Automated Utterance Generation

Soham Parikh,[1*] **Quaizar Vohra,**[2] **Mitul Tiwari**[2]
[1]University of Pennsylvania, Philadelphia, PA USA
[2]Passage AI, Mountain View, CA USA
[1]sohamp@seas.upenn.edu, [2]{quaizar, mitul}@passage.ai

Abstract

Conversational AI assistants are becoming popular and question-answering is an important part of any conversational assistant. Using relevant utterances as features in question-answering has shown to improve both the precision and recall for retrieving the right answer by a conversational assistant. Hence, utterance generation has become an important problem with the goal of generating relevant utterances (sentences or phrases) from a knowledge base article that consists of a title and a description. However, generating good utterances usually requires a lot of manual effort, creating the need for an automated utterance generation. In this paper, we propose an utterance generation system which 1) uses extractive summarization to extract important sentences from the description, 2) uses multiple paraphrasing techniques to generate a diverse set of paraphrases of the title and summary sentences, and 3) selects good candidate paraphrases with the help of a novel candidate selection algorithm.

1 Introduction

Utterance generation is an important problem in Question-Answering, Information Retrieval, and Conversational AI Assistants. Voice assistants like Alexa, Google Assistant, Apple Siri, and Microsoft Cortana are proliferating and now billions of devices have these voice assistants. Any conversational skill developed for these devices needs to understand various ways an end user is asking a question, and be able to respond accurately. While voice assistants are becoming very common, chatbots and conversational interfaces are getting adopted for various conversational automation use cases such as website assistants, customer service automation and IT and enterprise service automation. Question-answering is an important part of any conversational automation use case. It is critical that a conversational assistant understands various ways that could be used in asking the same question, essentially paraphrases of the later. Using relevant utterances as features in a question-answering system has shown to improve the accuracy both in terms of precision and recall to retrieve the right answer.

In this paper we address the problem of utterance generation in the context of conversational virtual assistant and question-answering. In case of question-answering, we generate utterances for questions (for example, Frequently Asked Questions or FAQs) so that we can identify the right answer for the corresponding question even if the question is asked in many different ways. We propose an ensemble method for the utterance generation problem with a novel candidate selection algorithm. Our method first uses extractive summarization to extract important sentences from the description. Second, we use multiple paraphrase generation techniques to generate a diverse set of paraphrases of the title and summary sentences. Some of the techniques we use are full sentence backtranslation, noun/verb phrase backtranslation using constituency parsing, synonym replacement and phrase replacement for paraphrase generation. We use an ensemble method to combine all these different paraphrasing techniques. Finally, we use a novel candidate selection algorithm that utilizes a combination of filtering and de-duplication techniques to select a good set of utterances by leveraging some of the latest contextual embedding techniques such as BERT (Devlin et al. 2019) to find semantically similar utterances and to filter out unrelated utterances and remove duplicate utterances. Our experimental results demonstrate that our ensemble method performs well. The main contributions of this paper are as follows.

- First, we propose an ensemble method for the utterance generation problem, which combines many approaches to utterance generation, and is scalable to add new approaches.
- Second, a novel candidate selection algorithm that uses a combination of filtering and de-duplication techniques to select generated utterances.
- Third, we adopt recent advances in large scale pre-trained contextual embeddings like BERT and Universal Sentence Encoding for candidate selection to find good utterances.
- Finally, we demonstrate through extensive experiments that the proposed techniques work well for the utterance generation problem.

The rest of the paper is organized as follows. Section 2

[*]Work was done during an internship at PassageAI

describes related work and gives contexts around our work. Section 3 defines the problem of utterance generation addressed in this paper. Section 4 discusses the proposed solution and algorithms for the utterance generation problem. Section 5 describes the experimental setup and discusses the results. Finally, we conclude in Section 6.

