Automated Paraphrase Generation with Over-generation and Pruning Services

Auday Berro[1], Marcos Baez[1], Boualem Benatallah[2,1], Khalid Benabdeslem[1], and Mohammad-Ali Yaghub Zade Fard[2]

[1] Université Claude Bernard Lyon 1, LIRIS UMR5205. Villeurbanne, France
[2] University of New South Wales. Sydney, Australia
{auday.berro, marcos-antonio.baez-gonzalez, khalid.benabdeslem}@univ-lyon1.fr, {m.yaghoubzadehfard, b.benatallah}@unsw.edu.au

Abstract. Conversational services are emerging as a new paradigm for accessing information by simply uttering questions in natural language, posing a whole new set of challenges to the design and engineering of information systems. Training conversational services to deal with the nuances of natural language often requires collecting a high-quality and diverse set of training samples (i.e., paraphrases). Traditional approaches such as hiring an expert or crowdsourcing involve data collection processes that are often costly and time-consuming. Automated paraphrase generation is a promising cost-effective and scalable approach to generating training samples. Current automatic techniques, however, tend to specialise in specific types of lexical or syntactic variations. As a result, generated paraphrases may not perform well in relevant quality aspects such as diversity and semantic relatedness. In this paper, we follow an approach inspired by services integration to address these issues and generate paraphrases in English that are semantically relevant and diverse. We propose an **extensible** and **reusable** pipeline that combines automatic paraphrasing techniques in a two-step process that first focus on i) leveraging the strengths of multiple techniques to generate the most diverse (and possibly noisy) set of paraphrases, to then ii) address common quality issues in a separate step. Through empirical evaluations we show the benefits of the two-step process design and of combining techniques for more balancing relevance and diversity.

1 Introduction

Conversational services such as chatbots and Question/Answering (Q&A) systems are emerging as the new frontier for human-machine natural language interactions [22]. Over the last few years, thousands of domain-specific bots have been used in a variety of significant cases: office tasks, IT, healthcare, sports, e-commerce, education, and e-government services. Users can obtain responses by uttering requests in natural language, e.g., *"which company makes the iPod"* instead of browsing a Website or reading a document. The design and engineering of such services pose a whole new set of challenges [39], now concerned with how to interpret and deliver natural experiences in human language.

This shift in the interaction paradigm introduces crucial gaps in the engineering of conversational services. Especially in rapid deployment situations (e.g., the COVID-19 crisis), fast acquisition of training data is a major roadblock to their fast deployment. Requiring the acquisition of large, high-quality training samples in such situations can lead to chatbots with low-quality comprehension and less natural interaction styles [10]. However, it is essential to have a linguistically diverse utterance set to train such systems on how to interpret different variations of the same user utterance. A user request can be expressed in many different ways. For our previous example, another user may ask "*who manufactures the iPod?*". Failing to correctly identify and process such nuances of natural language (i.e., intent matching) can have a negative impact on the effectiveness of the conversational services and, ultimately, on the user experience [20].

In this context, *paraphrasing* is an important natural language processing task that aims to reformulate a given natural language utterance into its many possible variations to generate additional training data [21]. Relying on experts to provide and annotate utterance paraphrases at scale can be costly, which has motivated research into other *utterance acquisition methods* [38]. These approaches fall into three main methods: i) *bot usage*, referring to those relying on deployed prototypes to collect utterances directly from users, ii) *crowdsourcing*, as those leveraging *crowdsourcing* to collect paraphrases at scale with non experts and iii) *automated approaches*, to those that generate paraphrases systematically. All the approaches involve trade-offs between relevant quality metrics, such as diversity, naturalness, correctness, and operational costs [38].

Automated paraphrasing offers a promising direction to address the challenge of fast acquisition of training paraphrasing sets. As we will see, current techniques focus on introducing specific *lexical* variations (e.g., synonyms substitutions) or *syntactic* variations (structural changes) on the input sentence while still maintaining semantic similarity to the original sentence [28]. Thus, important quality dimensions for assessing these techniques are semantic *relevance* and *diversity* of the resulting paraphrases [38]. While quality is a much involved concept [38], in this work we focus on these dimensions as they dictate to what extent a conversational service will interpret a relevant user request under its plausible expressions. Existing techniques, however, still fall behind in terms of quality, with the literature pointing to models often failing to produce sufficiently diverse and semantically related paraphrases [35, 21].

In this paper, we follow an approach inspired by services integration to address the key challenge of automatically generating paraphrases *in English* that are semantically relevant and diverse. We propose an extensible and reusable pipeline that unifies, integrates and extends various paraphrasing services, enabling the definition of paraphrase generation pipelines. In doing so, the pipeline contributes with the design and evaluation of a two-step process, including: i) paraphrase *candidate over-generation*, leveraging specialised techniques that can be combined to generate a large number of diverse but (potentially) noisy candidate paraphrases, and ii) *candidate selection*, with services that can be incorporated to discard semantically irrelevant paraphrases and duplicates, thus

filtering out low quality paraphrases. The rationale behind decoupling this process in two steps is that we can focus first on generating the most diverse possible set of candidate paraphrases by combining the variations introduced by different specialised paraphrasing services (e.g., the lexical diversity in the weak supervision technique, with the syntactical diversity of T5), to then have a dedicated step addressing the challenge of ensuring the semantic relevance of the outcome. Through an empirical evaluation we show the benefits of our pipeline approach to paraphrase generation, with combinations of paraphrasing services and automatic candidate selection leading to more balanced performance on relevance and diversity metrics. The resulting pipeline framework offers a Web interface, a Python SDK and REST APIs, that pushes paraphrasing as a service.

