Procesamiento del Lenguaje Natural, Revista nº 64, marzo de 2020, pp. 85-92 recibido 01-12-2019 revisado 10-01-2020 aceptado 13-01-2020

A light method for data generation: a combination of Markov Chains and Word Embeddings

Un método ligero de generación de datos: combinación entre Cadenas de Markov y Word Embeddings

Eva Martínez Garcia[1], **Alberto Nogales**[1],
Javier Morales Escudero[2], **Álvaro J. García-Tejedor**[1]
[1]CEIEC-Universidad Francisco de Vitoria
[2]Avanade Iberia S.L.U.
{eva.martinez, alberto.nogales, a.gtejedor}@ceiec.es, javier.morales.escud@avanade.com

Abstract: Most of the current state-of-the-art Natural Language Processing (NLP) techniques are highly data-dependent. A significant amount of data is required for their training, and in some scenarios data is scarce. We present a hybrid method to generate new sentences for augmenting the training data. Our approach takes advantage of the combination of Markov Chains and word embeddings to produce high-quality data similar to an initial dataset. In contrast to other neural-based generative methods, it does not need a high amount of training data. Results show how our approach can generate useful data for NLP tools. In particular, we validate our approach by building Transformer-based Language Models using data from three different domains in the context of enriching general purpose chatbots.
Keywords: Generation, Hybrid, Markov Chains, Embeddings, Similarity

Resumen: Las técnicas para el Procesamiento del Lenguaje Natural (PLN) que actualmente conforman el estado del arte necesitan una cantidad importante de datos para su entrenamiento que en algunos escenarios puede ser dificil de conseguir. Presentamos un método híbrido para generar frases nuevas que aumenten los datos de entrenamiento, combinando cadenas de Markov y word embeddings para producir datos de alta calidad similares a un conjunto de datos de partida. Proponemos un método ligero que no necesita una gran cantidad de datos. Los resultados muestran cómo nuestro método es capaz de generar datos útiles. En particular, evaluamos los datos generados generando Modelos de Lenguaje basados en el Transformer utilizando datos de tres dominios diferentes en el contexto de enriquecer chatbots de propósito general.
Palabras clave: Generación, Híbrido, Cadena de Markov, Embeddings, Similaridad

1 Introduction

Neural models have become the state-of-the-art for several Natural Language Processing (NLP) approaches such as Machine Translation (MT) (Bahdanau, Cho, and Bengio, 2015; Vaswani et al., 2017; Junczys-Dowmunt, 2019), Dialogue Systems (Sordoni et al., 2015; Vinyals and Le, 2015; Serban et al., 2016; Sankar et al., 2019) or Speech Recognition (Chan et al., 2016; Moritz, Hori, and Roux, 2019; Pham et al., 2019). The most successful ones rely on supervised methods that need a large amount of data. Unfortunately, data are sometimes difficult to obtain, depending on the considered languages or domains. There are several commonly used techniques to perform data augmentation (Tanner and Wong, 1987; Inoue, 2018) like *backtranslation* (Sennrich, Haddow, and Birch, 2016) for MT. We propose a light method to generate extra data to extend a given data set. Our method allows for generating new sentences using Markov Chains (MCs) (Gagniuc, 2017). Then, it filters the generated sentences by using the semantic knowledge enclosed in a word embedding, getting the more adequate ones. We focus our work on the use case of augmenting a corpus used to build a Language Model (LM) that will help to tune chatbots designed for a

ISSN 1135-5948. DOI 10.26342/2020-64-10

specific domain. We validate our approach evaluating the impact of using the generated data to build Transformer-based Language Models by comparing the perplexity [1] of the different models. The experiments show how our MC-based generative method is able to produce adequate sentences since the language models trained using the generated data for a dating domain perform up to 2.71 perplexity points better than the ones trained with only the original data.

The paper is organized as follows. Section 2 briefly explains the state-of-the-art of Natural Language Generation (NLG) and contextualize our approach. Section 3 presents the hybrid MC-word-embedding system revisiting first the main characteristics of each technique. Then, we explain the experiments we carried out to validate our techniques and discuss the obtained results in Section 4. Finally, Section 5 draws conclusions from the presented work and discusses some possible future work lines.

