Selecting Domain-Specific Concepts for Question Generation With Lightly-Supervised Methods

Yiping Jin and **Phu T. V. Le**
Knorex Pte. Ltd.
1003 Bukit Merah Central
Singapore 159836
{jinyiping,le_phu}@knorex.com

Abstract

In this paper we propose content selection methods for question generation (QG) which exploit domain knowledge. Traditionally, QG systems apply syntactical transformation on individual sentences to generate open domain questions. We hypothesize that a QG system informed by domain knowledge can ask more important questions. To this end, we propose two lightly-supervised methods to select salient target concepts for QG based on domain knowledge collected from a corpus. One method selects important semantic roles with bootstrapping and the other selects important semantic relations with Open Information Extraction (OpenIE). We demonstrate the effectiveness of the two proposed methods on heterogeneous corpora in the business domain. This work exploits domain knowledge in QG task and provides a promising paradigm to generate domain-specific questions.

1 Introduction

Automatic question generation (QG) has been successfully applied in various applications. QG was used to generate reading comprehension questions from text (Heilman and Smith, 2009; Becker et al., 2012), to aid academic writing (Liu et al., 2010; Liu et al., 2012) and to build conversational characters (Yao et al., 2012; Nouri et al., 2011).

In this work, we focus on generating a set of question and answer (Q&A) pairs for a given input document. Possible applications of this task are to automatically generate a Q&A section for company profiles or product descriptions. It can also help the reader to recapitulate the main ideas of a document in a lively manner.

We can coarsely divide QG into two steps: "what to ask" (*target concept selection* and *question type determination*), and "how to ask" (*question realisation*) (Nielsen, 2008).

It is important to view question generation not merely as realising a question from a declarative sentence. When the input is a document, the sentences (and candidate concepts) are of different importance. It is therefore critical for a QG system to identify a set of salient concepts as target concepts before it attempts to generate questions. In this work, we propose two novel target concept selection methods that lead to QG systems which can ask more important questions.

Our approaches are motivated by the conditions for a human reader to ask good questions. In order to ask good questions, he needs to satisfy three prerequisites: 1) good command of the language, 2) good reasoning and analytical skills and 3) sufficient domain knowledge. Some may argue prior knowledge is not necessary because we ask about things we do not know. However, it is no surprise that a professor in computational linguistics may not ask as important and relevant questions in the field of organic chemistry as a second-year chemistry student. What makes the difference is the domain knowledge.

Correspondingly, we hypothesize that a successful QG system needs to satisfy the following requirements: 1) able to generate questions that are grammatical and understandable by humans, 2) able to analyse the input document (e.g. keyword identification, discourse parsing or summarization), and 3)

Proceedings of The 9th International Natural Language Generation conference, pages 133–142,
Edinburgh, UK, September 5-8 2016. ©2016 Association for Computational Linguistics

able to exploit domain knowledge.

Previous works mainly focused on addressing the first two requirements. Researchers tend to prefer systems that ask open domain questions because the dependency on domain knowledge is usually regarded as an disadvantage. Several NLG applications successfully utilized domain knowledge, such as virtual shopping assistant (Chai et al., 2001) and sport event summarization (Bouayad-Agha et al., 2011). However, the domain knowledge that they used are manually constructed by human experts. To the best of our knowledge, this paper is the first work in QG that attempts to utilize domain knowledge obtained in a lightly-supervised manner.

Although we choose QG as the application in this work, the lightly-supervised content selection methods that we propose could also be applied to augment other NLG tasks such as summarization.

In section 2, we present previous works of QG and how we position this work into the full storyline. In section 3, we briefly describe the dataset we use. Section 4 introduces two target concept selection methods based on automatically constructed domain knowledge. Section 5 describes methods to generate Q&A pairs from target concepts. In section 6, we present our experimental results. Lastly, we present conclusions and suggest future directions. The contributions of this paper are:

1. Propose to select target concepts for question generation with lightly-supervised approaches.
2. Demonstrate that the use of domain knowledge helps to ask more important questions.
3. Quantitatively evaluate the impact of different ways to represent and select target concepts on question generation task.