2 Problem Statement

We frame the problem in the context of fast acquisition of utterance paraphrase sets for training the ability of conversational AI systems to interpret natural language user requests (i.e., intent recognition task). Given an input utterance x, we can define paraphrasing as the problem of generating a set of k utterance paraphrases $Y = \{y_1, y_2, ..., y_k\}$ so that each $y \in Y$ is generated by introducing *variations* of x while keeping the same meaning [3]. Thus, the goal is to produce a *diverse* set Y while preserving *semantic* equivalence to x.

Broadly speaking, automatic paraphrasing techniques rely on approaches that aim at introducing *lexical* and *syntactic* variations. The quality of these techniques is commonly measured[1] in terms of the semantic *relevance*, denoting the extent to which the output paraphrases are similar in meaning to the input utterance, and *diversity*, as the breath and variety of paraphrases in the resulting corpus [38]. The literature on automatic paraphrasing (see Section 6) has seen the development of a myriad of specialised techniques, but that still struggle in addressing and balancing these important quality aspects [35, 21]. For example, the diversity of a technique might be limited by the types of variations it specialises for (e.g., only lexical), and the relevance by the noise introduced in the generation process (e.g., semantically irrelevant paraphrases).

In this paper we explore, through design and empirical evaluations, an approach to paraphrase generation that aims at addressing the above limitations. The proposed automated paraphrase generation pipeline (see next section) contributes with the design and evaluation of the following key design decisions:

- Reusing and combining existing paraphrasing techniques so as to benefit from the diversity of variations in the state of the art
- Turning the generation in a two-step process that incorporates automated quality control, so as to address quality issues in automatic techniques.

[1] Quality aspects such as fluidity, grammatical correctness, and other dimensions explored especially in the context of crowdsourcing [38], are not addressed in this work

3 Automated Paraphrasing Pipeline

In our approach we see existing techniques as *services* that provide the building blocks for defining paraphrase generation data-flow pipelines. The idea is that by combining services we can leverage the variations introduced by specialised techniques and produce better results. As seen in Figure 1, the paraphrasing pipeline defines a two-step process that 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 services that both expand on paraphrase candidates while also pruning low quality ones from the final list.

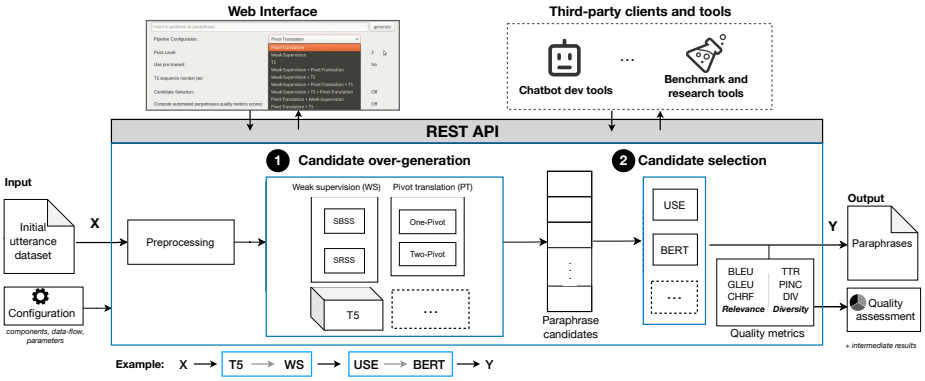

Fig. 1: Automated Paraphrase Generation Pipeline Architecture

The proposed framework supports *handcrafted* pipelines, i.e., the definition of data-flow pipelines as combinations of services. An expert can design the pipeline by selecting the services to be combined, their configuration parameters and the specific data-flow these services will describe. These complex pipelines are supported by leveraging a programmatic interface in Python, and can be enacted from a command-line client, a Web interface and REST API. To support developers, we also provide a ready-to-use pool of predefined pipelines that mirror combinations of techniques proposed in the literature. To support researchers, the pipeline comes with built-in automatic metrics (see next section) that facilitate benchmarks and ablation studies. The community can also contribute with new over-generation and candidate selection techniques by extending the current pool of services. The code and documentation is available as open source.[1] In the following the introduce the two main steps in the paraphrase generation pipeline and the type of services supported.

3.1 Candidate over-generation services

Candidate over-generation refers to the use of services that can be combined to expand on the input sentence to incrementally generate a larger and more

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

diverse set of paraphrase candidates. The services we currently support were implemented by taking existing techniques and models, extending them to offer higher flexibility, as well as offer sensible defaults based on experimentation.