2 Related Work

Natural Language Generation is the task of generating utterances from structured data representations. Data-driven NLG methods facilitate the task of corpus creation since they learn the textual structure and their surface, reducing the amount of human annotation effort. Puzikov and Gurevych (2018) propose a neural encoder-decoder model to participate in the end-to-end *E2E* NLG shared task[2]. Although their neural approach produces fluent utterances, they found out that a template-based model would obtain good results, saving developing and training time. Dušek and Jurčíček (2016) use a *seq2seq*-based generator model in combination with a re-ranking strategy for the n-best output to penalize sentences without required information or that add noise. Further, Liu et al. (2018) introduce a neural approach to generate a description for a table. Their neural model implements a seq2seq architecture consisting of a field-gating encoder, where they update the Long Short Term Memory (LSTM) cell by including a field gating mechanism, and a description generator with dual attention. This dual attention works at word and field level to model the semantic relevance between the generated description and the source table. These neural approaches are effective but they need a high amount of structured data (from 404 to $\sim 700K$ sentences used in the reviewed works). Although there are unsupervised NLG approaches that achieve state-of-the-art results (Freitag and Roy, 2018), they still need a considerable amount of data to train (they use $\sim 256K$ sentences for their unsupervised experiments), preferably in-domain data, which are sometimes scarce.

Similar to the data selection part of our approach, Inaba and Takahashi (2016) present a Neural Utterance Ranking (NUR) model to select candidate utterance according to their suitability regarding a given context. Their model processes word sequences in utterances and utterance sequences in context via Recurrent Neural Networks (RNNs) obtaining good results in ranking utterances more accurately than other methods. They also built a conversational dialog system based on their approach. In contrast, our approach uses a more simple neural model, the *word2vec* (Mikolov et al., 2013) embeddings, to select the more adequate generated sentences since we are not interested in handling the dialog context but in modeling the language that we want a chatbot to produce.

There exist approaches in the area of Dialog Systems that are similar to our method. Wen et al. (2015) use a Stochastic NLG strategy based on a joint RNN and Convolutional Neural Network (CNN). They generate sentences using a forward RNN-LM and then, they use a backward RNN-based LM and a CNN sentence model to re-rank the generated sentences. They can select the most suitable generated utterances without any semantic alignments or predefined grammar trees. Although their approach shows to be effective, they also state the need for a considerable amount of training data, using around 1,300 utterances as training set. In our case, we are dealing with at most hundreds of utterances per domain.

3 Generating More Data

First, we train a Markov Chain from a set of sentences in a given domain. Then, we use it to generate a new set of sentences, replicating the style and using the vocabulary of the

[1] The perplexity is a usual metric to evaluate LMs. It measures how well the language model can predict a word sequence.

[2] http://www.macs.hw.ac.uk/InteractionLab/E2E

original data set.

The second step of our approach is to calculate a semantic distance between the generated sentences and the sentences from the original corpus to filter out these sentences that are not sufficiently close to the target domain. Figure 1 depicts the general workflow. We want to discard those new sentences that are semantically too far from the corpus we want to extend since these sentences can add noise to the corpus and may lead to obtaining biased or wrong models in terms of adequacy regarding a given domain.

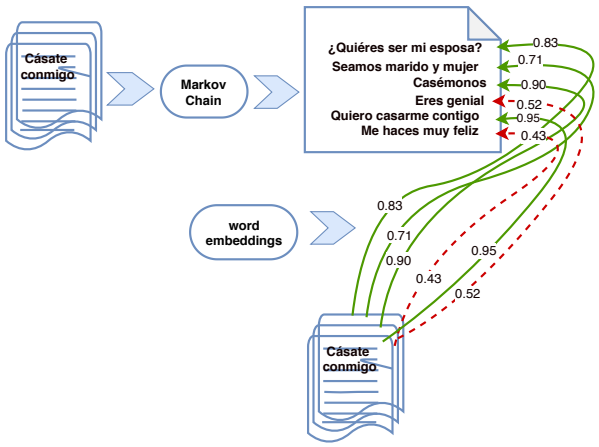

Figure 1: Schema of our generation and filtering method. First, we train a Markov Chain to generate a set of sentences. Then, we use word embeddings to calculate a semantic similarity (values on the arrows) to select the more adequate sentences (not-dashed arrows) and discard the less similar ones (dashed arrows) according to a similarity threshold.

