Generating Syntactic Paraphrases

Emilie Colin
Lorraine University/LORIA
Nancy, France
emilie.colin@loria.fr

Claire Gardent
CNRS/LORIA
Nancy, France
claire.gardent@loria.fr

Abstract

We study the automatic generation of syntactic paraphrases using four different models for generation: data-to-text generation, text-to-text generation, text reduction and text expansion, We derive training data for each of these tasks from the WebNLG dataset and we show (i) that conditioning generation on syntactic constraints effectively permits the generation of syntactically distinct paraphrases for the same input and (ii) that exploiting different types of input (data, text or data+text) further increases the number of distinct paraphrases that can be generated for a given input.

1 Introduction

The ability to automatically generate paraphrases (alternative phrasings of the same content) has been shown to be useful in many areas of Natural Language Processing such as question answering (Riezler et al., 2007), semantic parsing (Berant and Liang, 2014)), machine translation (Kauchak and Barzilay, 2006; Zhou et al., 2006), sentence compression (Napoles et al., 2011) and sentence representation (Wieting et al., 2015). From a linguistic standpoint, the automatic generation of paraphrases is an important task in its own right as it demonstrates the capacity of NLP techniques to handle a key feature of natural language.

In this paper, we focus on the automatic generation of *syntactic paraphrases* that is, texts which share the same meaning but differ in their syntax. Our work makes the following contributions. We show that conditioning text generation on syntactic information permits generating distinct syntactic paraphrases for the same input. We provide a systematic exploration of how different types of generation tasks impact paraphrasing and show that exploiting different types of input permits increasing the number of paraphrases produced for a given input. We make available four training corpora for syntactically constrained, data- and text-to-text generation, text expansion and text reduction.

2 Related Work

Previous work on paraphrase generation falls into three main groups. Based mainly on monolingual data, earlier approaches use data-driven, (Lin and Pantel, 2001), grammar- or thesaurus-based methods (Madnani et al., 2007; McKeown, 1983; Hassan et al., 2007; Kozlowski et al., 2003; Quirk et al., 2004; Zhao et al., 2008). In contrast, the pivot-based approach exploits bilingual data and machine translation methods to extract and generate paraphrases (Callison-Burch, 2008; Ganitkevitch and Callison-Burch, 2014; Ganitkevitch et al., 2011). Finally, neural approaches build upon the encoder-decoder architecture to learn paraphrase generation models (Mallinson et al., 2017; Prakash et al., 2016).

(Prakash et al., 2016) uses a stacked residual LSTM network with residual connections between LSTM layers and show that their model outperforms sequence to sequence, attention-based, and bi- directional LSTM model on three datasets (PPDB, WikiAnswers, and MSCOCO).

(Mallinson et al., 2017) introduces a neural model for multi-lingual, multi-pivot backtranslation and show that it outperforms a paraphrase model trained with a commonly used Statistical Machine Translation system (SMT) on three tasks, namely, correlation with human judgments of paraphrase quality; paraphrase and similarity detection; and sentence-level paraphrase generation.

(Iyyer et al., 2018) also use backtranslation as a mean to provide training data. In addition, it uses syntax to control paraphrase generation. Given

D2T$_{5best}$
Aktieselskab is the operating organisation for Aarhus Airport which has a runway name of "10R/28L" with a length of 2777.
The Aktieselskab is the operating organisation for Aarhus Airport which has a runway name of "10R/28L" with a length of 2777.
Aktieselskab is the operating organisation for Aarhus Airport which has a runway length of 2777 and is named "10R/28L".

T2T$_{5best}$
Operated by Aktieselskab, Aarhus Airport has a runway length of 2777 metres and a runway name "10R/28L".
Operated by the Aktieselskab organisation, Aarhus Airport, has a runway length of 2777 metres and a runway named "10R/28L".
Aarhus Airport is operated by Aktieselskab. Its runway name is "10R/28L" and its length is 2777 metres.
Aarhus Airport, operated by Aktieselskab, has a runway length of 2777 with a name of "10R/28L".
Aktieselskab is the operation organisation of Aarhus Airport where the runway "10R/28L" with a length of 2777.
The Aktieselskab is the operating organisation of Aarhus Airport., This airport has a runway length of 2777 metres and a runway named "10R/28L".
The "10R/28L" runway at Aarhus Airport is 2777 in length, which is run by the operating organization of Aktieselskab.
The "10R/28L" runway at Aarhus Airport, operated by the Aktieselskab, is 2777 in length.

