An Extensible and Reusable Pipeline for Automated Utterance Paraphrases

Auday Berro
University Claude Bernard Lyon 1,
LIRIS UMR5205
Villeurbanne, Lyon, France
auday.berro@univ-lyon1.fr

Mohammad-Ali Yaghub Zade Fard
University of New South Wales
Sydney, Australia
m.yaghoubzadehfard@unsw.edu.au

Marcos Baez
University Claude Bernard Lyon 1,
LIRIS UMR5205
Villeurbanne, Lyon, France
marcos.baez@liris.cnrs.fr

Boualem Benatallah
University of New South Wales,
Sydney, Australia
& LIRIS - Université Lyon 1, France
b.benatallah@unsw.edu.au

Khalid Benabdeslem
University Claude Bernard Lyon 1,
LIRIS UMR5205
Villeurbanne, Lyon, France
kbenabde@univ-lyon1.fr

ABSTRACT

In this demonstration paper we showcase an extensible and reusable pipeline for automatic *paraphrase generation*, i.e., reformulating sentences using different words. Capturing the nuances of human language is fundamental to the effectiveness of Conversational AI systems, as it allows them to deal with the different ways users can utter their requests in natural language. Traditional approaches to utterance paraphrasing acquisition, such as hiring experts or crowdsourcing, involve processes that are often costly or time consuming, and with their own trade-offs in terms of quality. Automatic paraphrasing is emerging as an attractive alternative that promises a fast, scalable and cost-effective process. In this paper we showcase how our extensible and reusable pipeline for automated utterance paraphrasing can support the development of Conversational AI systems by integrating and extending existing techniques under an unified and configurable framework.

PVLDB Reference Format:
Auday Berro, Mohammad-Ali Yaghub Zade Fard, Marcos Baez, Boualem Benatallah, and Khalid Benabdeslem. An Extensible and Reusable Pipeline for Automated Utterance Paraphrases. PVLDB, 14(12): 2839 - 2842, 2021.
doi:10.14778/3476311.3476358

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/AudayBerro/automatedParaphrase.

1 INTRODUCTION

Conversational AI is shifting the way we interact with digital services and devices, turning human-machine interactions into natural language dialogs. For example, users can fulfil their information needs by directly asking "How does COVID-19 spread" to a health chatbot and receive a direct response, instead of browsing a Website or reading a document. Clearly, supporting such dialogs requires more than exposing and interpreting commands in a fixed syntax or grammar, pushing systems to process and interpret user requests in natural language. This represents a challenge as systems now need to deal with the richness of human language, e.g., another user may ask "How is the coronavirus transmitted?" in allusion the previous COVID-19 question, thus illustrating the different ways the same request can be expressed. Failing to correctly identify and process such nuances of human language can have a negative impact on the effectiveness, user experience and ultimately the adoption of Conversational AI systems [18, 28].

Training Conversational AI systems to deal with the expressiveness of natural language often requires collecting a large and linguistically diverse dataset of utterances [12, 30]. Having experts to provide and annotate utterances at scale can be costly and time consuming, reasons that have motivated research into other *utterance paraphrasing methods* [29]. These approaches are generally grouped into those i) relying on deployed *prototypes* to collect utterances directly from users, ii) leveraging *crowdsourcing* to collect paraphrases at scale with non experts, and iii) *automated approaches* that generate paraphrases systematically. All these involve trade-offs between relevant quality metrics, such as diversity and naturalness, operational costs and acquisition times.

Automated paraphrasing is emerging as an attractive technique to address the challenge of fast acquisition of high quality training paraphrasing sets. The literature on automated paraphrases generation covers a wide range of approaches, including probabilistic, hand-written rules and formal grammars models [8], data-driven techniques [15], machine translation techniques [10, 16], and recently approaches that take advantage of contextual representations models (a.k.a embeddings, BERT [6] and USE [4]). Overall, existing automatic approaches still fall behind in terms of quality, with the literature pointing to models often failing to produce sufficiently diverse and semantically related paraphrases [19, 27], and generated paraphrases suffering from issues like grammatical errors and unnaturalness [27]. Thus, as far as quality is concerned, there is some progress to be made [14]. From a practical perspective, issues