2 Related Work

Paraphrases are sentences/phrases which contain different words or different sequence of words but have the same meaning. However, as explained in (Bhagat and Hovy 2013), loosely equivalent and semantically similar sentences/phrases can also be considered as paraphrases. There have been mainly three lines of work for generating paraphrases. The first line of work uses statistical and rule based methods (Fader, Zettlemoyer, and Etzioni 2013) to mine paraphrase pairs, typically for short phrases from large monolingual (Barzilay and McKeown 2001) or bilingual corpora (Bannard and Callison-Burch 2005). While these methods extract a large set of paraphrase pairs (*e.g.,* the PPDB Corpus (Ganitkevitch, Van Durme, and Callison-Burch 2013)), the phrases are mostly short. To overcome this limitation, the second line of work focuses on neural models for paraphrase generation. (Mallinson, Sennrich, and Lapata 2017) use neural machine translation to generate paraphrases by first translating the phrase from English to multiple translations in the reference language (*e.g.,* German) and translate these translations back to English. (Prakash et al. 2016) and (Li et al. 2018) use sequence-to-sequence networks to directly generate paraphrases of a given input sentence. To encourage diversity among the paraphrases generated, (Gupta et al. 2018) and (Kumar et al. 2019) propose different methods based on sequence to sequence models. The third line of work uses paraphrasing as an intermediate stage for the task of Question Answering *e.g.,* (Berant and Liang 2014). However, these papers focus on their end goal of Question Answering and they don't evaluate the quality of paraphrases themselves in their experiments.

There are mainly 2 limitations which we observed while experimenting with the methods presented in these papers: either they lacked variety in their paraphrases or they failed to generate relevant paraphrases when used in the domains we are interested in, *e.g.,* customer service automation, possibly due to lack of sufficient domain-specific data.

To generate a rich variety of paraphrases, our paper uses multiple techniques like back-to-back machine translation and replacement using PPDB (Ganitkevitch, Van Durme, and Callison-Burch 2013) and WordNet (Miller 1995) resources. We then use a candidate selection algorithm leveraging recent advances in contextual embeddings, ((Devlin et al. 2019) and (Cer et al. 2018)) to select high quality paraphrases with strong semantic similarity with the input sentence. To our knowledge, there is only one other paper which uses a candidate selection algorithm (Duboué and Chu-Carroll 2006) where they use a classifier for selecting good paraphrases. Moreover the focus is on Question Answering and hence there is no evaluation of the quality of their paraphrases themselves.

3 Problem Formulation

A knowledge base article (such as FAQs and manuals) usually consists of a *title* and an associated *description*. A user who needs help with a particular issue can frame the same issue in different ways. For example, a user who wants to pay their bill can use "How do I pay my bill?", "I want to pay my bill", "I wish to settle my dues". Whereas the article can be titled as "Pay your bill". Information Retrieval based models lack recall when the words chosen by a user are different from the article but are semantically related. Enriching articles with utterances that are semantically similar to their content have shown to significantly improve recall and precision of IR based models in our experiments.

This work addresses the problem of automatically generating utterances for a given article, which can be further curated and used by human annotators to prepare a final list of reference utterances for the article. The method described in this work first uses summarization techniques to find important sentences in an article. Next, paraphrase generation techniques are used to generate many candidate utterances for each of these sentences as well as the title. Finally, a novel candidate selection algorithm which performs filtering and de-duplication is used to discard candidates which do not make sense or are not semantically similar to the original sentence and to remove duplicate paraphrases.

4 Proposed Method

A conversational assistant user can refer to an article using a question which is either a paraphrase of the title or is related to the text present in the description. The former motivates the need for paraphrase generation. However, descriptions can be long and often contain sentences that users don't refer to. Hence, we use extractive summarization to select important sentences from the description, following which we generate paraphrases for each of the extracted sentences as well as the title of an article. Our aim here is to generate a diverse set of paraphrases for a sentence and hence, we choose to adapt the overgenerate and selection paradigm. We first use multiple paraphrasing techniques to generate a large pool of paraphrases following which we use a novel candidate selection algorithm to select useful and relevant paraphrases for each input sentence. Next, as parts of utterance generation, we will describe our methods for Paraphrase Generation, Candidate Selection and Summarization.

Paraphrase Generation

We use many different methods for generating paraphrases such as (1) full backtranslation, (2) noun/verb phrase backtranslation using constituency parsing, (3) synonym replacement, and (4) phrase replacement.

