Identifying Missing Information in Documents Using Graph-Based Approaches

Aylin Gunal, Hustin Cao, Samika Gupta
{gunala, hustin, samika}@umich.edu
University of Michigan-Ann Arbor

1 Problem Description

1.1 Task

Information retrieval tasks that rely on a knowledge base can be hindered by missing or partial information. In NLP problems, the issue of incomplete information in a knowledge base can translate to a unit or corpus of texts that contains missing or unclear entities and their relations. The identification of missing components in text can have a range of applications, including reading comprehension, text generation and our focus of question generation. Although such tasks that rely on a knowledge base have been studied a great deal, the subtask of classifying missing information, to the best of our knowledge, currently lacks direct evaluation metrics as well as a variety of robust solutions.

We propose using a graph-based clustering algorithm to identify missing information in text. We employ the core clustering method defined by (Le et al., 2008) to generate global clusters of text from all product reviews in our dataset. We then compare individual product reviews to these global clusters and extract the clusters that are least similar to the individual product review as the missing information from the individual product review. In this way, we accommodate for *approximate* similarity between local and global information; unlike the baseline method, which extracts specific key-phrases.

Our code is publicly available[1].

[1] https://github.com/aylingunal/EECS592_CourseProject.git

1.2 System Input and Output

Once the sections of missing or unclear information are identified, this information is fed into the question generation model built by (Majumder et al., 2021), which we will use as a baseline to compare our own system's performance to. We generate relevant questions about each product utilising the missing information as our main method of evaluation between the baseline and the graph-based method.

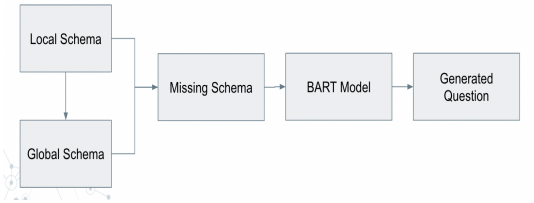

Figure 1: Method Overview of Project

1.3 Challenges

The main challenges faced in this work stemmed from working with a large language model. In order to avoid memory issues using our local environments, we used Google Colab to download and fine-tune the BART model. Training the model itself took up to five hours at a time, more or less depending on the amount of training data. Because of this, we had to scale down the amount of data we used for our experiments.
Additionally, we initially struggled with adapting the code from the baseline work. It was not clearly documented and quite a bit of it (e.g. the model training and evaluation metrics) was missing, and so the baseline work was not immediately reproducible. However, we have implemented the code to the best of our ability and it is all available in our repository.

1.4 Team Contributions

Samika and Hustin worked equally on dataset selection and preprocessing. Aylin wrote the implementation for the core clustering method and extracting the missing schema from the core clusters. Samika adapted the baseline method for extracting the missing schema, and Hustin adapted the local schema generation. Aylin implemented the question generation code and trained the model for the baseline method as well as the model for our clustering method of missing schema extraction. Hustin implemented the evaluation metrics.
On writing the proposal, progress report, and final report, everyone contributed equally. Everyone attended all group meetings.

2 Related Work

Graph-based modeling of text has shown promise in approaching NLP tasks, such as text segmentation (Glavaš et al., 2016), text summarization (Wang et al., 2020) (Jing et al., 2021), and pattern extraction (Majdabadi et al., 2020). We are particularly motivated by the benefit of a graph-based approach's ability to model inter-related data that has the potential to be partitioned into groups of unknown size and common attributes using link prediction between similar nodes (Mutlu and Oghaz, 2019). The identification of missing information in text is a relevant task across various types of text, including the generation of follow-up questions in response to ambiguous utterances (Xu et al., 2019). In long documents, the identification of missing information has typically been approached as a sub-task, and solutions have been implicitly measured by how well they solve the primary task (e.g. clarification question generation). For example, in the realm of question-answer problems, the task of retrieving the relevant information from a knowledge base given a query can be improved by identifying ambiguous queries and asking clarification questions; hence the primary task is the generation of good clarification questions, and the classification of a question as ambiguous or not is an intermediary step. (Majumder et al., 2021) accomplishes the classification of missing information by comparing local context to global context to ascertain which entities are missing locally. Other works have alternatively utilized candidate clarification questions and computed the probability of each candidate's usefulness to the original text to generate good clarification questions ((White et al., 2021), (Rao and Daumé III, 2018)).
In our work, we are introducing graph-based text segmentation using core clustering as a potential method to identify missing information. Due to promising results from our current experiments, there is potential that other graph clustering methods will also yield better results than the current baseline.

3 Methodology

Our methodology is composed of two main components: the identification of missing information in a local context, and generation a question based on that missing information.