Weak supervision. It is a learning approach that automatically creates its own training data through the use of noisy data [6, 24]. We rely on weak supervision to generate candidate paraphrases from the input utterances by replacing individual words with their synonyms. To do so, we begin by performing part-of-speech (POS) tagging to identify tokens (verbs and nouns) to be replaced using *SpaCy* [11], to then select relevant synonyms from *NLTK-Wordnet*.[1] Unlike previous work [20], we adopted two complementary strategies, discussed next, for synonym selection and replacement so as to balance relatedness of the generated candidates and exploration of diverse paraphrases.

Select Best Synonym Sentence (SBSS): This strategy generates the best possible candidate paraphrases by selecting variants with the highest semantic relatedness. To do so, the paraphrase candidate is generated by replacing each selected token with the WordNet synonym that has the highest cosine similarity respecting a predefined interval threshold $[\alpha, \beta]$. Let τ be the selected token, $\mathbb{S} = \{s_1, s_2, ..., s_N\}$ its list of WordNet synonyms and ψ the selected synonym. $\forall s \in \mathbb{S} : \psi = argmax[\cos(\vec{\tau}, \vec{s})]$ and $\alpha \leq \psi \leq \beta$, where $\vec{\tau}, \vec{s}$ are the *USE* sentence embeddings using τ and s respectively. This will generate three candidate paraphrases for each sentence, one by replacing all the tokens marked VERB, one by replacing all the tokens marked NOUN and the last by replacing all the tokens marked as VERB and NOUN at the same time.

Semantically Relevant Synonym Sentences (SRSS): This strategy follows a more exploratory approach, by relaxing the selection to include *all* synonyms above the threshold α. To do so, following the *POS tagging* phase, each selected token is replaced by the Wordnet synonyms that have a cosine similarity greater than a threshold α. Let τ be the selected token, $\mathbb{S} = \{s_1, s_2, ..., s_N\}$ its list of Wordnet synonyms. $\forall s \in \mathbb{S} : if \cos(\vec{\tau}, \vec{s}) \geq \alpha \Rightarrow$ generate a candidate by replacing τ with s. For each sentence, three different lists of paraphrases will be generated, one by replacing the token marked VERB, the other by replacing the token marked NOUN and the last by replacing the token marked VERB or NOUN.

Pivot translation. The intuition behind pivot translation is that two sentences that have the same foreign translation can be assumed to have the same meaning. Thus, paraphrases can be obtained by translating a sentence in source language S into a foreign language F and then back-translating it into S. In this component, we leverage multiple pivot languages and multiple translation engines to generate more candidate paraphrases per input sentence. Below we elaborate on two important dimensions of pivot translation:

Paraphrase system: A paraphrase system can be defined as a triple (MT_i, PL, MT_j) where a Machine Translation Engine MT_i translates a source sentence S into a pivot language PL and then Machine Translation Engine MT_j translates the result back into S, thus generating the paraphrase [40]. When one language is

[1] NLTK: https://www.nltk.org/ and Wordnet: https://wordnet.princeton.edu/

used as pivot, it is called a *single-pivot* paraphrase system, and a *multi-pivot* paraphrase when it is made up of a set of single-pivot systems, each generating one candidate paraphrase. In practical terms, it is preferable to have different MTs in order to maximise the chances of getting more diverse paraphrase options [40], since each engine has its own architecture and was trained differently. In this service, we adopted a multi-pivot system as default. In terms of implementation, the pivot translation service supports online NMT services,[1] such as Google Translate, Deepl and MyMemory. The pipeline is also shipped with pre-trained NMTs like the Huggingface Marian Machine Translator [13, 32]. The type of machine translator is a parameter of the pivot translation service.

Pivot-language level & selection: We informed the pivot selection on the work by Zhao et al. [40], but observed in our trial runs that languages with similar grammatical structure would lead to paraphrases very similar to the source sentence, thus hurting diversity. We thus selected as sensible defaults pivot languages that are not close to the source, i.e., given the source language in English, the system selects pivot languages such as Chinese and Arabic, instead of French and Spanish. Our observation aligns with the recent work by [8] recommending the pivot languages with unrelated grammar so as to improve diversity.

This service also supports different pivot-language levels, i.e., the number of intermediate pivot languages chained to generate the paraphrases. The pipeline can be set to work with a i) *single-level* pivot, including one intermediate language (e.g., *English*→*Italian*→*English*), and a ii) *two-level* pivot, with two intermediate pivot languages (e.g., *English*→*Arabic*→*German*→*English*).

Language-based models (T5). 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 summarization). Introduced by Vaswani et al. [29], the idea is to use the *attention mechanism* to eliminate the need for Recurrent Neural Networks (RNN), and their known issues, e.g., challenges in handling long-term dependencies and the sequential nature of RNN preventing parallelisation. We include a paraphrasing service based on T5 [23], a transformer implemented by Google to perform sequence transduction. By default T5 does not perform paraphrasing, so we fine-tuned it on the Quora Question Pairs dataset [25] and Para-NMT datsets [31] to generate paraphrases, following the work of Goutham [2]. For each given input sentence the T5 model will generate a list of candidate paraphrases.