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476358

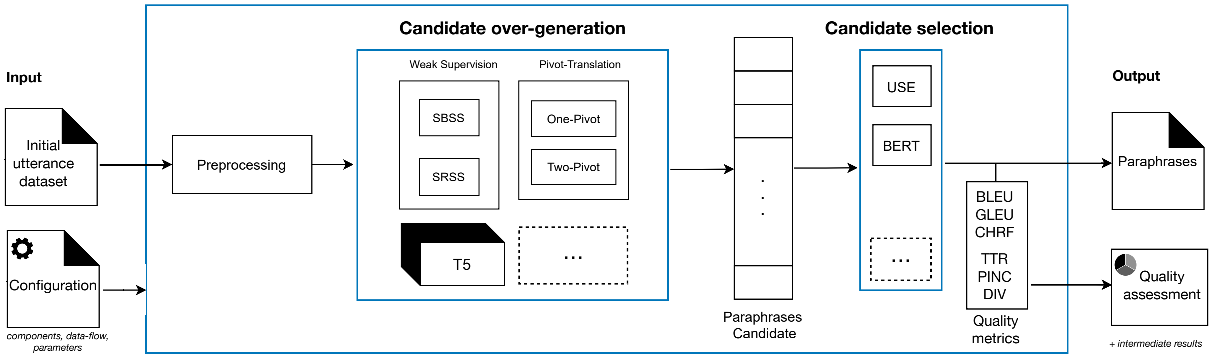


Figure 1: Automated Paraphrase Generation Pipeline Architecture

can also affect the engineering, evaluation and adoption of automatic paraphrasing approaches, such as the availability of training data, computational costs or deployment environment.

In this demonstration we showcase an extensible, reusable and configurable pipeline that unifies, integrates and extends various paraphrasing components. The pipeline aims at supporting the development of Conversational AI systems by facilitating the process of generating paraphrases that are semantically relevant and diverse. Accordingly, the pipeline organises the generation in two steps, namely: i) paraphrase *candidate over-generation*, supporting techniques such as pivot translation, weak supervision and Transformer models, that can be combined to generate candidate paraphrases, and ii) *candidate selection*, with pre-trained language models and techniques to discard semantically irrelevant and duplicates paraphrases candidates. As we will see, the overall architecture also facilitates the process of experimenting, extending and evaluating paraphrasing techniques and pipelines. Our work can support Conversational AI efforts in the data management community, such as effective natural language query (NLQ) systems and conversational data exploration [1, 3].

2 AUTOMATIC PARAPHRASING PIPELINE

The general pipeline and its main components are illustrated in Figure 1. These components and the underlying architecture were designed to i) facilitate the process of paraphrase generation, allowing practitioners and researchers to reuse and tailor configurations ii) support the evaluation and experimentation of competing configurations, and iii) allow the community to extend the pipeline with additional models and techniques. In the following we expand on these features.

2.1 Supporting paraphrase generation

In a nutshell, the pipeline takes an input sentence and generates a list of semantically relevant and diverse paraphrases as output, by performing *candidate over-generation* and *candidate selection*. We organise the pipeline in these two steps to make sure the process can leverage techniques that both expand on paraphrase candidates while also pruning low quality one from the final list. Researchers and practitioners can then decide on the coupling and configuration of the techniques, or reuse existing pipeline configurations.

2.1.1 Candidate over-generation. Over-generation refers to the use of techniques that can be combined to expand on the input sentence to incrementally generate a larger and more diverse set of paraphrase candidates. The current pipeline incorporates three techniques, with extensions to make them configurable.

Weak Supervision [22] is learning approach that automatically creates its own training data through the use of noisy data. This techniques is implemented in the pipeline to generate candidate paraphrases from an input utterance by replacing individual words (verbs, nouns) with their synonyms (from Wordnet). Unlike previous work relying on a similar technique [18], we adopted two strategies for synonym selection and replacement, i.e., the paraphrase candidate is generated by i) replacing each selected token (verb and/or noun) with the WordNet synonym that has the highest semantic relatedness (cosine similarity) within an interval $[\alpha, \beta]$ (strategy *SBSS*), or ii) relaxing the selection to include *all* synonyms above the threshold α (strategy *SRSS*). These strategies can be configured and combined to balance the semantic relatedness of the generated candidates and exploration of diverse paraphrases.

