Integrating Paraphrasing into the FRANK QA System

Nick Ferguson[1,*], Liane Guillou[1], Kwabena Nuamah[1] and Alan Bundy[1]

[1]School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB

Abstract

We present a study into the ability of paraphrase generation to increase the variety of natural language queries that the FRANK Query Answering system can answer. We choose an English-French backtranslation model to generate paraphrases, which we test using a small challenge dataset. *We conclude that this method is not useful for improving the variety of natural language queries that FRANK can answer.* Based on our observations, we recommend future work in the following directions: (1) allowing the ability to specify a form to paraphrase an input into; (2) constrained paraphrasing to avoid loss of information about query intent; and (3) the need for an automatic evaluation metric which captures semantic similarity, allows syntactic variation, and rewards preservation of query intent.

Keywords

Question Answering, Paraphrasing, Backtranslation

1. Introduction and background

Paraphrasing is the task of producing an output which captures the semantics of a given input, but with different lexical and/or syntactic features. One application of paraphrase generation is Question Answering (QA) [1, 2, 3, 4]. Humans can express queries in many ways, but a given QA system may not cover every variation. They should therefore be robust to this variation: ideally, humans should not have to consider underlying mechanics of a system before interacting with it. Paraphrasing may therefore be employed within QA systems to improve their ability to handle wider varieties of natural language queries.

Paraphrasing has been implemented into QA systems in a variety of ways. Corpora of paraphrase clusters can be used to learn equivalences for relations and entities, and to mine paraphrase operators [1, 2]. Intermediate logical forms can be generated from input utterances, from which paraphrases are generated [3]. Paraphrases can be learned in parallel to the training of neural QA models [4], and Neural Machine Translation (NMT) models can generate paraphrases via backtranslation [5, 6]. Paraphrasing via backtranslation has also shown to improve the prompting of Large Language Models (LLMs) [7]. Motivated by previous success in using paraphrasing for QA, we tested it on the task of transforming input queries, which FRANK cannot parse, into forms which FRANK's parser *can* parse. FRANK is introduced in section 1.1.

In this study, we test the merits of paraphrasing in FRANK, discuss its limitations, and propose future directions for paraphrase generation and evaluation which we believe will generalise

3rd International Workshop on Human-Like Computing, September 28–30, 2022, Cumberland Lodge, Windsor Great Park, United Kingdom

*Corresponding author.

nick.ferguson@ed.ac.uk (N. Ferguson)

well to other QA systems. These are:

1. Allowing a pre-determined target form to be specified for an input.
2. Constraining paraphrasing to avoid replacing key terms, e.g., named entities.
3. Reiterating the need for an improved automatic evaluation metric.

1.1. Frank

The Frank (Functional Reasoner for Acquiring New Knowledge) QA system uses a graph-based algorithm to produce answers to users' queries when no direct lookup is available [8]. Consider the query *'What will be the population of France in 2028?'*. A direct lookup is not possible, so Frank will look up past population data, apply regression and extrapolation over it, and estimate an answer for the user.

Frank parses natural language queries into a set of attribute-value pairs called an *association list*, or *alist*, using a template-based method. While a neural parser has been shown to increase performance [9], it was not able to perform the transformations we are hoping to create via paraphrasing. Questions are answered by recursively decomposing alists into an *inference graph* according to a set of rules; making queries to knowledge bases (KBs); and aggregating query results in a manner determined by query intent.

We can formulate the above query as *'How many people will be living in France in 2028?'*. While this form targets the same quantity, Frank cannot answer this version of the query as its parser is limited to a fixed set of query forms. We aim to address cases like this using paraphrasing. While an improved parser may help solve this issue, we may encounter cases where a user asks about a quantity (e.g., *surface area*) which is stored under a different name in a KB (e.g., *total area*). Paraphrasing should allow us to generate these synonyms, and, at the same time, create syntactic variation. It is for this reason that, should a better parser be implemented, we still require some level of paraphrasing. However, queries may contain terms which we do *not* want to paraphrase, such as named entities (e.g., *United Nations*), and technical terms (e.g., *coefficient of variance*). Paraphrasing could also help troubleshoot a QA system in the case an incorrect answer was returned by confirming that it has understood a query's intent. By repeating a paraphrase of the question to the user, we may be able to rule out misunderstanding of query intent and proceed to look further down the QA pipeline for error. However, this experiment has not been performed and is a speculative benefit.