- **Full Backtranslation (BT):** Inspired from (Mallinson, Sennrich, and Lapata 2017), we use neural machine translation models for generating paraphrases. We first generate multiple German translations of the input English sentence. For each of the German translations, we generate multiple English translations. In order to generate multiple translations, we use beam search at the time of decod-

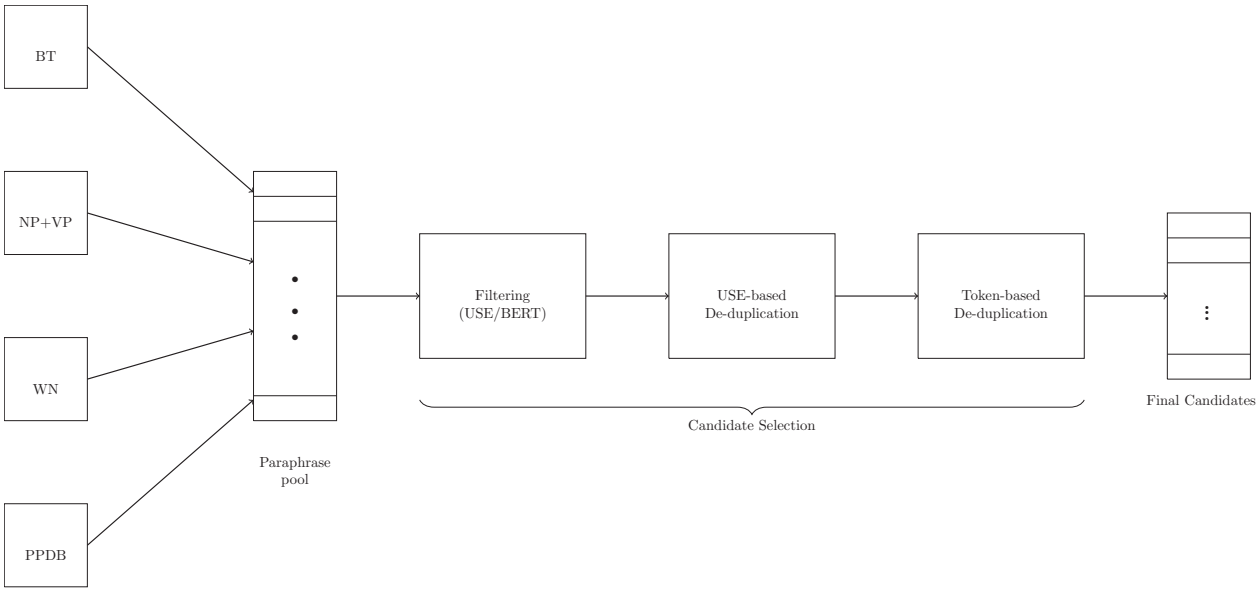

Figure 1: A figure representing the generation and candidate selection pipeline

ing. We also experimented with Czech, however, German seemed to work better for us.

- **Noun/Verb Phrase Backtranslation (NP/VP):** Backtranslating an entire sentence can often generate lots of duplicate paraphrases, especially when the input sentence is long. Hence, we also generate paraphrases for only a certain meaningful phrase from the input sentence. We use the Berkeley Neural Parser (Kitaev and Klein 2018) to perform constituency parsing and extract all noun and verb phrases from the input sentence. For each of these extracted phrases, we generate backtranslations and replace the phrase in the original sentence with its respective backtranslations.
- **Synonym Replacement (WN):** Often times, paraphrasing involves replacing a single word with another having equivalent meaning in the context. To account for this, we find synonyms for words in the input sentence from synsets obtained using WordNet (Miller 1995) and replace the word with its synonym. We do not consider words that are stopwords, whose Part-of-Speech tag belongs to a manually curated list of tags or that are less than 3 characters long.
- **Phrase replacement (PPDB):** WordNet usually contains synonyms for only single words, whereas noun and verb phrase backtranslation generate paraphrases for only certain types of phrases. PPDB (Ganitkevitch, Van Durme, and Callison-Burch 2013) is a database of paraphrases of commonly occurring phrases, extracted from a bilingual corpus. We use this resource to replace all matching phrases from the input sentence with their paraphrases.