2 Connections with Prior Work

Olney et al. (2012) classified question generation (QG) approaches into two categories: knowledge-poor and knowledge-rich.

The knowledge-poor approaches (Ali et al., 2010; Heilman and Smith, 2009; Kalady et al., 2010; Varga, 2010; Wyse and Piwek, 2009) focus mainly on question realisation. A representative approach was proposed by Heilman et. al (2009). Their system took an "overgenerate-and-rank" strategy. Firstly, they applied manual transformation rules to simplify declarative sentences and to transform them into questions. The system generated different types of questions by applying different transformation rules. Secondly, they utilized a question ranker to rank all the questions generated from a input document based on features such as length, language model and the presence of WH words.

The knowledge-poor approaches suffer mainly from two problems. Firstly, they have difficulty determining the question type (Olney et al., 2012). Secondly, it is difficult to evaluate the importance of the questions with respect to the input document.

In contrast, the knowledge-rich approaches build intermediate semantic representations before generating questions. Knowledge-rich approaches not only address "how to ask" but also propose promising methods to select target concepts to generate questions. Knowledge-rich approaches have the advantage of asking more important questions with the help of specific linguistic phenomena, discourse connectors or topic modelling.

Chen (2009) made use of discourse relations (conditions and temporal contexts) as well as modality verbs to generate questions. His work acknowledged that language understanding is tightly related to question asking (Graesser and Franklin, 1990; Olson et al., 1985). After knowing the discourse relation in the sentence, the system could ask questions like "what-would-happen-if" or "when-would-x-happen" using a handful of question templates. However, the system is limited to asking only condition, temporal and modality questions.

Olney et al. (2012) continued the progress made by Chen (2009). They semi-automatically built a concept graph using 30 abstract and domain-independent relations[1]. To extract the relation triples, they firstly applied semantic role labelling and then labeled the argument A0, A1 or A2 to the desired argument of the relations with a manual mapping created for every frequent predicate in the corpus. To generate questions from conceptual graph, they firstly rendered the relation triple as a declarative sentence. Then they substituted one of the relation nodes with "what" to form the question.

[1] Examples of relations are "after", "enables", "has-consequence", "requires", "implies".

Becker et al. (2010) utilized summarization to select key sentences for QG. Internally, the summariser identifies key concepts, links the concepts and selects the important ones through concept graph analysis.

Chali and Hasan (2015) employed similar sentence simplification and transformation pipeline as the knowledge-poor system proposed in Heilman (2009). However, the system performed topic modelling to identify subtopics of the document. It then ranked the questions based on how well they align towards the subtopics.

Our approach belongs to knowledge-rich category and is most similar to Becker et al. (2010) and Chali and Hasan (2015). However, these two systems do not take domain knowledge into consideration when selecting the target concepts. When the input document contains multiple topics, the underlying summarization and topic modeling methods may not select a balanced list of concepts (Gupta and Lehal, 2010; Lu et al., 2011). Instead of relying on the input document alone, we also exploit automatically constructed domain knowledge to select concepts that are important not only to the input document, but also to the underlying domain.

3 Datasets

We make use of two datasets obtained from the Internet. One is 200k company profiles from CrunchBase. Another is 57k common crawl business news articles. We refer to these two corpora as "Company Profile Corpus" and "News Corpus". Each article in News Corpus is also assigned a subcategory by editors (e.g. credit-debt-loan, financial planning, hedge fund, insurance.). There are altogether 12 subcategories.

We randomly selected 30 company profiles and 30 news articles for manual evaluation. The rest of the datasets are used for development.

4 Target Concept Selection

We propose two target concept selection methods based on the following intuitions:

1. Target concepts shall contain important semantic roles (e.g. company name, product name).
2. Target concepts shall contain important semantic relations (e.g. merger, acquisition).

Whether a target concept is important depends not only on itself, but also on the input document and the domain. Hence, we choose to rely primarily on contextual statistics calculated in a corpus instead of human-crafted knowledge in the form of annotated data, lexicons or rules.