3.1 Sentence generation with Markov Chains

Markov Chains are statistical models well suited for sequence processing. They are useful to compute a probability for a sequence of observable events like words. More formally, a Markov Chain is a probabilistic model that gives information about the probabilities of sequences of random variables or states that can take on values from some set (Jurafsky and Martin, 2008).

These models assume that to predict the future in a sequence all that matters is the current state. Thus, the probability of a state q_t taking on the value a can be expressed as follows[3]:

$$P(q_t = a|q_1 \dots q_{t-1}) = P(q_t = a|q_{t-1})$$

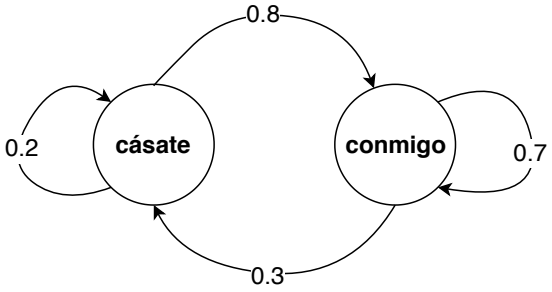

Figure 2: Two-state Markov Chain diagram. The states take values on a vocabulary in the *dating* domain. Each edge expresses the probability of generating a particular word given the preceding one.

For instance, Figure 2 shows how a two-state Markov Chain can model a proposition sentence like *"cásate conmigo"*, which is *"marry me"* in Spanish, as a sequence of words. Note that the edges represent the probabilities of generating *"cásate"* or *"conmigo"* depending on the word generated first.

In our approach, we take advantage of these properties of the Markov Chains to learn the style particularities from a given data set. Then, we generate a new set of sentences using the word probability distribution learned by the MC from the original in-domain corpus.

3.2 Data Filtering with word embeddings

The generated sentences that are more similar to a target domain corpus will help us to obtain more adequate NLP tools or models given a specific domain.

Word embeddings (Mikolov et al., 2013; Peters et al., 2018; Devlin et al., 2019) are distributed word representation models, typically based on neural networks, that are able to capture words' semantic information. These models have proved to be robust and powerful for predicting semantic relations between words and even across languages (Artetxe, Labaka, and Agirre, 2017; Devlin et al., 2019; Ruiter, España-Bonet, and van Genabith, 2019)

[3]This also represents a bigram language model, where the conditional probability of the next word is approximated by using only the conditional probability of the preceding word.

Following a usual approach to work with word embeddings and semantic distance, given a sentence $s = w_1 w_2 \dots w_t$, we define its vector representation as the resulting vector from the average of the vectors for each word in the sentence:

$$\vec{s} = \frac{1}{t} \sum_{1}^{t} \vec{w_i}$$

Then, we use the cosine similarity to measure the semantic relatedness between the generated sentences s_{gen_i} and the sentences in the target domain corpus $S_{tdomain}$:

$$cossim(\vec{s_{gen_i}}, \vec{S_{tdomain}}) = \frac{\vec{s_{gen_i}} \cdot \vec{S_{tdomain}}}{\|\vec{s_{gen_i}}\| \|\vec{S_{tdomain}}\|},$$

being $\vec{S_{tdomain}}$ the vector representation of the target domain data calculated as the average of the vector representation for each sentence in S_{orig}, which are calculated, as before, as the average of the vectors for each word in the sentence. Note that the cosine similarity takes values in $[-1, 1]$, understanding that a higher value will indicate a higher semantic closeness. We discard every s_{gen_i} that is not close enough to the corpus in the target domain. In other words, we only keep those s_{gen_i} that their *cossim* with the vector of the target domain corpus $S_{tdomain}$ is greater than a fixed threshold. This threshold will be set experimentally for each processed corpus.

4 In-Domain Language Models

In order to validate the data generated by applying our approach, we build several LMs: a baseline on a usual subtitles corpus, one on each in-domain original corpus and one on each generated in-domain corpus. Then, we evaluate them by calculating their perplexity on their corresponding in-domain test set.