Candidate Selection

Using multiple techniques for paraphrasing generates a large pool of paraphrases which could potentially contain sentences that are semantically different from the input sentence or synonyms replaced in the wrong context as well as duplicates of the title and each other. This necessitates a method to select relevant candidate paraphrases. As part of our candidate selection algorithm, we first remove the irrelevant sentences using a filtering mechanism, following which we use a de-duplication method to remove duplicates.

Filtering The goal of filtering is to remove paraphrases that are not semantically equivalent. We describe two different filtering methods that we experiment with.

- **USE-based:** We use the Universal Sentence Encoder (Cer et al. 2018) to get vector representations of the input sentence and the paraphrase and compute the cosine similarity between them. If the cosine similarity between the representations is less than 0.5, the paraphrase is considered to be semantically different and is discarded. Analogously, if the similarity is greater than 0.95, the paraphrase is considered to be a duplicate of the input and hence, also discarded. These thresholds were fine-tuned after experimenting with different thresholds on a set of positive and negative paraphrase pairs.
- **BERT-based:** USE-based similarity determines the semantic similarity between two sentences, however, it does not explicitly tell us if the sentences are semantically equivalent. For similar sentences (*e.g.,* sentences with high word overlap) that are not semantically equivalent, USE-based filtering fails to give the desired result. In order to improve precision of filtering, we use a paraphrase detection model based on BERT (Devlin et al. 2019). This model is trained on labeled pairs from Quora Question Pairs (as given by (Wang et al. 2019)), MRPC (Dolan and Brockett 2005), STS Benchmark (Cer et al. 2017) and PAWS (Zhang, Baldridge, and He 2019).

Source	Generated 1	Generated 2
how to resume the preset speed ?	how can i restart the default speed ?	how to recover the preset speed ?
how do i activate voice commands?	how do i activate the speech command ?	how do i activate voice control ?
change your payment method	payment method amendment	switch your payment method
credit limit increases	credit bound increase	raising the credit limit
when can i rely on icc?	when can i be dependent upon icc ?	when can i count on icc ?

Table 1: Examples of the some of the useful paraphrases generated by our method.

Deduplication In order to remove duplicates, we run the following two algorithms sequentially after the filtering step. Algorithm 1 uses similarity based on USE to de-duplicate the pool, that is, at every step, it finds the paraphrase that has the highest cosine similarity with the original sentence and selects it if it does not have a high similarity with any of the paraphrases already selected. Algorithm 2 focuses on diversifying the final set by selecting the paraphrase with the highest number of unique words at every step. We only consider words that are not stopwords, have a character length of more than 2 and whose POS tags do not belong to a manually curated list of POS tags (such as prepositions, conjunction words, and forms of the verbs "be" and "have")

```
output=[];
sort(pool); sentencoding=USE(input);
for paraphrase in pool do
    vector=USE(paraphrase);
    for paraphrase2 in output do
        if cosine(USE(paraphrase2), vector) > 0.95
          then
            break;
        end
    end
    if cosine(vector,sentencoding) < 0.95 then
        append paraphrase to output;
    end
end
```

Algorithm 1: Deduplication using USE

```
wordset={words in input};
output=[];
while len(output) < k do
    select paraphrase from pool with most number of
      unique words;
    if no such paraphrase exists then
        break;
    else
        output.append(paraphrase);
        pool.remove(paraphrase);
        wordset.append(new words in input)
    end
end
```

Algorithm 2: Word based deduplication

Selection during tie-breaking While performing deduplication, many of the paraphrases generated have just one or two keywords that are different and unique from the input sentence. It is important to select the sentences that are more related to the input sentence and which also are more probable as a sentence. Hence, for each paraphrase, we compute two scores, namely, the similarity between the USE encodings of the input sentence and the paraphrase, and a score computed using the cross entropy loss from the BERT model probabilities. We normalize both of these scores across examples and use the average for tie-breaking.

Summarization

To select the important sentences from the description, we use extractive summarization. We experimented with the pre-trained models provided by (Chen and Bansal 2018) and (Narayan, Cohen, and Lapata 2018) and chose to go with the former after evaluating on a private test set. For more details regarding the description of the model, we defer to the original paper. We only summarize a description if it is more than 3 sentences long. Otherwise, we pick all sentences as important sentences.