4.1 Role-Based Target Concept Selection

Our role-based concept selection method identifies different semantic roles and ask questions about them. This method is inspired by Wikipedia Infobox. Wikipedia Infobox contains key facts (concepts) of the entities. Extracting infobox-like information prior to generating questions solves the two problems of knowledge-poor QG systems. Firstly, we can easily determine the correct question type by knowing the semantic class. For example, for "customer" and "competitor", it is natural to ask a "Who" question while for "product" we will ask a "What" question. Secondly, because we extract concepts defined in Wikipedia Infobox, they are by nature important. Therefore the system is less likely to generate trivial or unrelated questions.

We could have chosen to manually define extraction rules to perform information extraction. However, such method is not portable to other domains. Suppose we build a rule-based QG system for company profiles, if we want to port it to product descriptions, we need to rewrite almost all the rules. We prefer a system that takes as little manual supervision as possible, yet able to capture the important semantic roles in a domain.

We employed bootstrapping to mine semantic roles. Bootstrapping is not limited to a predefined set of roles, but can adapt itself based on the seed words the user provides. We used Basilisk (Thelen and Riloff, 2002) to perform bootstrapping. Basilisk was originally designed to mine semantic lexicons. As shown in figure 1, Basilisk takes a small set of seed nouns for each semantic class, learns the patterns that extract these nouns and uses the patterns to extract more nouns playing the same semantic role. The authors applied this system on MUC-4 corpus and demonstrated it was able to learn high-quality semantic lexicons for multiple categories.

We used Basilisk to learn extraction patterns for different semantic categories. We chose the categories based on the frequency and whether we felt

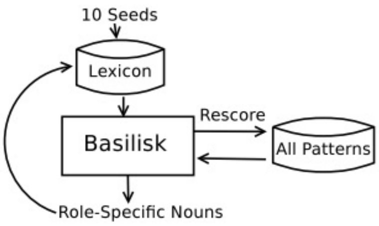

Figure 1: Basilisk algorithm.

the category is important. For this work, we used the following categories: company, location, product/service, customer, partner and date. Following Phillips and Riloff (2007), we only used patterns whose anchor is a verb. We empirically tuned the number of iterations for bootstrapping to avoid domain drifting. The number of iterations in our experiment ranged from 50 to 500.

Note that some of the categories (company, location and date) can also be identified using named entity recognisers (NER) trained on annotated corpora. The difference between bootstrapping and NER is bootstrapping determines the semantic class of a word not by the surrounding window, but by the semantic role it plays. Since it is not the focus of this work, we neither use information from NER, nor compare the accuracy of our bootstrapping method with NER systems.

Basilisk tends to prefer low frequency terms that occur only with patterns in the pattern dictionary. In our experiment, the highest ranked locations were "Rijsenhout", "Dunston" and "Endicott". All of them are little-known towns. The low frequency terms did not provide robust statistics and easily caused domain drift. We modified the original formula (formula 1) to boost more frequent candidate words (formula 2). [2] We do not add 1 to F_j, so all the infrequent patterns that co-occur with only one candidate word will be ignored. We take square root to the denominator P_i to encourage words that co-occur with more patterns. Table 1 shows example words learned for each semantic category along with the top patterns in the corresponding category.

$$AvgLog(word_i) = \frac{\sum_{P_i}^{j=1} log_2(F_j + 1))}{P_i} \qquad (1)$$

Company:	*Communications, Electronics, Networks, Energy, Media, Packaging* <SUBJ>_passive_verb(base) <SUBJ>_active_verb(offer) noun(subsidiary)_prep_of_<POBJ>
Location:	*East, Africa, Republic, Asia, Zealand, Kingdom, America, Europe* passive_verb(base)_prep_in_<POBJ> passive_verb(headquarter)_prep_in_<POBJ> noun(office)_prep_in_<POBJ>
Product:	*equipment, devices, food, material, electronics, infrastructure, vehicles* active_verb(provide)_<DOBJ> noun(manufacture)_prep_of_<POBJ> active_verb(sell)_<DOBJ>
Customer:	*consumers, manufacturers, professionals, organizations, retailers, agencies* active_verb(serve)_<DOBJ> active_verb(provide)_prep_to_<POBJ> active_verb(enable)_<DOBJ>
Partner:	*alliance, partnership, agreement, relationship, shareholding, royalty* active_verb(sign)_<DOBJ> noun(alliances)_have_<DOBJ>
Date:	*March, August, 2009, 2010* passive_verb(found)_prep_in_<POBJ> active_verb(announce)_prep_on_<POBJ> active_verb(introduce)_prep_during_<POBJ>