ALL$_{syn}$
Aarhus Airport has a runway length of 2777.0 metres and is operated by Aktieselskab. The name of the runway is "10L/28R".
Aarhus Airport is operated by Aktieselskab; its runway name is "10L/28R" and its runway length is 2777.0.
Aarhus Airport is operated by Aktieselskab, its runway name is "10R/28L" and has a length of 2777.
Aarhus Airport is operated by the Aktieselskab organisation. Its runway name is "10R/28L" and has a length of 2777.
Aarhus Airport, operated by Aktieselskab, has a runway length of 2777 and the runway name is "10R/28L".
Aarhus Airport, operated by Aktieselskab, has a runway length of 2777 with a name of "10R/28L".
Aarhus Airport, which is operated by the Aktieselskab organisation, has a runway that 's 2777.0 long and is named "10L/28R".
Aktieselskab is the operation organisation of Aarhus Airport where the runway "10R/28L" with a length of 2777.
Aktieselskab is the operation organisation of Aarhus Airport, where the runway is named "10R/28L", with a length of 2777.
Aktieselskab is the operation organization for Aarhus Airport, where the runway is named "10L/28R", with a length of 2777.0.
Aktieselskab operates Aarhus Airport which has a runway that is 2777 meters long and the runway name "10R/28L".
Operated by Aktieselskab, Aarhus Airport, has a runway length of 2777 metres and a runway named "10R/28L".
Operated by Aktieselskab, Aarhus Airport, has a runway length of 2777 metres and is named "10R/28L".
Operated by the Aktieselskab organisation, Aarhus Airport has a runway that is 2777.0 metres long. It also has a runway with the name "10L/28R".
The "10R/28L" runway which is 2777 meters long is located in Aarhus Airport which is operated by the Aktieselskab organisation.

Table 1: Some Example Output

TX$_{syn}$		$T_i, k, t \Rightarrow T_o$ with $mg(T_o) = mg(T_i) \cup \{t\}, k \in K(T)$
	T_i	*Madrid is part of Community of Madrid whose leader party is the Ahora Madrid. The Adolfo Suarez Madrid-Barajas Airport is located there.*
	k	possessive
	t	{ (Madrid country Spain) }
	T_o	*Adolfo Suarez Madrid-Barajas Airport is located in Madrid, part of the Community of Madrid in Spain where the leader party is Ahora Madrid.*
TR$_{syn}$		$T_i, k, t \Rightarrow T_o$ with $mg(T_i) = mg(T_o) \cup \{t\}, k \in K(T)$
	T_i	*Al Asad Airbase is located at "Al Anbar Province, Iraq" and operated by the United States Air Force. The base 's runway called "08/26" and 3990 meters long.*
	k	Subject Relative
	t	{ (Al Asad Airbase operatingOrganisation United States Air Force) }
	T_o	*Al Asad Airbase (in "Al Anbar Province, Iraq"), has a runway named "08/26" and a runway that is 3990 metres long.*

Table 2: TR: Text Reduction, T_i, T_o: input/output text, k: syntactic constraint, M: meaning representation, a set of RDF triples, $mg(T)$: meaning of text T, $K(T)$: syntactic constraints realised by text T).

a syntactic template T and an input sentence S, the model first generates a full syntactic parse P_T. Next this syntactic parse is used together with the input sentence to predict a syntactic paraphrase of S which realises the input syntactic template T.

Our approach is closest to (Iyyer et al., 2018) but differs from it in that instead of restricting paraphrase generation to a text rewriting problem, we explore how various sources of input impacts the number and the type of generated paraphrases. It also differs from the former two approaches (Prakash et al., 2016; Mallinson et al., 2017) in that we focus on syntactic paraphrases and condition generation on syntax. In that sense, our approach also shares similarities with recent models for controllable text generation (Hu et al., 2017; Semeniuta et al., 2017), which use variational autoencoders to model holistic properties of sentences such as style, topic and various other syntactic features. Our work is arguably conceptually simpler, focuses on syntactic paraphrases and introduces a new text production mode based on hybrid "data and text" input.

3 Generating Syntactic Paraphrases

In order to generate syntactically distinct paraphrases, we formulate the generation task as a structured prediction task conditioned on both some input I and some syntactic constraint k. In this way, the same input I can be mapped to several output T_i each satisfying a different syntactic constraint k_i. Table 1 shows some examples.