5 Experiments

In this section, we focus on evaluating the paraphrase generation method. While metrics like BLEU (Papineni et al. 2002), ROUGE (Lin 2004) and METEOR (Lavie and Agarwal 2007) have been proposed for automatic evaluation, as pointed out in (Callison-Burch, Osborne, and Koehn 2006) and (Lavie and Agarwal 2007), these measures are inadequate since they perform n-gram matching, and do not capture diversity. Since the aim is to produce paraphrases that are as diverse as possible, it is hard to come up with all possible reference sentences. Hence, we focus on results of manual evaluation and for completeness, we also report the BLEU score. We also present evaluation results of the CVAE-based model described in (Gupta et al. 2018). Note that we only evaluate the paraphrase generation method as we believe that is the main contribution of this work. Since the generated utterances will be further curated and referred to by humans, we believe that evaluation on the end task of information retrieval is more nuanced and should be explored as part of future work.

Dataset

For manual evaluation, we prepare a test set of 100 sentences from which useful utterances can be generated, manually chosen from articles in auto manuals, telecom company FAQs, and retail FAQs. We report BLEU score on two publicly available datasets, namely, the STS Benchmark (371 sentences) and the MRPC corpus (273 sentences) as well as a private dataset (567 sentences) which contains sentences from auto manuals along with manually generated paraphrases by upworkers. For training the CVAE-based model,

Method	Avg. Fraction of useful		Avg. number of useful	
	USE	BERT	USE	BERT
BT	**0.6**	**0.608**	2.31	1.16
NP + VP	0.42	0.5	3.16	1.76
BT + VP + NP	0.40	0.5	3.37	2.78
WordNet	0.216	0.315	2.79	2.73
PPDB	0.26	0.298	2.31	2.03
BT + VP + NP + WN + PPDB	0.24	0.44	**4.27**	**5.97**
CVAE	0.05	-	0.37	-

Table 2: Results for manual evaluation. We explain in Discussion why we don't perform filtering with BERT on CVAE.

Method	MRPC			STS			Private		
	No CS	USE	BERT	No CS	USE	BERT	No CS	USE	BERT
BT	0.201	0.169	0.199	0.225	0.14	**0.239**	0.217	0.205	**0.234**
NP + VP	0.345	**0.325**	0.325	0.172	0.173	0.173	0.155	0.172	0.172
BT + VP + NP	0.329	0.306	0.306	0.18	0.173	0.174	0.171	0.183	0.183
WordNet	**0.393**	0.27	**0.36**	**0.3**	**0.26**	0.173	**0.27**	**0.256**	0.173
PPDB	0.27	0.236	0.262	0.246	0.213	0.169	0.249	0.24	0.171
BT + NP + VP + WN + PPDB	0.334	0.308	0.307	0.178	0.175	0.176	0.172	0.176	0.176
CVAE	0.089	0.058	-	0.073	0.0297	-	0.194	0.139	-

Table 3: Result of BLEU scores on different datasets. No CS indicates no candidate selection algorithm.

we use the Quora Question Pair dataset along with paraphrase pairs from ComQA (Abujabal et al. 2018) and our private dataset.

Evaluation Methodology

For each input sentence, we generate paraphrases and restrict the number of generated paraphrases to a maximum of 20. For backtranslation, we first generate 5 German translations and further generate 5 English translations for each of these, resulting in a total of 25 backtranslations. For the CVAE-based model, we sample 25 random seeds to generate 25 different paraphrases. We use the final pool obtained from aggregating the paraphrases generated using all methods as an input to the candidate selection.

Manual Evaluation Each of the generated paraphrase is marked as either useful (1) or not useful (0) by the human annotator. We report two metrics for manual evaluation. The first metric is a precision-based metric which computes the fraction of paraphrases that are useful for each sentence and takes an average of this fraction. The second metric that we report is the absolute number of useful paraphrases generated per input sentence averaged across all input sentences. We report this in Table 2.