Table 1: Example semantic lexicon entries and extraction patterns.

$$AvgLog^*(word_i) = \frac{\sum_{P_i}^{j=1} log_2(F_j))}{\sqrt{P_i}} \qquad (2)$$

We used the bootstrapped patterns to extract semantic roles. The system first identifies all the noun phrases in the input document. A noun phrase will be tagged if it triggers one of the patterns in the pattern dictionary. [3] We noted that a few general patterns also appeared in the pattern dictionary (e.g. <PRODUCT>_active_verb(include). Subsequently all the subject of the trigger "include" will be regarded as "product"). This may cause problem when we determine the question word based on the semantic type. However, we did not manually edit the bootstrapped pattern dictionary, trying to adhere to our lightly-supervised paradigm.

[2] P_i is the number of patterns that extract $word_i$, and F_j is the number of distinct category members extracted by pattern j.

[3] We also tried to restrict the head word of the noun phrase to appear in the bootstrapped lexicon. However, it will reduce the recall significantly.

4.2 Relation-Based Target Concept Selection

Our second approach selects salient relations as target concepts. Traditionally, relation extraction systems worked only for predefined relation types and required sizeable training data for each type of relation (GuoDong et al., 2005). Open Information Extraction (OpenIE) becomes the right choice because we neither want to limit the types of relations, nor want to spend many hours annotating training data.

OpenIE systems extract <subject, relation, object> triples using surface, part-of-speech or dependency patterns (Fader et al., 2011; Angeli et al., 2015). Some OpenIE implementations also provide confidence measure for the extracted triples. However, this measure only evaluates the validity of the triples, but not the importance. Balasubramanian et al. (2013) observed that one of the major error sources of OpenIE systems was generating trivial and not informative triples.

We borrowed idea from an early work in semi-supervised information extraction to rank the relation triples based on domain relevance. Riloff (1996) proposed to rank patterns based on unlabelled relevant and irrelevant corpora. A pattern is regarded important if it occurs relatively frequently in the relevant corpus and much less frequently in the irrelevant corpus. She used the $RlogF$ score (formula 3) to rank all the patterns.

$$RlogF_i = log_2(relfreq_i) * P(relevant|pattern_i) \quad (3)$$

We first ran OpenIE [4] on News Corpus and extracted roughly 1.7 million relation triples. Extending the idea of Riloff (1996), we ran one-versus-all experiments for each subcategory. In each run, we treated the documents in one subcategory as the relevant corpus and the rest as irrelevant corpus. Every relation phrase would receive a $RlogF$ score for each subcategory (it received 0 score for subcategories where it did not appear in). If a relation phrase appeared in multiple subcategories, we simply took the highest $RlogF$ score it received as the final score. More formally, we used formula 4 to calculate the salience for each relation phrase. Where $count_{i,j}$ is the number of times relation phrase i appears in documents in subcategory j while $count_i$ is the number of times relation phrase i appears in the whole News Corpus.

Hedge Fund	Investing
lose value in be underwriter for pend against	have trade between cross below represent premium to
Stocks	**Retirement Planning**
be pay on trade dividend on release earnings on	retire at be underfund by contribute at_time
Credit Debt Loans	**Financial News**
contribute from be cut at_time downgrade	arrest in outraise have donate

Table 2: Top relation phrases for selected subcategories.

$$RlogF_i^* = \arg\max_j (log_2(count_{i,j}) * \frac{count_{i,j}}{count_i}) \quad (4)$$

Table 2 shows the top relation phrases for selected subcategories.