3.1 Datasets

This project will use a combination of both the Amazon Products Review Dataset (Jianmo Ni, 2019) and the Amazon Question Answering dataset (Wan and McAuley, 2016) in tandem. Both of these datasets are divided by the category of products (e.g. Electronics, Clothing, Food) and contain the same products, therefore they can be merged together.
The dataset used by the existing works (Majumder et al., 2021) is the Electronics Categories Dataset which has 231,449 questions in the Question Answering dataset and 786,868 products for metadata. However, the size of this dataset is quite large and hard to run and train programs locally in a reasonable amount of time. Hence we will use a smaller subset of the data, in particular a combination of product descriptions from the Beverages, Cooking and Baking, and Herbs and Spices sets.
See Table 1 for an example of the type of data that we are using before creating a local and global schema. The training and validation sets were generated using the transformers library before using the BART model. The training

Example Product	Great Lakes Select Honey, 32-Ounce Bottles (Pack of 3)
Description	Great Lakes Select Honey is a U.S. Grade A , True Source Certified, Extra Light Amber, honey. This honey has a light floral taste reminiscent of the floral smells of its namesake region. This honey is great for mixing in tea, spreading on your morning toast, or using in a recipe. Honey is a natural sweetener for your healthy lifestyle. Check out all our Great Lakes honeys on this site.
User Questions	Is this 100% pure honey? Where in Michigan is this made (I'm guessing that would be near where the honey is harvested)? Is this by any chance Raw Honey?

Table 1: Example of Dataset after initial preprocessing of information

set consisted of 90% of the data and the validation set consists of 10% of the total data. We used a smaller number of products resulting in 386 total description-question pairs. This data will be used to generate the missing schema(s) which will then be inputted into the BART model after the pre-processing methods discussed within sections 3.2.

For the Reviews Dataset, the metadata of the product will be utilized since it provides a succinct summary of each product. The metadata contains the title and description of each product. For the Questions Answering Dataset, we will only be using the corresponding questions for each product as the goal of this project is not related to answer output. The goal is to combine these two datasets together to develop a schema and training and validations sets to use to find the missing information schema and question pairs that will then be trained by the BART model to generate new questions that can be asked. See Table 1 for more information.

3.2 Data Processing

We process the dataset by creating a global schema and local schemas for each product description, and then extracting the missing information of each local schema by comparing to the global schema (this leads to a smaller number of products overall, especially since there are a very limited number of questions). After processing the data and extracting the missing schema, the missing schema will be fed into the question generation model.

We dropped any columns in the dataset that are not directly relevant (i.e. any answers data from QA dataset). Therefore, we mostly extracted only the data that was related to product information (metadata) and user questions. While creating the local schemas, there was some degree of information loss as the questions and the products must be related to each other. This process was similar to an inner join between the QA and the Reviews Metadata datasets which gives the output of the actual number of products that will be available to use in each dataset. These schemas were also divided into sub-categories as well, which we initially utilized as a small testing datasets before scaling up our experiments.

3.2.1 Baseline Method

The baseline method is (Rao and Daumé III, 2018), in which they compare triples between local schema and global schema, taking the set difference as the missing information. A schema for a sentence is defined as a set of one or more triples and one or more key-phrases:

$$schema_s = \{\ element\ \};\ \text{where}$$
$$element \in \{(key\text{-}phrase, verb, relation),\ key\text{-}phrase\}$$

A local schema is defined as the union of all sentence schemas for some product and the global schema is defined as the union of all of the local schemas combined. Now the missing schema, which is our desired schema to identify missing information is derived from the set difference between each local schema and the global schema as a whole:

$$missing\_schema_c = global_K \setminus local_c$$

For more information on the implementation of this method, refer to (Majumder et al., 2021) for further settings and reference code for implementation. The dataset that was mentioned in (Majumder et al., 2021) was changed to the one mentioned above. Other than this step, the steps of the baseline were replicated as close as possible, excluding the next step of inclusion of PPLM with human annotation. An example of the process described above can be referred to in Table 2.

Description Schema	coconut cream, percent pure, carton, thai, thailand, aroy-d, meaning, statements, fda, delicious
Local Schema	coconut milk, carton, coconut, consistency, thick, thick consistency, product, cream, watery, aroy-d, milk, aroy-d coconut
Missing Schema	coconut cream, percent pure, thai, thailand, meaning, statements, fda, delicious

Table 2: Example of the baseline description schema, local schema, and the set difference missing schema.

3.2.2 Graph-Based Method

We aim to improve upon the work by (Rao and Daumé III, 2018) by incorporating *approximate* similarity. We accomplish this by using graph-based clustering methods in order to develop global clusters which represent themes across a higher level class, and assign local nodes to their global counterparts. Global clusters that are not assigned local nodes—otherwise, *approximate* set difference– are identified as the themes of missing information.