Pivot Translation [2]. The idea behind pivot translation is that two sentences with the same (foreign) translation can be assumed to have the same meaning. Thus, paraphrases can be generated by translating a sentence in source language S into a foreign language F and then back-translating it into S. Informed by previous literature [7, 31], we support the configuration of various parameters of pivot translation that are known to affect the generation of paraphrases. The *paraphrase system*, refers number of pivot languages that can be used to generate paraphrases, a *single-pivot* system when one language is used as pivot, and *multi-pivot* when it is made up of a set of single-pivot systems. Having multiple *machine translation engines* can maximise the chances of getting more diverse paraphrase options [31], and so we support a selection of online neural machine translators (NMTs), such as Google Translate[1], Deepl[2] and MyMemory[3]; as well as pre-trained NMTs like the Huggingface Marian Machine Translator [11, 26], and OpenNMT [13]. The pipeline also supports different *pivot-language levels*, i.e., the number of intermediate pivot languages chained to generate the paraphrases, and the selection of language pairs. By default, the

[1] https://translate.google.com/
[2] https://www.deepl.com/translator
[3] https://mymemory.translated.net/

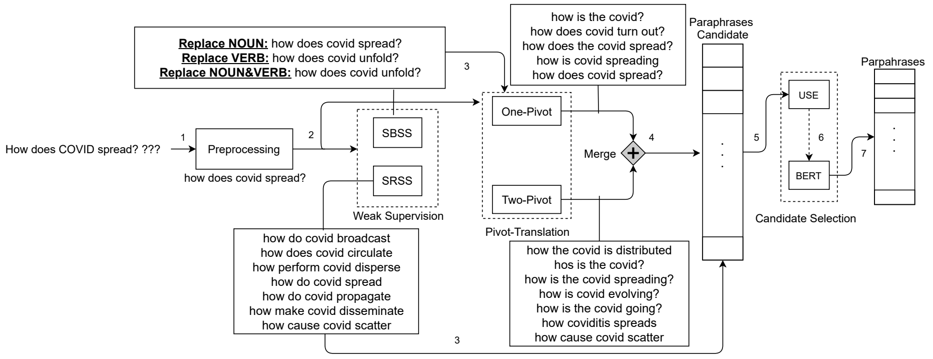


Figure 2: Pipeline at work when configured to use Weak-supervision and Pivot Translation

pipeline chooses pivot languages with grammar unrelated to the target language so as to improve diversity, as suggested by [7].

Text-To-Text Transfer Transformer (T5) [21]. Transformers are a type of neural network architecture developed to perform *Sequence Transduction*, meaning any task that transforms an input sequence to an output sequence (e.g. machine translation, text-to-speech, text summarization). Introduced by Vaswani et al. [24], the idea is to use the *attention mechanism* to eliminate the need for Recurrent Neural Network (RNN), and its known issues, e.g., challenges in handling long-term dependencies and the sequential nature of RNN preventing parallelization. In the pipeline we include a paraphrasing technique based on T5 [21], a transformer implemented by Google to perform sequence transduction. By default T5 does not perform paraphrasing, so we trained it on the Quora Question Pairs dataset [23] and Para-NMT datsets [25] to generate paraphrases, following the work of Goutham [9]. For each given input sentence the T5 model will generate a list of candidate paraphrases.

2.1.2 Candidate selection. While the techniques in the over- generation step aim at exploring different variations of the input sentence, the candidate selection aims at pruning the candidate list to remove paraphrases that do not contribute to the desired paraphrase dataset quality attributes. We mentioned earlier that existing techniques can lead to semantically unrelated, duplicates, or grammatically incorrect paraphrases, so by adding a quality control step in the pipeline, we enable the application, experimentation and evaluation of quality control techniques. The current version of the pipeline provides two techniques that can be combined to filter out semantically irrelevant paraphrases and duplicates: applying cosine similarity between the input sentence and a generated paraphrase using the i) *Universal Sentence Encoder* (USE) [4] embeddings, and ii) *Bidirectional Encoder Representations from Transformers* (BERT) [6] embeddings. Lower and upper bound thresholds can then be applied to the similarity score to discard irrelevant and duplicate paraphrases. Default thresholds are based on prior work [18].