We measured the salience of each triple based on information collected on sentence, triple and word level. The $RlogF$ score measures the relevance of a *triple* to a domain. We denote this score as S_{triple}. We also used LexRank (Erkan and Radev, 2004), a summarization algorithm to calculate the salience of the source *sentence* where the question is generated. We denote this score as S_{sent}. Lastly, we used TF-IDF scores of the triple's subject head *word* to estimate the importance of the subject. We denote this score as S_{subj}.

We also incorporated trigram language model score S_{lm} of the triple [5] to ensure the fluency of the generated QA pairs. The final score of a triple is calculated as linear combination of the individual scores. We empirically tuned the weights of the terms and obtained the final equation: [6]

$$S = 2 \cdot S_{triple} + 1 \cdot S_{sent} + 0.3 \cdot S_{subj} + 10 \cdot S_{lm} \quad (5)$$

5 Question Generation From Concepts

We used SimpleNLG (Gatt and Reiter, 2009) to realise questions for both role-based and relation-

[4] We used the implementation of Angeli et al. (2015).

[5] We did not calculate language model scores based on generated questions because our language model is trained on a large News Corpus, where questions are relatively rare.

[6] Scores are not normalized to [0,1], so the weights cannot be directly interpreted as the contribution of each component.

active_verb(offer) dobj(PRODUCT)
What does Zoho offer?
Zoho offer Office Suite.
passive_verb(acquire) prep_in(DATE)
Q: When was StumbleUpon acquired in?
A: StumbleUpon was acquired in May 2007.
passive_verb(acquire) agent(COMPANY)
Q: StumbleUpon was acquired by whom?
A: Ebay.

Table 3: Sample output of role-based QG.

based systems. SimpleNLG is a natural language generation framework which has been widely used for summarization, sentence simplification and data-to-text generation (Gatt et al., 2009; Genest and Lapalme, 2010). SimpleNLG can also transform declarative sentences to questions simply by declaring the interrogative type.

For role-based QG, we proceed to generate question if at least one semantic role is extracted from the sentence. We also identify from the sentence the subject, direct and indirect object and open clausal complement. We choose one of the noun phrases as answer phrase [7] and determine the question word (Who, What, When, Where, How many, How much) based on the semantic type of the answer phrase. Table 3 shows examples of Q&A pairs role-based QG generated together with the patterns that extracted the answer phrase.

For relation-based QG, we proceed to generate questions from a triple if the triple's final score is above 1.0. We set the maximum number of questions for an input document to 15.

The triples are in <subject, relation, object> format. However, the "object" of the triple is not always the direct object or indirect object of the sentence. It can be an object of a preposition or even a verb compliment. As observed by Genest and Lapalme (2010), the syntactical roles known to SimpleNLG are not the same as those known to a dependency parser. There is a need to treat the arguments differently based on their syntactic roles. We followed Genest and Lapalme (2010)'s approach to build noun phrase, prepositional phrase, verb compliment and verb phrase using SimpleNLG.

<Mr. Gibbs, consulting with, White House chief of staff>
Who is consulting with the White House chief of staff?
Mr. Gibbs.
<estimated cost, is, $6.65 billion>
How much is the estimated cost?
$6.65 billion for the 43 banks.
<finance minister, post, link to satirists video>
What did the finance minister post?
A link to satirists video on affair on Twitter.

Table 4: Sample output of relation-based QG.

We followed algorithm 1 to select the answer phrase (subject, object or none if it is a Yes/No question). If the answer phrase is a named entity, we choose the question word according to the entity type. Table 4 shows example relation triples and the Q&A pairs generated from the triples.

Algorithm 1 Algorithm to select the answer phrase

```
if relation is a single frequent verb (e.g. do, go) then
    generate Yes/No question
else if object is a named entity then
    select object as answer phrase
else if subject is a named entity then
    select subject as answer phrase
else if object is longer than subject then
    select object as answer phrase
else
    select subject as answer phrase
end if
```

6 Evaluation

We benchmarked our two systems with Heilman and Smith(2009), which is often used as a baseline for later QG systems [8]. Heilman's system took an overgeneration approach which relied on a question ranker to rank the Q&A pairs. We noted that many top questions the system generated are near duplicates of each other [9]. Hence, we manually removed the near duplicate Q&A pairs before the evaluation and kept only the ones with the highest score.