1.2. Paraphrase generation

We tested paraphrasing using backtranslation with NMT models [10] from Huggingface [11] and the Separator model [12], which specifically aims to change the form of an input query. To paraphrase via backtranslation, a phrase in one language is translated into another (the *pivot* language), then from the pivot language back into the original. We selected pre-existing methods to avoid the labour-intensive task of creating paraphrase templates, as templates would not achieve the coverage of, e.g., backtranslation. Before testing integration of paraphrasing into Frank (section 2), we evaluated different pivot languages and the Separator model in order to choose a single method. Paraphrases were generated from source queries in the LC-QuAD 2.0 dataset [13] and evaluated against a reference (also given in the dataset). We used iBleu [14]

and cosine similarity with sentence embeddings [15] as automatic evaluation metrics. iBLEU is based on BLEU, and by extension n-gram overlap between the candidate and the source or the reference. We (the lead author) performed human evaluation to assess the performance of the automatic metrics. We found that English-French backtranslation produced the highest number of paraphrases which preserved the intent of their source query, according to human judgement. Further details about paraphrase generation and evaluation are given in [16].

2. FRANK-based evaluation

We tested the English-French backtranslation on its ability to generate alternative forms of queries that a user has asked FRANK. We created a small test dataset containing 4 queries (with alists), which are representative of the 4 query types that FRANK can answer (these types are given in [17]). For each of the four queries, we hand-created 5 paraphrases, encoding the same intent as the original query, but introducing different syntax such that FRANK could not parse these human-generated forms. We also introduced synonyms for certain words in some paraphrases (e.g., *total population* for *population*). Each of these human-generated paraphrases are then 're-paraphrased' by the backtranslation model, then passed to FRANK. If, after parsing the 're-paraphrased' query FRANK returns an equivalent alist (i.e., an equivalent set of attribute-value pairs) as that of the original query, then this constituted a success.

3. Results and discussion

Out of the 20 test cases, only 1 candidate paraphrase could be parsed into an alist equivalent to that of the source query. All 20 candidates were adequate, fluent English, but were often only trivially different from the hand-created paraphrases from which they were generated - meaning FRANK still could not parse them. For example, paraphrasing *'What will be the population of France in 2028?'* into *'What will the population of France be in 2028?'*. This highlights the fact that we had no way to specify a target form to paraphrase a given input into.

Backtranslation preserved named entities very well, while SEPARATOR less so. While LLM-based paraphrasing has been shown to produce high quality paraphrases [18], they may not preserve named entities or technical terms better than backtranslation.

We also observed weaknesses in paraphrase evaluation metrics. While we were only evaluating against a single reference, observations about available metrics still apply: no off-the-shelf automatic metric could be found that simultaneously rewarded semantic similarity, syntactic variation, and preservation of query intent. To verify performance of existing off-the-shelf metrics, the lead author performed human evaluation based on preservation of query intent alone, rating paraphrases as adequate (preserving intent), or inadequate (vice versa). We found that there was very weak correlation between automatic and human evaluation, and in the end proceeded to select pivot language on human evaluation alone. By resorting to this, the optimal pivot language (French) only generated quite trivial paraphrases. Other pivot languages which are less similar to English may produce more syntactically varied paraphrases, but differences in the amount of training data used for other translation models meant that overall, they performed significantly worse.

We were limited by the small dataset size in that it only gave us a coarse-grained understanding that the method did not work. A larger dataset will be required to better understand any successes of the method when applied to Frank. However, the proportion of negative results given the small dataset size provided us with a degree of confidence that the method was not suitable. Additionally, human-generated paraphrases were created by one person. A larger dataset should involve multiple people of different backgrounds to create greater variation.

Another limitation, and one which may have affected our interpretation of the result, is the extent to which Frank's parser is limited. Since Frank's parser is very brittle, our analysis that the paraphrasing methods performed poorly is influenced by the fact that there were many adequate paraphrases which could still not be parsed by Frank. Therefore, the paraphrasing methods themselves performed well, but were limited by the sheer brittleness of Frank's parser. One analysis of the paraphrasing methods which is not affected by the performance of Frank's parser is the observation that generated paraphrases were often trivial. While we report the presence of trivial paraphrases as a negative result for Frank, the fact remains that these paraphrases were good, fluent English, and adequately preserved the intent of the query. This has the potential to benefit QA systems with parsers which have coverage greater than Frank's, but are still limited to some extent.

The observation of the trivial similarity of candidate paraphrases links to our discussion about evaluation metrics, and may suggest why iBleu in particular was a poor proxy for measuring paraphrase quality for our use case. Firstly, we must acknowledge that different types of paraphrasing may take place: *syntactic*, where the goal is to change the form of an input sequence and the type that we desired in this experiment; and *lexical*, or *phrasal* paraphrasing, in which certain words or phrases are substituted with synonyms. Reference paraphrases from LC-QuAD 2.0, against which candidate paraphrases were evaluated, were inconsistent in type - some were syntactic, others lexical or phrasal. This naturally affects iBleu scores, which would be weighted lower in the case that the reference paraphrase were of a significantly different form, as opposed to a lexical paraphrase with only one or two words changed.