4.1 Data Generation Settings

We generate new data for three different domains. Each original domain corpus contains a set of utterances[4] gathered from a real running chatbot and reflects different users' interactions. The *dating* domain corpus has sentences that can appear in romantic conversations. The *recipes* domain corpus gathers sentences that express user's recipes preferences. And finally, the *livefb* corpus contains sentences with living queries and mentions to Facebook[5]. We kept 50 sentences from each domain corpora as test set for the LM evaluation we will pursue later on.

threshold	dating	recipes	livefb
original	131	94	89
0.50	78	315	156
0.60	78	315	**156**
0.70	78	315	154
0.75	**78**	**315**	154
0.8	77	310	154
0.85	72	308	154
0.9	69	293	144
0.95	38	250	104
0.98	1	100	26
concat	209	409	245
vocab	192	137	135

Table 1: Number of unique generated sentences for different domains using different thresholds. The *original* row is for the number of unique sentences for each original in-domain corpus. The *concat* is for the number of unique sentences after concatenating the selected generated dataset (in bold) plus the original in-domain corpus. The *vocab* row is for the vocabulary size for each domain data.

We train a two-states MC to generate new sentences with Markovify[6] for each domain. Table 1 shows the number of unique sentences used as MC training set for each domain in the *original* row. In particular, we generate up to 1,000 sentences in our experiments using the MC to filter out afterward the more adequate ones.

For the filtering task, we use the *es_core_news_md*[7] word embedding model available in the *spacy* library[8]. It is a multitask CNN-based word embedding trained on the AnCora (Taulé, Martí, and Recasens, 2008) [9] and WikiNER (Ghaddar and Langlais, 2017) corpora. We carry out a grid

[4]We understand an utterance as a dialog act, in the context of a conversational dialog, that serves a function in the dialog.

[5]www.facebook.com

[6]https://github.com/jsvine/markovify

[7]https://spacy.io/models/es#es_core_news_md

[8]https://spacy.io/

[9]http://clic.ub.edu/corpus/ancora

domain	original sentence	generated sentence
dating	¿Quieres ser mi esposa?	¿Te gustaría ser mi esposa?
recipes	Confío en ti, ¿Me recomiendas la mejor receta?	Confío en ti, ¿Me ayudas a elegir una receta?
livefb	Me dijeron que estás en China, ¿es cierto?	Me dijeron que estás en China, ¿En qué lugar vives?

Table 2: Examples of generated sentences for each of the studied domains in comparison with sentences from the original datasets.

search to adjust the similarity threshold value to keep the more adequate sentences.

Table 1 shows the figures of the different generated data. As long as the similarity threshold increases, the number of filtered sentences decreases as expected. For thresholds below 0.75, the number of generated sentences is the same as per 0.75. For *dating* and *recipes* domains, our method generates the same number of sentences for all thresholds bellow 0.75 whereas for the *livefb* domain it is for threshold 0.60. In particular, we choose these values for the similarity thresholds respectively for each domain, using the resulting generated sentences to build the training corpora by concatenating them to the original in-domain sentences. On the other hand, the number of unique generated sentences by the MC for each domain coincides with the number of sentences indicated in Table 1 for the lower values of the similarity threshold. These facts indicate that the Markov Chain generates sentences that are semantically related to the original domain. This is expected since Markov Chains were built using only these data, sharing then the vocabulary.

Furthermore, it is noticeable that there is only a small overlapping between the original corpus and the generated sentences as shown in the *concat* row in Table 1, that shows the number of unique sentences after concatenating the selected generated dataset plus the original in-domain corpus. These numbers reflect that the generation method is able to propose new adequate sentences. Table 2 shows some examples of generated examples for the different domains.

4.2 Language Models Settings

All the LMs that we built are Transformer-based Language Models trained using Marian (Junczys-Dowmunt et al., 2018) with 128 dimensional embeddings and hidden layers with 256 units. We build a baseline LM using the Spanish side of the English-Spanish OpenSubtitles2018 corpus (Tiedemann, 2012) as training set (61,434,251 sentences), fixing a vocabulary size of 50,000. We kept the last 1,000 sentences as out-domain test set. We chose this corpus to build a reference baseline since it is a collection of movie subtitles and thus they are close to the language particularities of the utterances we want to handle.