3.2 Candidate selection services

The use of automatic paraphrasing techniques and the emphasis on diversity in the over-generation phase can lead to potential quality issues that must be addressed. We mentioned that generated paraphrases can be semantically different from the input phrase (e.g., selecting wrong synonyms for the context), and

[1] Available at https://translate.google.com/, https://www.deepl.com/translator and https://mymemory.translated.net/

[2] Paraphrase any question with T5 (2020), https://git.io/JEYQM

duplicated paraphrases formed (e.g., techniques generating to very similar paraphrases). Hence, given a pool of noisy candidate paraphrases at this stage, the objective of the candidate selection services is to address specific issues to ensure higher quality outcomes by removing irrelevant and duplicates paraphrases. We currently support services adapted from Parikh et al. [20] that perform filtering of semantically unrelated paraphrases and de-duplication.

Let v be a vector representation of the initial utterance sentence and P its set of N-paraphrases $P = \{p_1, p_2, ..., p_N\}$. To discard irrelevant and duplicate paraphrase, we first obtain the embedding representation of v and $\forall p \in P$, and then compute the semantic similarity between the v embedding and each paraphrases embedding. We use cosine similarity for the semantic similarity, with values ranging from -1 (exact opposite) to 1 (identical) with intermediate values indicating the degree of (dis)similarity. On the cosine similarity score, we define a lower and upper thresholds for selecting semantically relevant paraphrases, borrowing the values defined by [20]. The candidate selection services then perform the following:

Filtering out irrelevant paraphrases. Semantically irrelevant candidates are discarded evaluating the cosine similarity between the vector representations of the input utterance and each candidate paraphrase. We first compute the cosine similarity of the USE [7] embeddings and, in a second pass, using the cosine similarity of the BERT [4] embeddings. If the cosine score is below 0.5, for any of the two embedding models, we consider the candidate paraphrase not to be semantically related and it is filtered out. The reason for using two different models is that some semantically irrelevant candidates are not identified when filtering with USE or BERT. As we confirmed experimentally, a combination of both models achieves better performance.

Filtering out duplicates. Duplicate paraphrases are discarded using cosine similarity between the vector representation of the input utterance and each generated paraphrase using BERT embeddings. If the cosine score is above 0.95, we consider the candidate paraphrase to be a duplicate and it is filtered out.

We should note that to make BERT work with sentence embeddings, we tested various *pooling strategies* [34] but observed that the concatenation of the last four layers of each token embedding vector to be the most suitable for the semantic similarity task. USE already supports sentence embeddings.

4 Experimental Setup

The goal of the evaluation is to assess our approach that considers automatic paraphrase generation as a pipeline that combines specialised services in a two-step process. In this section, we describe the experimental setup for how we: (a) Investigate whether there are gains in terms of *relevance* and *diversity* of resulting paraphrases when organising the generation process in over-generation and candidate selection steps; (b) Explore the benefits of combining existing paraphrase generation techniques for *relevance* and *diversity*.

Dataset. We run our experiments on two relevant datasets. We used the *GraphQuestions* dataset [27], a benchmark paraphrasing corpora for Q&A that contains 5,166 pairs of crowdsourced paraphrases questions with their answers in English. We chose this dataset as it is representative of the type of source sentences for our paraphrasing task. For our experiments, we selected a random sample of 237 questions. We also selected the *WebQuestions* dataset [2], a Q&A dataset that uses Freebase as the knowledge base. This dataset was created by crawling questions through the Google Suggest API to then crowdsource answers on Amazon Mechanical Turk. In our experiment, we use the *devtest* dataset, containing 189 questions. Notice that we only use the *questions* for paraphrasing.

Experimental procedure. To test the impact of our approach, we selected configurations of the pipeline based on two dimensions: (i) *process design*, with over-generation only (OG) and over-generation with candidate selection (CS) as alternatives, and (ii) *service combination*, with individual and combined services as alternatives. We used as baseline services those reported in Section 3.[1]

To assess the impact of the process design, we first run pipeline configurations with the individual services: weak supervision (WS), pivot translation (PT) and T5, and for each, we generated paraphrases with the two process design alternatives (OG, CS). We leveraged the evaluation metrics (presented below) to assess the impact of candidate selection on the resulting paraphrases. The results from these metrics were complemented with qualitative observations of the generated paraphrases of each configuration, for a small random sample of 20 sentences.[2]

Next, to assess the benefits of combining automatic paraphrasing techniques, we run the pipelines configurations that combined the services and compared them to the individual services. We created the sequences WS $\rightarrow$ PT and WS $\rightarrow$ T5 to combine observed properties of the underlying services. These pipelines used the same configurations for the underlying services as the individual service pipelines. The resulting paraphrased were evaluated using our reference metrics.