4. Conclusion

In this study, we found that employing paraphrase generation did not improve the variety of natural language queries that Frank can answer. While generated paraphrases were adequate, Frank's parser remains a bottleneck. Continuation of work on parsing is therefore a more appropriate direction for Frank. We highlight potential future directions for paraphrase generation and evaluation, which are relevant to Frank and wider QA systems. Firstly, we desire control over the form that a source query is paraphrased into, to better match the ability of a given parser. Secondly, we require the masking of named entities and technical terms from paraphrasing - rarer phrases which may be incorrectly paraphrased. Lastly, we discuss the need for an automatic evaluation metric that can reward semantic similarity between a candidate paraphrase and a set of references, promote rich syntactic paraphrasing, and reward preservation of a query's intent. The first recommendation is more Frank-specific, but the following two are more widely applicable. Masking technical terms will be key in domain-specific applications, while high-quality automatic evaluation is critical for analysis of any paraphrasing model.

Acknowledgements

For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission. The authors also wish to thank the reviewers for their feedback.

References

[1] A. Fader, L. Zettlemoyer, O. Etzioni, Paraphrase-driven learning for open question answering, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 1608–1618. URL: https://aclanthology.org/P13-1158.

[2] A. Fader, L. Zettlemoyer, O. Etzioni, Open question answering over curated and extracted knowledge bases, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 1156–1165. URL: https://doi.org/10.1145/2623330.2623677. doi:10.1145/2623330.2623677.

[3] J. Berant, P. Liang, Semantic parsing via paraphrasing, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 1415–1425. URL: https://aclanthology.org/P14-1133. doi:10.3115/v1/P14-1133.

[4] L. Dong, J. Mallinson, S. Reddy, M. Lapata, Learning to paraphrase for question answering, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 875–886. URL: https://aclanthology.org/D17-1091. doi:10.18653/v1/D17-1091.

[5] J. Mallinson, R. Sennrich, M. Lapata, Paraphrasing revisited with neural machine translation, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 2017, pp. 881–893.

[6] J. Wieting, J. Mallinson, K. Gimpel, Learning paraphrastic sentence embeddings from back-translated bitext, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 274–285. URL: https://aclanthology.org/D17-1026. doi:10.18653/v1/D17-1026.

[7] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How Can We Know What Language Models Know?, Transactions of the Association for Computational Linguistics 8 (2020) 423–438. doi:10.1162/tacl_a_00324.

[8] K. Nuamah, A. Bundy, Explainable Inference in the FRANK Query Answering System, in: ECAI 2020, IOS Press, 2020, pp. 2441–2448. doi:10.3233/FAIA200376.

[9] Y. Li, Models to Translate Natural Language Questions to Structured Forms and Back, Master's thesis, University of Edinburgh, 2021.

[10] J. Tiedemann, S. Thottingal, OPUS-MT – building open translation services for the world, in: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, European Association for Machine Translation, Lisboa, Portugal, 2020, pp. 479–480. URL: https://aclanthology.org/2020.eamt-1.61.

[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface's transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).

[12] T. Hosking, M. Lapata, Factorising Meaning and Form for Intent-Preserving Paraphrasing, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 1405–1418. URL: https://aclanthology.org/2021.acl-long.112. doi:10.18653/v1/2021.acl-long.112.

[13] M. Dubey, D. Banerjee, A. Abdelkawi, J. Lehmann, LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham, 2019, pp. 69–78.

[14] H. Sun, M. Zhou, Joint learning of a dual SMT system for paraphrase generation, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Jeju Island, Korea, 2012, pp. 38–42. URL: https://aclanthology.org/P12-2008.

[15] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410. doi:10.18653/v1/D19-1410.

[16] N. Ferguson, L. Guillou, K. Nuamah, A. Bundy, Investigating the use of Paraphrase Generation for Question Reformulation in the FRANK QA system, 2022. URL: https://arxiv.org/abs/2206.02737. doi:10.48550/ARXIV.2206.02737.

[17] K. Nuamah, A. Bundy, C. Lucas, Functional inferences over heterogeneous data, in: International Conference on Web Reasoning and Rule Systems, Springer, 2016, pp. 159–166. doi:10.1007/978-3-319-45276-0_12.

[18] S. Witteveen, M. Andrews, Paraphrasing with large language models, in: Proceedings of the 3rd Workshop on Neural Generation and Translation, Association for Computational Linguistics, Hong Kong, 2019, pp. 215–220. URL: https://aclanthology.org/D19-5623. doi:10.18653/v1/D19-5623.