4.3 Evaluating the Language Models

We carried out a simple evaluation task. We obtained the perplexities [10] of the different models on their corresponding in-domain test sets of 50 dating utterances each, shown in Table 3.

model	dating	recipes	livefb
baseline	28,37	92.99	52.54
original	28.69	**5.28**	**5.55**
original++	**25.98**	5.76	5.68

Table 3: Perplexity values on the in-domain test sets (the lower the better). The *baseline* row is for the LM trained on OpenSubs2018, the original row is for the LM trained on the original in-domain training dataset and the original++ is for the LM trained using the in-domain corpus including the newly generated sentences using the selected threshold for each domain.

It is easy to observe the importance of having in-domain data. Recall that the baseline LM was trained on millions of sentences from subtitles whereas the in-domain LMs were trained using only hundreds of sentences. The LM trained only using the in-domain corpus achieves almost the same perplexity values as the baseline LM for the *dating* domain. Whereas for the other two domains, the LMs trained using only the original in-domain data highly improve the perplexity

[10] The more information an LM gives about a word sequence the lower the perplexity. Better LMs can help to select a more adequate answer in a chatbot workflow.

values. A possible reason for that could be the specificity of the corpora for these two domains in comparison with the data in the *dating* domain. More open domains have larger vocabularies and a higher variability margin that results in obtaining LMs with lower perplexities. In our case, *livefb* and *recipes* domains have fewer data and smaller vocabularies. Thus, it is easier to obtain lower perplexities in these domains than in larger domains like for the *dating* case.

The model trained on the *dating* extended corpora achieves better perplexities than the LM on the *dating* corpus, also better than the baseline LM. For the *recipes* and *livefb* domains, the LMs trained on the extended corpora achieve a similar perplexity than the ones for the LMs trained on the original in-domain data. Note that the baseline LM also achieves the worst perplexities on the *recipes* and the *livefb* test set. These results support also the fact that the *dating* domain data represents a more open domain than the *recipes* and *livefb* ones. Thus, being easier to improve the results achieved using only the original *dating* dataset than in the other two scenarios. Therefore, the results clearly show the importance of the adequacy of the data regarding a specific domain. Furthermore, the numbers also indicate the impact of the specificity of a domain, being more necessary to generate data for more open domains than for the more specific ones.

The best LM in the *dating* domain, the more open one, is the model trained on all the generated sentences, getting a 2.71 points better perplexity. This shows the usefulness of the sentences generated by our method even though having a small original in-domain corpus as a starting point.

5 Conclusions

We propose a light hybrid method to generate extra data to extend a corpus for a specific domain. Our approach is simple yet effective, and it does not need a large amount of data. Our method comprises two phases: first, it uses a Markov Chain to generate sentences. Then, it filters the most similar sentences according to the cosine similarity of their vector representation. The generation method is able to create a significant amount of new sentences with a small overlapping with the original in-domain corpus.

We assess the validity of the generated data by evaluating a set of in-domain LMs trained using a corpus extended with the data generated by applying our method. We found out that our method works well when dealing with data from more open domains. The LMs trained for the *dating* domain, using the data generated by our approach, show the highest quality gain in terms of perplexity. In contrast, the impact of the generated data for LM models on more specific domains, like the *recipes* and *livefb* ones, is not as noticeable since it is more difficult to achieve lower perplexities in this kind of scenario because they are more predictable.

As future work, we want to make a better evaluation of our method using more data, both for training and testing, as soon as they are available. Performing also an external evaluation of the LMs trained using the data generated by our method by including them in a reranking procedure for generating the answer of a chatbot.

We are interested in exploring variations of our method that can lead to quality improvements. Improve sentence representations by using sentence embeddings (Reimers and Gurevych, 2019), (Le and Mikolov, 2014). Generate better utterance candidates by using trigram or 4-gram CMS or even using neural-based generative approaches (Puzikov and Gurevych, 2018; Liu et al., 2018; Bahdanau, Cho, and Bengio, 2015). Also, we want to refine our sentence filtering approach by using other similarity measures like some margin-based scores (Artetxe and Schwenk, 2019) or the CSLS (cross-domain similarity local scaling) (Lample et al., 2018).