Automatic Evaluation For each of the generated paraphrase, we use the set of reference sentences to compute BLEU score. The score for an input sentence is the average of the BLEU score over all the generated paraphrases for the sentence. We report the average of this score over all input sentences, for the paraphrase sets obtained both before and after filtering. This is reported in Table 3.

Discussion

We make multiple observations from Tables 2 and 3.

- Combining multiple techniques generates more number of useful paraphrases than each of the individual methods.
- Using BERT for filtering improves the precision notably. Although it reduces the number of useful paraphrases for individual methods, it works better when all methods are combined. Analysis of the results using USE filtering showed that it gave high score to irrelevant paraphrases generated using WordNet (lot of overlapping words), resulting in a high fraction of them being included in the final set, while useful paraphrases from other methods were given lower score and not included in final set. BERT-based filtering helped in reducing the total number of WordNet-based paraphrases, allowing paraphrases from other methods to also be included in the final set.
- BLEU scores are not consistent with the human evaluation and hence are not very reliable for our purposes.
- The CVAE-based model does not perform as well for our purpose as the other methods. We trained multiple models on different combinations of datasets and saw that the model failed to generate good quality paraphrases for previously unseen sentences. It is for this reason that we did not proceed with experimenting BERT-filtering on CVAE.

6 Conclusion

In this work, we addressed the problem of utterance generation. We use the title of an article along with sentences extracted using summarization of the description as reference sentences. We propose an ensemble method that uses multiple paraphrasing techniques to generate a set of paraphrases from an input sentence. We also propose some innovative ways of paraphrasing using constituency parsing and noun/verb phrase neural backtranslation. Finally, we devel-

oped a novel candidate selection algorithm to filter out bad utterances and remove duplicates for diversity using some of the latest contextual embedding techniques such as BERT and Universal Sentence Encoder. Our experimental results show that our overall approach works well. We tried conditional variation autoencoder (CVAE) based techniques but did not get good utterances generated. In the future, we plan to further fine tune and experiment with sequence-to-sequence models like CVAE for generating utterances.

References

Abujabal, A.; Roy, R. S.; Yahya, M.; and Weikum, G. 2018. Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. *CoRR* abs/1809.09528.

Bannard, C., and Callison-Burch, C. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 597–604.

Barzilay, R., and McKeown, K. R. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL)*, 50–57.

Berant, J., and Liang, P. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1*, 1415–1425.

Bhagat, R., and Hovy, E. H. 2013. What is a paraphrase? *Computational Linguistics* 39:463–472.

Callison-Burch, C.; Osborne, M.; and Koehn, P. 2006. Re-evaluation the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval)*, 1–14.

Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.; Strope, B.; and Kurzweil, R. 2018. Universal sentence encoder. *CoRR* abs/1803.11175.

Chen, Y., and Bansal, M. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1*, 675–686.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1*, 4171–4186.

Dolan, B., and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP)*. Asia Federation of Natural Language Processing.

Duboué, P. A., and Chu-Carroll, J. 2006. Answering the question you wish they had asked: The impact of paraphrasing for question answering. In *Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*.

Fader, A.; Zettlemoyer, L. S.; and Etzioni, O. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, (ACL)*, 1608–1618.

Ganitkevitch, J.; Van Durme, B.; and Callison-Burch, C. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 758–764.

Gupta, A.; Agarwal, A.; Singh, P.; and Rai, P. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI)*, 5149–5156.

Kitaev, N., and Klein, D. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1)*.

Kumar, A.; Bhattamishra, S.; Bhandari, M.; and Talukdar, P. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, 3609–3619.

Lavie, A., and Agarwal, A. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT, 228–231.

Li, Z.; Jiang, X.; Shang, L.; and Li, H. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3865–3878.

Lin, C.-Y. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 74–81.

Mallinson, J.; Sennrich, R.; and Lapata, M. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1*, 881–893.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Volume 1*, 1747–1759.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318.

Prakash, A.; Hasan, S. A.; Lee, K.; Datla, V.; Qadir, A.; Liu, J.; and Farri, O. 2016. Neural paraphrase generation with stacked residual LSTM networks. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, 2923–2934.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Zhang, Y.; Baldridge, J.; and He, L. 2019. PAWS: paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of (NAACL-HLT), Volume 1*, 1298–1308.