In addition, we consider different, semantically equivalent, sources of information. That is, we compare the paraphrases obtained when generating text from data, from text or from text and data. For the later, we consider two subtasks namely text expansion and text reduction. For each of these two tasks, the input is a text and a data unit. For text expansion, the output is a text verbalising both the input text and the input data. Conversely, for text reduction, the output is a text verbalising the input text minus the text verbalising the input data. Table 2 shows some example input and output for text expansion and text reduction.

4 Training and Test Data

Training data. The WEBNLG dataset (Gardent et al., 2017) associates sets of RDF triples with one or more texts verbalising these sets of triples. We derive training corpora for syntactically constrained generation from this dataset as follows.

We enrich the WEBNLG texts with labels indicating syntactic structures that are realised by these texts by first, parsing[1] these texts and then using syntactic templates to identify the target structures occurring in those texts. We use the following list of syntactic labels: subject relative, object relative, sentence coordination, VP coordination, passive voice, apposition, possessive relative, pied piping, transitive clause, prepositional object, ditransitive clause, predicative clause.

Based on the resulting, syntactically enriched, WEBNLG corpus, we then build four training corpora (T2T$_{syn}$, TX$_{syn}$, D2T$_{syn}$, TR$_{syn}$) using the sets of RDF triples as pivots to relate paraphrases. For data-to-text generation (D2T$_{syn}$), the input is a linearised and delexicalised version of the set of RDF triples representing the meaning of the output text, for text-to-text generation (T2T$_{syn}$), the input is a text and for hybrid data-and-text-to-text generation (TX$_{syn}$ and TR$_{syn}$), the input is a text and a linearised RDF triple.

For the text-to-text datasets, we additionally require that, for any corpus instance $\langle k, T_i, T_o \rangle$, T_o differs from T_i on exactly one syntactic label[2].

Test data. For any input $\langle k, I \rangle$ occuring in the test data, we ensure that $\langle k, I \rangle$ does not occur in the training data. (where I is either a set of RDF triples, a text or a text and an RDF triple).

5 Experimental Setup

Models and Baselines D2T$_{5best}$ and T2T$_{5best}$ For each generation task, we aim to learn a model that maximises the likelihood $P(T|I; k; \theta)$ of a text given some input I, some model parameters θ and some syntactic constraint k. We use a simple encoder-decoder model where both encoder and decoder are bidirectional LSTMs and the encoder receives as input a sequence including both the input I and the syntactic constraint k.

We compare our models with the output produced by beam search when no syntactic constraint applies. For D2T$_{5best}$, we take the 5 best output generated from data. For T2T$_{5best}$, there may be several input sentences associated with the same meaning: we take the 5 best output for each

[1] We used the Stanford CoreNLP dependency parser version 3.8, 2018-06-09

[2] $K(T_i) = (K(T_i) \cap K(T_o)) \cup \{k\}$ and $K(T_o) = (K(T_i) \cap K(T_o)) \cup \{k'\}$ for some $k \neq k'$.

of these sentences hence T2T$_{5best}$ may (and does) in fact yield more than 5 output per input meaning. Finally, ALL$_{syn}$ groups together all output generated by the four syntactically constrained models for a given meaning.

Implementation Details We use the OpenNMT$_{py}$ sequence-to-sequence model (Klein et al., 2017) with attention and a bidirectional LSTM encoder. The encoder and decoder have two layers. Models were trained for 13 epochs, with a mini-batch size of 64, a dropout rate of 0.3, and a word embedding size of 500. They were optimised with SGD with a starting learning rate of 1.0.

Evaluation. We assess both the linguistic/syntactic adequacy of the generated texts and the diversity of the paraphrases being generated.

Syntactic and Linguistic Adequacy (BLEU, Synt, BLEU$_{syn}$). For the syntactically constrained models, given an input syntactic constraint k, the BLEU score[3] is computed with respect to those references which satisfy k. In that way, the BLEU score indicates how close to the syntactic target the generated sentence is and therefore how well the model succeeds in generating the required syntactic constructs – as the number of references varies across inputs, we use BLEU at the sentence level (Papineni et al., 2002). In addition, we compute the proportion of output satisfying the input syntactic constraint (Synt) and the BLEU score for these output which satisfy the input syntactic constraint (BLEU$_{syn}$). The number of output satisfying the input syntactic constraint is computed by first parsing the generated output and then applying the templates used for the automatic annotation of the training data.