Evaluation metrics. The pipeline configurations were evaluated using automatic evaluation metrics commonly used in assessing paraphrase quality [37]

To capture the **relevance** of the generated paraphrases to the input utterance, we use two different metrics. This includes the *Bi-Lingual Evaluation Understudy* (BLEU) [19], 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. In our tests we consider $n = 2, 3, 4$. We also incorporate *Google's BLEU* (GLEU) [33], which measures sentence-level similarity by recording first all sub-sequences of 1, 2, 3 and 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. For these metrics, the score for a list of resulting paraphrases is computed as the average of the individual sentence scores.

[1] Services configured with their default values, listed here https://bit.ly/3fHFNgB

[2] Notice that the goal of the qualitative observation was to characterise the limitations and strengths of the techniques and not to provide a full human evaluation.

We assess the **diversity** of the generated paraphrases with n-grams metrics that capture diversity at *corpus* level, i.e., of all the candidate paraphrases for a reference sentence, and at *sentence* level, i.e., between a single candidate and the reference sentence. The *Type-Token Ration* (TTR) calculates lexical diversity at corpus level, as the rate of unique words in a candidate paraphrase to the total number of words in the candidates set. Then, *Paraphrase In N-gram Changes* (PINC) [5], computes diversity at sentence level as the percentage of n-grams that appear in the candidate sentence but not in the reference sentence. The PINC score for the candidate paraphrase set is computed as the mean of the sentence scores. *Diversity* (DIV) [14] computes diversity at corpus level by calculating n-grams changes between all the pairs in the candidate paraphrases set, rewarding the unique n-grams between each two candidates pairs. In our evaluation, the score for each experimental condition is the mean of the metric scores of all reference sentences in the given dataset.

5 Results

5.1 Impact of two-step process design

The performance of the baseline configuration pipelines for process designs with and without candidate selection is illustrated in Table 1. To properly dissect the impact of candidate selection, we start by separately analysing the impact of filtering out duplicates and semantically irrelevant paraphrases.

Table 1: Performance of over-generation services for a process design with over-generation only (OG), and over-generation and candidate selection services (CS), after removing irrelevant paraphrases ($\star$) and removing duplicates ($\dagger$)

Metric	**GraphQuestions**						**WebQuestions**					
	WS		**PT**		**T5**		**WS**		**PT**		**T5**	
Relevance	OG	CS*	OG	CS*	OG	CS*	OG	CS*	OG	CS*	OG	CS*
$BLEU_2$	0.494	0.497	0.451	0.511	0.403	0.407	0.572	0.577	0.350	0.446	0.406	0.411
$BLEU_3$	0.377	0.380	0.368	0.416	0.319	0.323	0.487	0.491	0.292	0.370	0.319	0.322
GLEU	0.409	0.412	0.389	0.444	0.338	0.342	0.474	0.479	0.275	0.365	0.320	0.324
Diversity	OG	CS†	OG	CS†	OG	CS†	OG	CS†	OG	CS†	OG	CS†
TTR	0.223	0.233	0.312	0.589	0.281	0.422	0.255	0.307	0.314	0.421	0.304	0.426
PINC	0.539	0.546	0.568	0.771	0.587	0.653	0.469	0.478	0.684	0.845	0.642	0.718
DIV	0.611	0.614	0.733	0.724	0.732	0.704	0.532	0.552	0.830	0.849	0.775	0.770

As seen in the table, for all over-generation services in both datasets, removing *irrelevant paraphrases* contributes to higher scores in the BLEU and GLEU metrics (CS*), indicating more relevant paraphrases as a result. Similarly, removing *duplicates* (CS†) has the effect of higher diversity, in terms of more diverse vocabulary in the resulting paraphrase corpus (TTR), as well as when comparing the generated paraphrases at a sentence level (PINC). However, this does not affect the overall (lexical and syntactical) diversity at the corpus level (DIV).

	Weak Supervision	Pivot Translation	T5
Duplicate	which company make the ipod	What company makes iPod? Which company makes the iPod?	Which company makes iPod? What company makes the iPod?
Relevant	which company **produce** the ipod ? which company **build** the ipod which company **develop** the ipod	Which company **manufactures** ipods? **What** company **manufactures** the ipod? Which company **does** the ipod? **What** kind of company **does** an iPod?	Which company ***sells*** the iPod? Which company makes the iPod Touch? ***Who makes iPod***, and what brand is it from? **What** are the famous companies that make iPods?
Irrelevant	which company **attain** the ipod which **companionship establish** the ipod	What company does the iPod **do**? What company does the iPod **make**?	Which company makes **iPhones**? Which company creates **iPad**?

Fig. 2: Example paraphrases generated for the input sentence "which company makes the ipod?", highlighting type of variations introduced.

A close inspection of the generated paraphrases, and those filtered out, gave us insights into the strengths and limitations of the candidate selection services. The duplicate filtering is effective in removing paraphrases that result from simple lexical permutations, contractions, switching plural and singular, adding and removing articles, simple wh-question substitution, single synonym substitution in long sentences, among other basic variations. In turn, the filtering of semantically irrelevant paraphrases is good at removing those that result from significant variations of the input sentence (e.g., "What is the reason that 9/11 attacks occurred?" as a paraphrase for "find terrorist organizations involved in September 11 attacks") but less effective in identifying semantic differences resulting from subtle changes, such as replacing a word with the wrong synonym. This limitation also includes cases where important entities and concepts are replaced by synonyms (e.g., "who wrote twilight[name of book]?" as "who wrote dusk").