[7] The term "answer phrase" refers to phrases which may serve as targets for questions, and therefore as possible answers to generated questions.

[8] The source code is available at *www.ark.cs.cmu.edu/mheilman/questions/*.

[9] Generated by applying different question templates on the same source sentence. E.g. "Q: Is Windows Microsoft's product? A: Yes." and "Q: Whose product is Windows? A: Microsoft".

We generated questions with the three systems (Heilman, role-based QG and relation-based QG) on the evaluation set, which consists of 30 company profiles and 30 news articles.

6.1 Method

Following 2010 Question Generation Shared Task Evaluation Challenge (QG-STEC) (Boyer and Piwek, 2010) Task A[10], we assigned individual scores for different aspects to assess the quality of the generated question and answer pairs.

Besides the five criteria used in QG-STEC[11], we added another measure "importance" as we prefer questions that ask about the main idea of the document. We also modified the "specificity" criterion to require the question to be sufficiently specific. A question like "Tell me about IBM." is not specific enough and "What system does IBM provide?" is preferred in our evaluation.

The "specificity", "syntax", "semantics", "importance" and "question type correctness" scores are assigned for each question. They receive a binary score (0 for unacceptable and 1 for acceptable/good).

The "overall" and "diversity" scores are assigned for the set of questions a system generated for an input document. They receive a score between 0 (worst) to 3 (best). 0 means "unacceptable", 1 means "slightly unacceptable", 2 means "acceptable" and 3 means "good". The "overall" score is not an average of the individual scores. It is the subjective judgement on whether the set of Q&A pairs resembles the Q&A pairs a human would construct after reading the same document. We assign high "overall" score if the individual questions are of good quality and the set of questions covers the main ideas of the input document.

We invited two human judges to rate all the Q&A pairs independently. Both of the judges are native English speaker and are not involved in the development of this work. The judges were asked to read the input document before rating the Q&A pairs. They blindly rated the system output without being told which system generated the Q&A pairs.

Measure	κ	% Agreement
Overall	0.51	(0.82)
Specificity	0.18	(0.77)
Syntactic	0.11	(0.85)
Semantic	0.18	(0.79)
QType	0.27	(0.87)
Importance	0.10	(0.50)
Diversity	0.80	(0.91)

Table 5: Inter-Rater reliability.

	Heilman		Role-Based		Relation-Based	
Corpus	Prf.	News	Prf.	News	Prf.	News
Overall	1.65	1.9	1.67	1.7	**1.85**	1.88
Diversity	**2.15**	2.27	1.68	1.98	2.1	2.28
Specificity	0.84	0.88	0.83	0.76	**0.93**	0.87
Syntactic	0.86	0.89	0.88	0.92	**0.93**	**0.95**
Semantic	0.82	0.87	0.83	0.83	**0.90**	0.86
QType	0.86	0.9	0.9	0.84	**0.93**	**0.94**
Importance	0.87	0.85	0.87	0.86	**0.92**	**0.92**

Table 6: Mean ratings across different systems and genre. "Prf." denotes results on the 30 company profiles in the evaluation dataset and "News" denotes results on the 30 news articles in the evaluation dataset. The best score for each measure is bolded. If there is a tie for the best score (difference <1%), both scores are underlined.

We used weighted Cohen's κ to measure inter-rater reliability between the two judges. For "overall" and "diversity" scores, we penalized only when the scores assigned by the two annotators differed for more than 1. Table 5 show both κ and percentage of agreement between them.

Although κ is consistently low, the judges assigned the same score about 80% of the times (except for the importance measurement). There are two main reasons for the low κ score. Firstly, both the annotators assigned 1 (acceptable) for most questions, making the probability of random agreement very high. Secondly, we observe annotator 1 is consistently more generous than annotator 2 when assigning scores. Most of the disagreement cases consist of annotator 1 assigning 1 (acceptable) and annotator 2 assigning 0 (unacceptable).

6.2 Results

Table 6 presents the mean ratings of the three systems assigned by the two human judges.