Diversity (Sim, #Txt/Mg). To measure the level of paraphrasing obtained, we group together inputs which share the same meaning (i.e., inputs that are linked in the WEBNLG dataset to the same set of RDF triples) and we compute the number of distinct texts generated per meaning (# Txt/Mg). We further analyse these sets by computing the average pairwise similarity (Sim) of the texts present in these sets. We use the Ratcliff/Obershelp algorithm (Black, 2004) to compute similarity[4]. A low similarity indicates more diversity across the set of outputs sharing the same meaning.

Human Evaluation (% SPar). For each model, we manually examined for 50 meanings, a maximum of 10 randomly chosen output and recorded the average number (# SPar) of syntactically correct paraphrases per input.

6 Results

Table 3 summarises the results.

Diversity. The results for ALL$_{syn}$ (aggregating all output texts generated for a given meaning) shows that combining different generation models increases diversity (# Txt/Mg:13.25, Sim:0.61)) while maintaining a good level of linguistic (BLEU:62.87) and syntactic adequacy (Synt:0.91).

The human evaluation further shows that the distinct outputs generated by the ALL$_{syn}$ model are indeed syntactic, not purely lexical, variants. Table 1 shows some example output for ALL$_{syn}$.

Expansion, Reduction and Generation. Interestingly, the text expansion and reduction models markedly improve on traditional T2T and D2T models both in terms of linguistic adequacy (higher BLEU score) and in terms of diversity (higher number of distinct output per meaning, lower similarity between texts generated from the same meaning). The comparison with T2T generation is particularly striking as the training data is 3 to 5 times larger for the T2T$_{syn}$ model than for the TX$_{syn}$ and the TR$_{syn}$ model respectively. Similarly, it is noticable that although the T2T$_{syn}$ training corpus is 3 times larger than the D2T$_{syn}$ corpus, the T2T$_{syn}$ and the D2T$_{syn}$ models show similar results. This is in line with results from (Aharoni and Goldberg, 2018) which shows that rephrasing is a difficult task.

Linguistic Adequacy. Overall the linguistic adequacy of the syntactically constrained models is high with a BLEU score with respect to a single reference ranging from 46.20 (D2T$_{syn}$) to 83.87 (TX$_{syn}$). Moreover, the generated sentences show close similarity with the reference sentence realising the input constraint (BLEU$_{syn}$: from 48.16 to 89.32).

[3] We use the sacrebleu script with BLEU-4.

[4] The Ratcliff/Obershelp similarity score varies between O and 1 where 1 is a complete match. It is expressed by the formula $sim(S1, S2) = \frac{2*match(S1,S2)}{len(S1)+len(S2)}$ where a match is defined as the sum of the length of the matching segments ($match(S1, S2) = \sum_{m \in overlap(S1,S2)} len(m)$.

Model	BLEU -	Synt -	BLEU$_{syn}$ -	Sim test (vs ref)	# Txt/Mg (# txt/Mg ref)	# Corpora	# SPar/Mg (# outputs)
T2T$_{5best}$	6.21	N/A	N/A	0.76 (0.61)	5.85 (44.98)	715910	3.98 (5.5)
D2T$_{5best}$	4.71	N/A	N/A	0.81 (0.63)	4.71 (5.98)	27156	1.86 (4.42)
TR$_{syn}$	66.63	0.70	80.45	0.62 (0.63)	2.92 (6.10)	40936	2.56 (3.9)
TX$_{syn}$	83.87	0.88	89.32	0.68 (0.68)	2.56 (6.55)	74844	0.92 (1.16)
T2T$_{syn}$	49.95	0.98	50.23	0.72 (0.68)	1.92 (18.49)	202218	1.58 (1.70)
D2T$_{syn}$	46.20	0.84	48.16	0.81 (0.61)	1.04 (3.87)	66595	1 (1.1)
ALL$_{syn}$	62.87	0.91	65.11	0.61 (0.62)	13.25 (68.95)	NA	6.3 (15)

Table 3: Results and Corpora Size (BLEU score wrt reference satisfying the input constraint k, Synt: proportion of output satisfying k; BLEU$_{syn}$: BLEU score for output satisfying k; Sim: average pair-wise similarity between sentences output for a given (data or text) input (in brackets: the similarity score calculated on the reference corpus); # Txt/Mg: avg nb of distinct text generated per meaning (in brackets: the average number of texts per meaning occurring in the reference data); # Corpora : size of the corpora; # SPar/Mg (%): Avg number of syntactic paraphrases per meaning found by human evaluation (in brackets: the average total number of output considered)

While the baseline models underperform in terms of BLEU scores, the manual evaluation (# SPar/Mg) indicates that they, in fact, produce acceptable output. The low BLEU scores for these models are probably due to the fact that each output is evaluated against a single reference while the dataset is constructed to maximise the number of paraphrases available for a given input.