The above tells us that current techniques indeed suffer from quality issues, and that by designing a process that ensures candidate selection we can have higher quality paraphrases. However, there is still room for improving and developing better candidate selection services.

5.2 Characterising over-generation services

The results in Table 2 helps us draw comparisons between the performance of the over-generation services (WS, PT, T5) after candidate selection[1]. For both datasets, we can see WS leading with higher scores for the relevance metrics (BLEU, GLEU) compared to PT and T5. This can be attributed to the word-level substitutions performed by WS, which introduce variations that are still close to the input sentence. For the same reason, this service can only provide lexical diversity, limiting the diversity and the characteristic of the resulting paraphrases (see Figure 2). In terms of the type of mistakes introduced by WS, we observed the selection of wrong synonyms (due to the lack of sentence-level context) as the main reason leading to irrelevant paraphrases.

On the other hand, PT scored the lowest on the relevance metrics and on all but one of the diversity metrics (TTR). The higher score on TTR (only) tells

[1] For a qualitative comparison of the paraphrases generated by the various techniques, refer to our Appendix at https://bit.ly/3go11zU

Table 2: Performance of pipelines featuring individual and combined over-generation services. Bold values denote best result compared to individual services, and italics second best. Gray denotes best result among individual services.

Metric	GraphQuestions					WebQuestions				
	WS	**PT**	**T5**	**WS-PT**	**WS-T5**	**WS**	**PT**	**T5**	**WS-PT**	**WS-T5**
$BLEU_2$	0.490	0.275	0.294	*0.356*	*0.458*	0.580	0.216	0.267	*0.372*	*0.446*
$BLEU_3$	0.372	0.210	0.227	*0.263*	*0.344*	0.493	0.181	0.209	*0.309*	*0.369*
GLEU	0.405	0.224	0.235	*0.282*	*0.374*	0.482	0.150	0.190	*0.286*	*0.356*
TTR	0.233	0.488	0.423	*0.308*	*0.248*	0.309	0.479	0.428	*0.329*	*0.323*
PINC	0.541	0.525	0.650	**0.676**	*0.576*	0.471	0.612	0.713	**0.680**	*0.616*
DIV	0.612	0.448	0.697	**0.789**	*0.656*	0.547	0.481	0.759	**0.783**	*0.722*

us that PT can lead to a richer vocabulary but an overall lower diversity at a corpus level. However, our observations of the resulting paraphrases showed that it can offer not only lexical but also syntactic diversity by introducing grammatical variations in the sentences. Among the limitations, we observed a higher percentage of duplicate paraphrases compared to the other services, due to the back-translation process generating paraphrases very similar to the original sentence for some language pairs. We also observed substitution of wrong synonyms and the meaning of questions getting lost in the translation process.

T5 shows a solid performance, coming second in terms of relevance metrics but featuring the highest sentence and corpus level diversity scores. A close inspection of the resulting paraphrases revealed the different ways T5 contributes to diversity (see Figure 2 for illustrative examples). It introduces lexical diversity by replacing words with synonyms, although these tend to be fewer but context-aware and therefore significantly less noisy than WS. We also observed the richest syntactic diversity in terms of grammatical changes, summarisation of sentences (e.g., "Who makes iPod"), generalisation and extrapolation ("..and what brand is it from?"), and adding details (e.g., "iPod" with "iPod Touch"). In terms of frequent types of mistakes, the higher diversity introduced candidate sentences that, while on the same topic, are semantically different from the original.

5.3 Combining over-generation services

The comparison of pipeline configurations featuring individual and combined over-generation services is shown in Table 2. For both datasets, we can see that the configurations with combined services yield the most balanced performances, improving on the weaknesses of their individual services while achieving results comparable to the best performing one. In the case of WS $\rightarrow$ PT, this resulted into paraphrases that showed improved scores in relevance metrics (BLEU, GLEU) compared to PT and on diversity metrics compared to WS and even PT (PINC, DIV). We can observe a similar trend with WS $\rightarrow$ T5.

We should note that this balanced performance was obtained with a simple combination of the over-generation services, without optimising the parameters

to better combine the characteristics of each service. Tuning parameters to better leverage synergies could result in better performances.

6 Related Work

Crowdsourcing is a widely used approach to paraphrase generation [30]. In a crowdsourced process, an initial utterance, usually provided by an expert or generated using generative models or grammars [26, 30], is presented as a starting point, and workers are asked to paraphrase the expression to new variations. It is a popular strategy as it can help scale the paraphrases generation efforts while reducing the costs, compared to hiring experts [15]. However, the generated paraphrases may suffer from various quality problems (e.g., cheating, semantic errors, spelling and linguistic errors, task misunderstanding) [36]. Thus, quality control in this context is an important step, typically requiring quality control tasks run with the crowd or involving experts. The costs of running such a crowdsourcing process can still be significant, depending on the configuration of the process and the task design [38].