We can observe that relation-based QG outperformed the other two systems by large margin

[10] Task A is "Question Generation from Paragraph", while task B is "Question Generation from Sentence".

[11] The five criteria are for "specificity", "syntax", "semantics", "question type correctness" and "diversity"

on Company Profile Corpus. For News Corpus, relation-based QG and Heilman's system performed roughly equally well. Relation-based QG outperformed Heilman's system in terms of "question type" and "importance" on both corpora, confirming that exploiting domain knowledge helped QG systems to ask more important questions.

Our two systems also generated more grammatical Q&A pairs. Heilman's system relied heavily on manual transformation rules on the parse tree to simplify sentences. Instead of trying to remove unimportant constituents (e.g.: relative clauses, temporal modifiers), our systems focused on important concepts and generated questions about them. As a result, the questions our systems generated are often more concise compared to the questions generated by Heilman's system. The average length of questions generated by role-based and relation-based QG were 7.4 and 9.1 words. Heilman's system generated questions with average length of 14.4 words, 95% and 58% longer.

The performance of role-based QG was lackluster. It managed to obtain similar scores as the baseline on Company Profile Corpus, yet still lagging behind relation-based QG. On News Corpus, it performed noticeably worse than the other two systems.

Why relation-based QG performs better than role-based QG? OpenIE triples have been widely used in different tasks, including question answering, information retrieval and inference (Angeli et al., 2015). Their advantage is that they are concise and yet are able capture either a static relation or an event. It is relatively simple to realise sentences from relation triples and we do not need to refer to the original sentence to realise the questions.

We identified two major problems with the role-based approach. Firstly, not all sentences containing an important semantic role should be considered for QG. Some sentences only mention the semantic role briefly, making it difficult to generate self-contained questions. That is why relation triples might be a more preferable unit than single semantic roles to represent target concepts. Secondly, although we used lightly-supervised method, we still need to handpick the semantic categories. For company profiles, it is acceptable because the number of candidate concepts are fewer. For news articles, the categories we predefined may fail to cover the variety of topics (E.g. semantic types like stock name, funding rounds are not covered in our list).

While individual questions received relatively high scores (>80%) across different measures, none of the three systems managed to obtain comparable overall score (the highest being 63%). This suggests possible directions for future work to select, organize and present a set of questions generated from a text document in a meaningful manner to replace manually compiled FAQs.

7 Conclusions and Future Works

Motivated by the prerequisites for humans to ask good questions, we proposed two target concept selection methods for question generation (QG) that acquire and exploit domain knowledge.

We divided QG into two steps: firstly to extract target concepts in the form of semantic roles or relation triples, secondly to ask questions about the extracted concepts. Aiming to make the approach general and easily adaptable, both target concept selection approaches are lightly-supervised and do not require manually written rules or lexicons.

One of our proposed systems, relation-based QG, was able to generate more important questions on heterogeneous corpora, showing the feasibility of building a domain-specific question generation system without heavy human supervision. By focusing on the most important concepts, our systems could also to ask more concise and grammatical questions.

In future work, we plan to benchmark our systems with other knowledge-rich QG systems such as Olney et al.(2012), Becker et al.(2010) and Chali and Hasan.(2015). We want to quantitatively evaluate the advantage of using domain knowledge over relying on content analysis of the input document alone. We also aim to generate high-level questions that are beyond single sentence and to learn paraphrases of questions from community-based Q&A websites.

Acknowledgments

We would like to thank Tang Wei Ren and Hoe Xiu Ming for helping to evaluate the questions; as well as Dr. Justin Choo and the anonymous reviewers for their helpful feedback. This work was supported by Capabilities Development Grant – Technology Innovation (CDG-TI) offerred by SPRING Singapore.

References

Husam Ali, Yllias Chali, and Sadid A Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67.

Gabor Angeli, Melvin Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL*, pages 26–31.

Niranjan Balasubramanian, Stephen Soderland, Oren Etzioni Mausam, and Robert Bart. 2013. out of the box information extraction: a case study using bio-medical texts.

