Palshikar. 2021. Extracting events from industrial incident reports. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 58–67.

Jannik Strötgen and Michael Gertz. 2015. A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 541–547, Lisbon, Portugal. Association for Computational Linguistics.

Yawei Sun, Gong Cheng, and Yuzhong Qu. 2018. Reading comprehension with graph-based temporal-casual reasoning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 806–817.