Automated Paraphrases Generation. The literature on automated paraphrases generation covers a wide range of approaches, including probabilistic, hand-written rules and formal grammar models [9], data-driven techniques [17], machine translation techniques [12, 18], and recently approaches that take advantage of contextual representations models a.k.a embeddings, BERT [7] and USE [4]. Here we provide an overview of the most prominent approaches.

Recent work has focused on approaches based on Machine Translation (MT) techniques. This includes the Rule-based Machine Translation (RBMT), Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) [18, 12]. SMT relies on statistical analysis of bilingual text corpora to generate paraphrases. It treats translation as a machine learning problem, applying a learning algorithm to a large parallel corpus, parallel text or bitext so that the learner is then able to translate previously unseen sentences [16]. NMT is another prominent MT approach. In its conventional form, the so called encoder–decoder approach, it encodes a whole input sentence into a fixed-length vector from which a translation will be decoded [1], enabling a sentence to be paraphrased into new variations [38]. In this work, we take these existing automatic paraphrasing techniques as the foundation, adopting three prominent techniques to conceptualise, develop and evaluate a pipeline approach to automatic paraphrase generation.

The closest to the approach presented in this paper is the work by Parikh et al. [20]. They proposed an ensemble of techniques and automatic filtering algorithms in the context of the generation of question utterances from documents. Their approach takes a document, applies extractive summarisation to identify key sentences to then apply automatic paraphrasing. For the paraphrasing, they combine the *output* of four over-generation techniques *running in parallel* that were selected for their problem so as to produce *larger number* of candidate paraphrases. They then propose a novel candidate selection algorithm that assesses the semantic relatedness of each resulting paraphrase to the source sen-

tence by computing the cosine similarity between the vector representations of the sentences (USE and BERT). While this approach is very valuable and informs our approach, we differ and contribute in distinct ways. (i) We propose a framework that supports the definition, enactment and evaluation of automatic paraphrasing pipelines, whereas [20] leverage a specific configuration of techniques applicable to a specific problem and system. (ii) We provide and support an extensible and configurable pool of services, instead of a static set of techniques. We do adopt two general techniques (WS and PT), also present among the four in [20], but implemented them with higher configurability. We propose synonym and replacement strategies for WS, and support different paraphrasing systems, pivot language level and selection for PT. Unlike [20], we also include a language model based technique (T5). (iii) We add a layer of composition on top of a pool of available techniques. The combination of techniques allows developers and researchers to *chain* or *merge* the outcomes of techniques so maximise diversity by leveraging the variations introduced by specialised techniques – thus not limited to a specific configuration or set of techniques. A separate quality control step, while currently based on the algorithms by Parikh et al. [20], is designed to incorporate a broader set of candidate selection services. (iv) In addition to these design contributions, we also offer empirical evidence supporting these design decisions, and a framework for the exploration, development and evaluation of paraphrasing pipelines and services.

Thus, our proposed framework conceptualises the automatic paraphrase generation process in a two step, adds service composition on top of an evolving pool of services, and supports the definition, enactment and evaluation of automatic paraphrasing pipelines.

7 Discussion & Concluding Remarks

In this paper we proposed a data-flow pipeline that unifies, integrates and extends various paraphrasing services, in a two-step process. The experiments provided empirical evidence in support for the pipeline design. The two-step process enables us to first focus on leveraging the good properties of over-generation techniques to generate the most diverse set of paraphrases – even if, as we have seen, they might provide noisy output. Thus, in considering candidate selection as a whole separate problem, we are able to redirect the efforts towards solving specific quality issues, such as duplicates and semantically irrelevant paraphrases. We showed that this approach can indeed increase the relevance and diversity of the outcomes. However, we also pointed out limitations in, among others, detecting semantic changes from subtle variations. This calls for a deeper investigation into specific issues arising from automatic paraphrase generation and development of more effective candidate selection techniques to address them.

Combining over-generation services was successful in producing more balanced results. We have seen that individual techniques have different strengths, introducing distinct types of variations. We observed that combining over-generation services could lead to paraphrases with a better balance of relevance and diversity

compared to using individual services. These observations were obtained even without optimising the pipelines to create better synergies between techniques.