Lee Becker, Rodney D Nielsen, Ifeyinwa Okoye, Tamara Sumner, and Wayne H Ward. 2010. Whats next? target concept identification and sequencing. In *Proceedings of QG2010: The Third Workshop on Ques-tion Generation*, page 35.

Lee Becker, Sumit Basu, and Lucy Vanderwende. 2012. Mind the gap: learning to choose gaps for question generation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 742–751. Association for Computational Linguistics.

Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. 2011. Content selection from an ontology-based knowledge base for the generation of football summaries. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 72–81, Nancy, France, September. Association for Computational Linguistics.

Kristy Elizabeth Boyer and Paul Piwek. 2010. Proceedings of qg2010: The third workshop on question generation.

Joyce Yue Chai, Malgorzata Budzikowska, Veronika Horvath, Nicolas Nicolov, Nanda Kambhatla, and Wlodek Zadrozny. 2001. Natural language sales assistant-a web-based dialog system for online sales. In *IAAI*, pages 19–26.

Yllias Chali and Sadid A Hasan. 2015. Towards topic-to-question generation. *Computational Linguistics*.

Wei Chen, Gregory Aist, and Jack Mostow. 2009. Generating questions automatically from informational text. In *Proceedings of the 2nd Workshop on Question Generation (AIED 2009)*, pages 17–24.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Albert Gatt and Ehud Reiter. 2009. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 90–93. Association for Computational Linguistics.

Albert Gatt, Francois Portet, Ehud Reiter, Jim Hunter, Saad Mahamood, Wendy Moncur, and Somayajulu Sripada. 2009. From data to text in the neonatal intensive care unit: Using nlg technology for decision support and information management. *Ai Communications*, 22(3):153–186.

Pierre-Etienne Genest and Guy Lapalme. 2010. Text generation for abstractive summarization. In *Proceedings of the Third Text Analysis Conference, Gaithersburg, Maryland, USA. National Institute of Standards and Technology*.

Arthur C Graesser and Stanley P Franklin. 1990. Quest: A cognitive model of question answering. *Discourse processes*, 13(3):279–303.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3):258–268.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report, DTIC Document.

Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das. 2010. Natural language question generation using syntax and keywords. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 1–10. questiongeneration. org.

Ming Liu, Rafael A Calvo, and Vasile Rus. 2010. Automatic question generation for literature review writing support. In *Intelligent Tutoring Systems*, pages 45–54. Springer.

Ming Liu, Rafael A Calvo, and Vasile Rus. 2012. G-asks: An intelligent automatic question generation system for academic writing support. *Dialogue and Discourse: Special Issue on Question Generation*, 3(2):101–124.

Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. 2011. Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. *Information Retrieval*, 14(2):178–203.

Rodney D Nielsen. 2008. Question generation: Proposed challenge tasks and their evaluation. In *In Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge. Arlington, VA*.

Elnaz Nouri, Ron Artstein, Anton Leuski, and David R Traum. 2011. Augmenting conversational characters with generated question-answer pairs. In *AAAI Fall Symposium: Question Generation*.

Andrew M Olney, Arthur C Graesser, and Natalie K Person. 2012. Question generation from concept maps. *Dialogue and Discourse*, 3(2):75–99.

Gary M Olson, Susan A Duffy, and Robert L Mack. 1985. Question-asking as a component of text comprehension. *The psychology of questions*, pages 219–226.

William Phillips and Ellen Riloff. 2007. Exploiting role-identifying nouns and expressions for information extraction. In *Proceedings of International Conference on Recent Advances in Natural Language Processing*, pages 165–172. Citeseer.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049.

Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 214–221. Association for Computational Linguistics.

Andrea Varga. 2010. Le an ha 2010 wlv: A question generation system for the qgstec 2010 task b. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 80–83.

Brendan Wyse and Paul Piwek. 2009. Generating questions from openlearn study units.

Xuchen Yao, Emma Tosch, Grace Chen, Elnaz Nouri, Ron Artstein, Anton Leuski, Kenji Sagae, and David Traum. 2012. Creating conversational characters using question generation tools. *Dialogue and Discourse*, 3(2):125–146.