2.1.3 Pipeline reuse and configuration. The pipeline provides a pool of predefined configurations that we have experimentally identified as providing good performance or that mirror combinations of techniques proposed in the literature. An example of such configurations is shown in Figure 2, which i) starts by applying weak supervision to generate an initial list of candidate paraphrases (coupling SBSS and SRSS), ii) applies pivot translation (online MT, and both one- and two- level pivot language) to further expand on each candidate paraphrase, and finally iii) applies candidate selection in two steps by applying first USE and then BERT embeddings to filter out duplicates and semantically irrelevant paraphrases.

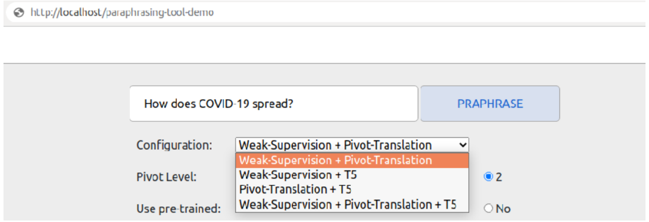
Researchers and practitioners can configure the pipeline by providing command line parameters or defining a configuration file. The current version supports the definition of i) what over-generation and selection techniques to incorporate in the pipeline, and ii) configuration parameters for those techniques. Complex pipelines can be supported by leveraging the programmatic interfaces (Python APIs, described in the project repository) to write the pipeline logic. The community can also contribute with new techniques by extending the current pool of techniques. The code and documentation is available as open source.[4]

2.2 Supporting evaluation and experimentation

The pipeline supports the evaluation and experimentation of existing and novel pipelines by leveraging the configuration and extension features, as well as built-in metrics to assess important quality metrics such as semantic relatedness and diversity [29].

To capture the relevance of the generated paraphrases we incorporate three different metrics. This includes the *Bi-Lingual Evaluation Understudy* (BLEU) [17], a widely adopted metric that measures the similarity between two given sentences. It considers the exact match between the reference sentence and the generated paraphrase by counting overlapping n-grams (n being a parameter). We incorporate *Google's BLEU*, which measures sentence-level similarity by recording first all sub-sequences of 1, 2, 3 or 4 tokens in output and target sequence (n-grams), to then calculate precision and recall based on matching n-grams. The GLEU score is then the minimum of precision and recall. We also include the *Character n-gram F-score* (CHRF) [20], which computes the precision, recall and f-score from the n-gram overlaps, and returns the support which is the true positive score. Then, to assess the diversity of the set of generated paraphrases, we incorporate metrics such as the

[4] https://github.com/AudayBerro/automatedParaphrase



http://localhost/paraphrasing-tool-demo

How does COVID-19 spread?

Configuration: Weak-Supervision + Pivot-Translation

Pivot Level: ● 2

Use pre-trained: ○ No

PRAPHRASE
Weak-Supervision + Pivot-Translation
Weak-Supervision + T5
Pivot-Translation + T5
Weak-Supervision + Pivot-Translation + T5

Figure 3: Web client for testing pipeline configurations

Type-Token Ration (TTR), *Paraphrase In N-gram Changes* (PINC) [5] and *Diversity* (DIV) [12].

The pipeline can then allow researchers and practitioners to run benchmarks of existing techniques, or combination of techniques, by setting up different configurations, as well as running ablation studies by removing components from the pipeline to assess their contribution to the quality of the generated paraphrases. The pipeline is also able to generate intermediate results after the execution of each individual component so as to support evaluation.

3 DEMONSTRATION

The demonstration will showcase two main usage scenarios: the support for paraphrase generation and experimentation.

Paraphrase generation. This demonstration scenario will enable the audience to test and inspect the result of automatic paraphrasing techniques, and in the process learn about the strengths and limitations of current automatic paraphrasing approaches. To facilitate the hands-on testing and analysis, we will offer a web client (see Figure 3) that will allow the audience to i) insert a input sentence, ii) select and tune one of the pre-defined configurations, and iii) inspect the resulting paraphrases. To the interested audience, we will also demonstrate how to define and run their own pipeline by simply creating a configuration file.

Experimentation. This scenario will demonstrate the experimentation features of the pipeline, and in the process highlight the differences between current metrics and their limitations, as well as the trade-off between semantic relatedness and diversity. We will showcase these aspects by walking the audience through an ablation study that will assess the contributions of weak supervision and pivot translation to the default pipeline *"Weak supervision + Pivot translation"* (Figure 2) and see the benefits of combing techniques. This will imply running three separate configurations, and comparing their performance based on the supported metrics described in Section 2.2. This scenario will leverage the command line client to generate the quality assessment metrics, that will then be leveraged for analysis.