As part of our ongoing efforts, we are integrating more over-generation and selection services, experimenting with novel pipelines, and exploring the integration of crowdsourcing for candidate generation and selection.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of EMNLP. pp. 1533–1544 (2013)
3. Cao, Y., Wan, X.: Divgan: Towards diverse paraphrase generation via diversified generative adversarial network. In: EMNLP. pp. 2411–2421 (2020)
4. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
5. Chen, D., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: Proceedings of HLT. pp. 190–200 (2011)
6. Dehghani, M., Zamani, H., Severyn, A., Kamps, J., Croft, W.B.: Neural ranking models with weak supervision. In: Proceedings of ACM SIGIR. pp. 65–74 (2017)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
8. Federmann, C., Elachqar, O., Quirk, C.: Multilingual whispers: Generating paraphrases with translation. In: Proceedings of W-NUT. pp. 17–26 (2019)
9. Fujita, A., Furihata, K., Inui, K., Matsumoto, Y., Takeuchi, K.: Paraphrasing of japanese light-verb constructions based on lexical conceptual structure. In: Proceedings of MWE: Integrating Processing. pp. 9–16 (2004)
10. Hoehn, S., Bongard, K.: Heuristic evaluation of covid-19 chatbots. Proceedings of CONVERSATIONS 2020 (2020)
11. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020). https://doi.org/10.5281/zenodo.1212303
12. Huang, S., Wu, Y., Wei, F., Luan, Z.: Dictionary-guided editing networks for paraphrase generation. In: Proc. AAAI. vol. 33, pp. 6546–6553 (2019)
13. Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., et al.: Marian: Fast neural machine translation in c++. arXiv preprint arXiv:1804.00344 (2018)
14. Kang, Y., Zhang, Y., Kummerfeld, J.K., Tang, L., Mars, J.: Data collection for dialogue system: A startup perspective. In: Proc. HLT, Vol 3. pp. 33–40 (2018)
15. Lee, W., Huang, C.H., Chang, C.W., Wu, M.K.D., Chuang, K.T., Yang, P.A., Hsieh, C.C.: Effective quality assurance for data labels through crowdsourcing and domain expert collaboration. In: EDBT. pp. 646–649 (2018)
16. Lopez, A.: Statistical machine translation. CSUR **40**(3), 1–49 (2008)
17. Madnani, N., Dorr, B.J.: Generating phrasal and sentential paraphrases: A survey of data-driven methods. Computational Linguistics **36**(3), 341–387 (2010)
18. Mallinson, J., Sennrich, R., Lapata, M.: Paraphrasing revisited with neural machine translation. In: Proc. EACL: Volume 1, Long Papers. pp. 881–893 (2017)
19. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of ACL. pp. 311–318 (2002)

20. Parikh, S., Vohra, Q., Tiwari, M.: Automated utterance generation. arXiv preprint arXiv:2004.03484 (2020)
21. Park, S., Hwang, S.w., Chen, F., Choo, J., Ha, J.W., et. al: Paraphrase diversification using counterfactual debiasing. In: AAAI. vol. 33, pp. 6883–6891 (2019)
22. Piccolo, L.S., Mensio, M., Alani, H.: Chasing the chatbots. In: International Conference on Internet Science. pp. 157–169. Springer (2018)
23. Raffel, C., Shazeer, N., Roberts, A., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
24. Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S., Ré, C.: Snorkel: Rapid training data creation with weak supervision. In: VLDB. vol. 11, p. 269 (2017)
25. Shankar, I., Nikhil, D., Kornel, C.: First quora dataset release: Question pairs (2017), https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs
26. Su, Y., Awadallah, A.H., Khabsa, M., Pantel, P., Gamon, M., Encarnacion, M.: Building natural language interfaces to web apis. In: CIKM. pp. 177–186 (2017)
27. Su, Y., Sun, H., Sadler, B., Srivatsa, M., Gür, I., Yan, Z., Yan, X.: On generating characteristic-rich question sets for qa evaluation. In: EMNLP. pp. 562–572 (2016)
28. Thompson, B., Post, M.: Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. arXiv preprint arXiv:2008.04935 (2020)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
30. Wang, W.Y., Bohus, D., Kamar, E., Horvitz, E.: Crowdsourcing the acquisition of natural language corpora: Methods and observations. In: 2012 IEEE Spoken Language Technology Workshop (SLT). pp. 73–78. IEEE (2012)
31. Wieting, J., Gimpel, K.: Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. arXiv:1711.05732 (2017)
32. Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S., et al.: Transformers: State-of-the-art natural language processing. In: EMNLP 2020 System Demonstration. pp. 38–45 (2020)
33. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
34. Xiao, H.: bert-as-service. https://github.com/hanxiao/bert-as-service (2018)
35. Xu, Q., Zhang, J., Qu, L., Xie, L., Nock, R.: D-page: Diverse paraphrase generation. arXiv preprint arXiv:1808.04364 (2018)
36. Yaghoub-Zadeh-Fard, M.A., Benatallah, B., Barukh, M.C., Zamanirad, S.: A study of incorrect paraphrases in crowdsourced user utterances. In: Proceedings of NAACL-HLT, Vol 1. pp. 295–306 (2019)
37. Yaghoub-Zadeh-Fard, M.A., Benatallah, B., Casati, F., Barukh, M.C., Zamanirad, S.: Dynamic word recommendation to obtain diverse crowdsourced paraphrases of user utterances. In: Proc. of IUI. pp. 55–66 (2020)
38. Yaghoub-Zadeh-Fard, M.A., Benatallah, B., Casati, F., Barukh, M.C., Zamanirad, S.: User utterance acquisition for training task-oriented bots: A review of challenges, techniques and opportunities. IEEE Internet Computing (2020)
39. Yang, Q., Steinfeld, A., Rosé, C., Zimmerman, J.: Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In: CHI. pp. 1–13 (2020)
40. Zhao, S., Wang, H., Lan, X., Liu, T.: Leveraging multiple mt engines for paraphrase generation. In: Proceedings of Coling. pp. 1326–1334 (2010)