We employ the core clustering method introduced in (Le et al., 2008) by adapting it to accommodate text data. The modified algorithm is as follows:

1. **Initialize the graph.** We initialize the graph by tokenizing the input text, i.e. the conglomerate of product reviews, into sentence tokens. Tokens are then stemmed and made lower-case before being assigned each to a single node. Every pair of nodes is connected with an edge that is weighted with the cosine similarity of the two tokens.

2. **Compute the density variation sequence.** At each iteration t, the minimum density of the graph D_t as well as the nodes that have that minimum density M_t are computed. Minimum density is the lowest density across the individual nodes within the graph. Density per node is computed by averaging the weights of the node's associated edges. At the end of an iteration, the nodes in M_t are removed from the graph. This process is repeated until the graph is empty.

3. **Identify the core nodes.** Candidate core nodes are each set of nodes that were removed at the end of an iteration in the previous step. Candidate sets are formally assigned as core nodes if they meet the following conditions:

   (a) $(D_t - D_{t+1})/D_t > \alpha$

   where α is some pre-determined value.

   (b) $k \in N$ s.t. $D_t \in D_{k+1}, D_{k+2}, ..., Dk + \beta$, where $D_{k+1}, D_{k+2}, ..., D_{k+\beta}$ are β successive density values that satisfy condition 1.

4. **Expand clusters.** With core nodes determined, non-core nodes can be assigned to core nodes and clusters can start to be determined. Non-core nodes are added to the core nodes that they have the maximum average cosine similarity to.

We apply the core clustering method to the full set of product reviews in the training set and then assign local nodes to the cluster to which they have the highest average cosine similarity to. Global clusters that do not receive an assignment are used as input to the question generation model as the missing information.

4 Experiments

4.1 Model Training

Once the missing schema is generated, we feed this missing schema into a BART model. The model we used is the conditional generation pre-trained BART base, which is then fine-tuned with the missing schema and associated questions as labels. We train with 10 epochs with a batch size of 4.

4.2 Evaluation Methods

BLEU is a commonly used low-cost automatic machine evaluation technique that is comparable in quality to human evaluators (Papineni

	BLEU-4	METEOR	Google BLEU
Baseline	0.972	0.951	0.974
Core Cluster	0.991	0.962	0.985

Table 3: Results comparing performance of baseline method and core clustering method. Core clustering outperforms baseline method in all metrics.

	Unigram	Bigram
Ground-Truth	0.408	0.806
Baseline	0.416	0.783
Core Cluster	0.434	0.800

Table 4: Sentence level n-gram diversity of questions.

et al., 2002). It uses n-gram precision and matching or counting n-gram pairs that results in a score between 0 to 1 where 1 is the exact copy of the reference and 0 indicates a complete mismatch. The METEOR score is also an automated machine translation evaluation that makes hypotheses based on alignments based on exact, stem, synonym or paraphrase matches from a reference and addresses some of the weakness of BLEU (Denkowski and Lavie, 2014). The Google BLEU (GLEU) score also improves on the BLEU score for single sentences. BLEU was designed for a corpus level measure and has undesirable properties for single sentences. GLEU limits the undesirable properties and is designed for single sentences measurements (Wu et al., 2016).

4.3 Results

Each product description was used as input for both the baseline model and the core cluster model. Afterwards, all generated questions were evaluated using the proposed evaluation methods. From both models, we compare the average scores over all evaluation scores (See Table 3). The evaluations for the baseline model and the core cluster model are abnormally high. The high scores may be attributed to single sentence measurements and repeated question generation. We still note that the core cluster model performs better than the baseline model over all evaluation methods.

Example generated questions are shown in Table 5. The core cluster model for the first product description performs better. The baseline model inquires about carbs even though the product description contains information about carbs. In the second example, the baseline model performs better than the core cluster model. The core cluster model inquires about nutritional content which was described. Overall, the core cluster model inquires for information more related to the context than the baseline method.

We also calculate the n-gram diversity at a sentence level for the questions in the Amazon Question Answering dataset, the baseline generated question, and the core cluster generated question in Table 4. The n-gram diversity of all three data are relatively similar showing that complexity of the generated questions are similar to the ground-truth data.

4.4 Discussion

Our results are not satisfying to our standard as the numbers are abnormal. The generated questions do provide some level of usefulness using missing context. However, many of the generated questions are repeated throughout a number of products. Questions such as "Is this gluten free?" and "What is the expiration date on this product?" are generated for many products, suggesting that there is heavy overlap in the global clusters extracted as missing schemas within the training set. While the question does ask for related information which was not within the context of the description, the information would not be useful. We identify the following areas to improve upon the question generation model. It could possibly be due to the topic of the dataset (food) which leads to repetitive and non-specific questions, which is another point to investigate.