REFERENCES

[1] Katrin Affolter, Kurt Stockinger, and Abraham Bernstein. 2019. A comparative survey of recent natural language interfaces for databases. *The VLDB Journal* 28, 5 (2019), 793–819.

[2] Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL'05*. 597–604.

[3] Nicola Castaldo, Florian Daniel, Maristella Matera, and Vittorio Zaccaria. 2019. Conversational data exploration. In *ICWE*. Springer, 490–497.

[4] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[5] David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proc. of ACL: Human Language Technologies*. 190–200.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Christian Federmann, Oussama Elachqar, and Chris Quirk. 2019. Multilingual whispers: Generating paraphrases with translation. In *Proc. of W-NUT*. 17–26.

[8] Atsushi Fujita, Kentaro Furihata, Kentaro Inui, Yuji Matsumoto, and Koichi Takeuchi. 2004. Paraphrasing of Japanese light-verb constructions based on lexical conceptual structure. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*. 9–16.

[9] Ramsri Goutham. 2020. Paraphrase any question with T5. https://github.com/ramsrigouthamg/Paraphrase-any-question-with-T5-Text-To-Text-Transfer-Transformer-

[10] Shaohan Huang, Yu Wu, Furu Wei, and Zhongzhi Luan. 2019. Dictionary-guided editing networks for paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6546–6553.

[11] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in C++. *arXiv preprint arXiv:1804.00344* (2018).

[12] Yiping Kang, Yunqi Zhang, Jonathan K Kummerfeld, Lingjia Tang, and Jason Mars. 2018. Data collection for dialogue system: A startup perspective. In *Proceedings of NAACL-HLT 2018, Volume 3 (Industry Papers)*. 33–40.

[13] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*. ACL, Vancouver, Canada, 67–72.

[14] Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872* (2017).

[15] Nitin Madnani and Bonnie J Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics* 36, 3 (2010), 341–387.

[16] Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the ACL: Volume 1, Long Papers*. 881–893.

[17] 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*. 311–318.

[18] Soham Parikh, Quaizar Vohra, and Mitul Tiwari. 2020. Automated Utterance Generation. *arXiv preprint arXiv:2004.03484* (2020).

[19] Sunghyun Park, Seung-won Hwang, Fuxiang Chen, Jaegul Choo, Jung-Woo Ha, Sunghun Kim, and Jinyeong Yim. 2019. Paraphrase Diversification Using Counterfactual Debiasing. In *Proceedings of the AAAI-19*, Vol. 33. 6883–6891.

[20] Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. 392–395.

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[22] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment.*, Vol. 11. NIH Public Access, 269.

[23] Iyer Shankar, Dandekar Nikhil, and Csernai Kornel. 2017. First Quora Dataset Release: Question Pairs. (2017). https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[25] John Wieting and Kevin Gimpel. 2017. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732* (2017).

[26] Thomas Wolf et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proc. of EMNLP: System Demonstrations*. ACL, Online, 38–45.

[27] Qiongkai Xu, Juyan Zhang, Lizhen Qu, Lexing Xie, and Richard Nock. 2018. D-page: Diverse paraphrase generation. *arXiv preprint arXiv:1808.04364* (2018).

[28] Mohammad-Ali Yaghoub-Zadeh-Fard, Boualem Benatallah, Fabio Casati, Moshe Chai Barukh, and Shayan Zamanirad. 2020. Dynamic word recommendation to obtain diverse crowdsourced paraphrases of user utterances. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 55–66.

[29] Mohammad-Ali Yaghoub-Zadeh-Fard, Boualem Benatallah, Fabio Casati, Moshe Chai Barukh, and Shayan Zamanirad. 2020. User Utterance Acquisition for Training Task-Oriented Bots: A Review of Challenges, Techniques and Opportunities. *IEEE Internet Computing* (2020).

[30] Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 9604–9611.

[31] Shiqi Zhao, Haifeng Wang, Xiang Lan, and Ting Liu. 2010. Leveraging multiple MT engines for paraphrase generation. In *Proc. of Coling 2010*. 1326–1334.