Furthermore, we would like to study the impact of using sentences generated using our method to fine-tune the newest word representation models, like BERT (Devlin et al., 2019), ELMO (Peters et al., 2018) or XLNET (Yang et al., 2019), for the language modeling task.

Acknowledgments

We would like to thank Francisco del Valle Bas, Ángel Melchor and Miguel Pajares for their assistance during the development of this research work.

References

Artetxe, M., G. Labaka, and E. Agirre. 2017. Learning bilingual word embeddings with

(almost) no bilingual data. In *Proceedings of the ACL2017 (Volume 1: Long Papers)*, pages 451–462.

Artetxe, M. and H. Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the ACL2019 – Volume 1*, pages 3197–3203.

Bahdanau, D., K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR 2015*.

Chan, W., N. Jaitly, Q. Le, and O. Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of IEEE ICASSP 2016*, pages 4960–4964.

Devlin, J., M. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the NAACL-HLT 2019*, pages 4171–4186.

Dušek, O. and F. Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the ACL2016 (Volume 2: Short Papers)*, pages 45–51.

Freitag, M. and S. Roy. 2018. Unsupervised natural language generation with denoising autoencoders. In *Proceedings of the EMNLP 2018*, pages 3922–3929, October-November.

Gagniuc, P. A. 2017. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons.

Ghaddar, A. and P. Langlais. 2017. WiNER: A Wikipedia annotated corpus for named entity recognition. In *Proceedings of the IJCNLP 2017(Volume 1: Long Papers)*, pages 413–422.

Inaba, M. and K. Takahashi. 2016. Neural utterance ranking model for conversational dialogue systems. In *Proceedings of the SIGDIAL 2016*, pages 393–403.

Inoue, H. 2018. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*.

Junczys-Dowmunt, M. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the WMT19 Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233.

Junczys-Dowmunt, M., R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Jurafsky, D. and J. H. Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Nueva Jersey: Prentice Hall.

Lample, G., A. Conneau, M. Ranzato, L. Denoyer, and H. Jégou. 2018. Word translation without parallel data. In *Proceedings of the ICLR 2018*.

Le, Q. and T. Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Liu, T., K. Wang, L. Sha, B. Chang, and Z. Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *32nd AAAI Conference on Artificial Intelligence*.

Mikolov, T., K. Chen, G. S. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Moritz, N., T. Hori, and J. L. Roux. 2019. Unidirectional Neural Network Architectures for End-to-End Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 76–80.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings NAACL-HLT 2018 – Volume 1*, pages 2227–2237.

Pham, N.-Q., T.-S. Nguyen, J. Niehues, M. Müller, and A. Waibel. 2019. Very Deep Self-Attention Networks for End-to-End Speech Recognition. In *Proc. Interspeech 2019*, pages 66–70.

Puzikov, Y. and I. Gurevych. 2018. E2E NLG challenge: Neural models vs. templates. In *Proceedings of the INLG 2018*, pages 463–471.

Reimers, N. and I. Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the EMNLP-IJCNLP 2019*, pages 3982–3992.

Ruiter, D., C. España-Bonet, and J. van Genabith. 2019. Self-Supervised Neural Machine Translation. In *Proceedings of the ACL 2019, Volume 2: Short Papers.*, pages 1828–1834.

Sankar, C., S. Subramanian, C. Pal, S. Chandar, and Y. Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the ACL2019*, pages 32–37.

Sennrich, R., B. Haddow, and A. Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the ACL2016 (Volume 1: Long Papers)*, pages 86–96.

Serban, I. V., A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *30th AAAI Conference on Artificial Intelligence*.

Sordoni, A., M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the NACCL-HLT 2015*, pages 196–205.

Tanner, M. A. and W. H. Wong. 1987. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540.

Taulé, M., M. A. Martí, and M. Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the LREC'08*.

Tiedemann, J. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the LREC2012*, pages 2214–2218.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Proceedings of the NIPS2017*, pages 6000–6010.

Vinyals, O. and Q. V. Le. 2015. A neural conversational model. In *Proceedings of the 31 st International Conference on Machine Learning*, volume 37.

Wen, T.-H., M. Gašić, D. Kim, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *Proceedings of the SIGDIAL2015*, pages 275–284.

Yang, Z., Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *ArXiv*, abs/1906.08237.