7 Some examples

Table 1 shows some example outputs illustrating the main differences between the D2T$_{5best}$, T2T$_{5best}$ and the ALL$_{syn}$ model. As these examples show, syntactically constrained generation (ALL$_{syn}$) outputs a much larger number of paraphrases. The difference is due both to the fact that ALL$_{syn}$ groups together the output of 4 (syntactically driven) generation models and to the input syntactic constraint, which ensures greater diversity. Thus in the example shown, ALL$_{syn}$ yields 15 paraphrases each with strong syntactic differences as summarised below.

Sentence Segmentation. The number of verb phrases, clauses and sentences used to verbalise the same input varies. One output text is made of 2 sentences and one VP coordination, another of 3 coordinated clauses and a third of two coordinated clauses and a VP coordination.

Syntax. The same input property is realised by different syntactic structures. For instance, the property *operatingOrganisation* is alternatively realised by an active verb (*operates*), a passive verb (*is operated by*), a participial apposition (*,operated by ..,*), a subject relative (*which is operated by*), a nominal predicative construction (*is the operation organization*) and a preposed participial (*Operated by .., *).

Word Order. The same content is verbalised using varying word order and clause ordering. Thus the ALL$_{syn}$ output shows four different ways of ordering the realisation of the three properties *operatinOrganization (oO), runwayLength (rL), runwayName (rN)* contained in the input namely, *rL-oO-rN* (once), *oO-rN-rL* (6 times), *oO-rL-rN* (6 times) and *rN-rL-oO* (once).

By constrast, the baseline models output a much smaller range of syntactic paraphrases. The D2T$_{5best}$ model is particularly weak as among the five best outputs it produces, only three are distinct and all have almost identical syntax. The T2T$_{5best}$ model produces more outputs (8 against 3 for the D2T$_{5best}$ model and 15 for the ALL$_{syn}$ model). One reason for this is that, contrary to the D2T$_{5best}$ model which has a single input (namely a set of RDF triples), this model can have several inputs for the same set of RDF triples.

8 Conclusion

We have proposed new syntactically constrained models for text generation and shown that their use effectively supports the generation of syntactic paraphrases. In future work, we plan to investigate to what extent these methods can be used to support the automatic generation of grammar exercises.

References

Roee Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and a stronger baseline. In *ACL*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1415–1425.

Paul E Black. 2004. Ratcliff/obershelp pattern recognition. *Dictionary of Algorithms and Data Structures*, 17.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 196–205. Association for Computational Linguistics.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In *LREC*, pages 4276–4283.

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin Van Durme. 2011. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1168–1179. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. 2007. Unt: Subfinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 410–413. Association for Computational Linguistics.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Controllable text generation. *arXiv preprint arXiv:1703.00955*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885. Association for Computational Linguistics.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 455–462. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Raymond Kozlowski, Kathleen F McCoy, and K Vijay-Shanker. 2003. Generation of single-sentence paraphrases from predicate/argument structure using lexico-grammatical resources. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 1–8. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Nitin Madnani, Necip Fazil Ayan, Philip Resnik, and Bonnie J Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 120–127. Association for Computational Linguistics.

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 Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 881–893.

Kathleen R McKeown. 1983. Paraphrasing questions using given and new information. *Computational Linguistics*, 9(1):1–10.

Courtney Napoles, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme. 2011. Paraphrastic sentence compression with a character-based metric: Tightening without deletion. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 84–90. Association for Computational Linguistics.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098*.

Chris Quirk, Chris Brockett, and Bill Dolan. 2004. Monolingual machine translation for paraphrase generation.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390*.

John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Dan Roth. 2015. From paraphrase database to compositional paraphrase model and back. *arXiv preprint arXiv:1506.03487*.

Shiqi Zhao, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. 2008. Combining multiple resources to improve smt-based paraphrasing model. *Proceedings of ACL-08: HLT*, pages 1021–1029.

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating machine translation results with paraphrase support. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 77–84. Association for Computational Linguistics.