4.4.1 Using Larger and Diversified Data

The Amazon Grocery and Gourmet Food data is used to train the BART models. The Grocery and Gourmet Food consisted of 15,373

Description
Net Carbs 2gTotal Carbs: 19gNon-Impact Carbs: -17gThe -17 non-impact carbs are derived from maltitol (a sugar alcohol) and dietary fiber.
Unsweetened Chocolate, Maltitol (from wheat but gluten free), Cocoa Butter, Cream (Milk), Cocoa Mass, Soy Lecithin, Natural Vanilla, may contain traces of various nuts.
Statements regarding dietary supplements have not been evaluated by the FDA and are not intended to diagnose, treat, cure, or prevent any disease or health condition.
Baseline Model
How many carbs per serving?
Core Cluster Model
How many calories per packet?
Description
Unsulfured blackstrap molasses is more than just a sweetener.
Its full of health promoting vitamins and minerals and makes a great nutritional supplement.
Taken from the second pressing of the cane, all of the white sugar has been removed from this molasses leaving all the trace elements, especially all of those valuable minerals.
Delicious mixed in herbal teas or in hot water with lemon and milk.
There are 3 different grades of molasses.
The first and second grades are fairly sweet.
Blackstrap molasses is the third grade its not very sweet, but it does have the highest vitamin and mineral content of all three grades.
100% certified organic blackstrap molasses.
Some foam at the top is normal.
Statements regarding dietary supplements have not been evaluated by the FDA and are not intended to diagnose, treat, cure, or prevent any disease or health condition.
Baseline Model
What is the shelf life of this product?
Core Cluster Model
What is the nutritional content of this product?

Table 5: Example generated questions from baseline model and core cluster model

questions and 62,243 answers. Metadata was missing for some questions: the assigned product ID was not in the meta file. Additionally, some product IDs in the meta file had no associated questions and answers data. The actual number of questions and answers used was greatly reduced.

Additionally, the category may not be diversified enough with larger classes. Many of the generated questions were the same or very similar. The similarity issue may be caused by using a category that contains many classes with small sizes. We can attempt to resolve the similar generating questions by using a larger category dataset that contains larger classes. The subject matter of the dataset could also be another influence on the results, since food is very different that the technology category that the baseline approach (Majumder et al., 2021) has used.

4.4.2 Human Annotation

Due to time constraint, we were unable to implement a human annotation user study on our results. Currently, qualitative evaluation is best done by human evaluation. Human annotation is important for our product question generation due to the difficulty of evaluating usefulness given the context of the product description. Human annotation is a tedious task that requires long manual work that we were unable to produce for the time constraint. Future work can implement human annotation using Amazon Mechanical Turk.

The human annotation can be used in the Plug and Play Language Model (PPLM) to help our model generate more useful questions. PPLM allows the model to generate more useful questions according to the human annotation without retraining the model.

References

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. Unsupervised text segmentation using semantic relatedness graphs. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Berlin, Germany. Association for Computational Linguistics.

Julian McAuley. Jianmo Ni, Jiacheng Li. 2019. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. *Empirical Methods in Natural Language Processing (EMNLP)*.

Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. 2021. Multiplex graph neural network for extractive text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thang V Le, Casimir A Kulikowski, and Ilya B Muchnik. 2008. Coring method for clustering a graph. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE.

Zahra Majdabadi, Behnam Sabeti, Preni Golazizian, Seyed Arad Ashrafi Asli, Omid Momenzadeh, and Reza

Fahmi. 2020. Twitter trend extraction: A graph-based approach for tweet and hashtag ranking, utilizing no-hashtag tweets. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6213–6219, Marseille, France. European Language Resources Association.

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. 2021. Ask what's missing and what's useful: Improving clarification question generation using global knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4300–4312, Online. Association for Computational Linguistics.

Ece C Mutlu and Toktam A Oghaz. 2019. Review on graph feature learning and feature extraction techniques for link prediction. *arXiv preprint arXiv:1901.03425*.

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.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746, Melbourne, Australia. Association for Computational Linguistics.

Mengting Wan and Julian J. McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. *CoRR*, abs/1610.08095.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Julia White, Gabriel Poesia, Robert Hawkins, Dorsa Sadigh, and Noah Goodman. 2021. Open-domain clarification question generation without question examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 563–570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. Asking clarification questions in knowledge-based question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1618–1629, Hong Kong, China. Association for Computational Linguistics.