Experiments in training transformer sequence-to-sequence DRS parsers

Ahmet Yıldırım and Dag Trygve Truslew Haug
Department of Linguistics and Scandinavian Studies
University of Oslo
{ahmetyi,daghaug}@uio.no

Abstract

This work experiments with various configurations of transformer-based sequence-to-sequence neural networks in training a Discourse Representation Structure (DRS) parser, and presents the results along with the code to reproduce our experiments for use by the community working on DRS parsing. These are configurations that have not been tested in prior work on this task. The Parallel Meaning Bank (PMB) English data sets are used to train the models. The results are evaluated on the PMB test sets using Counter, the standard evaluation tool for DRSs. We show that the performance improves upon the previous state of the art by 0.5 (F_1%) for PMB 2.2.0 and 1.02 (F_1%) for PMB 3.0.0 test sets. We also present results on PMB 4.0.0, which has not been evaluated using Counter in previous research.

1 Introduction

Discourse representation structures (DRSs) are a way of representing meaning based on Discourse Representation Theory (Kamp and Reyle, 1993; Kamp et al., 2011). In addition to predicate-argument structures, DRSs express temporal relations, anaphora, modals, negation, and presuppositions, and can be further employed by other automatic processes to understand natural language.

The task of mapping sentences to their DRS meaning representations is called DRS parsing. There now exists a large dataset with DRSs for corpus examples, the Groningen Parallel Meaning Bank (PMB, Abzianidze et al. 2017), which makes it possible to train deep neural networks of the kinds that provide state-of-the-art performance on a variety of NLP tasks these days.

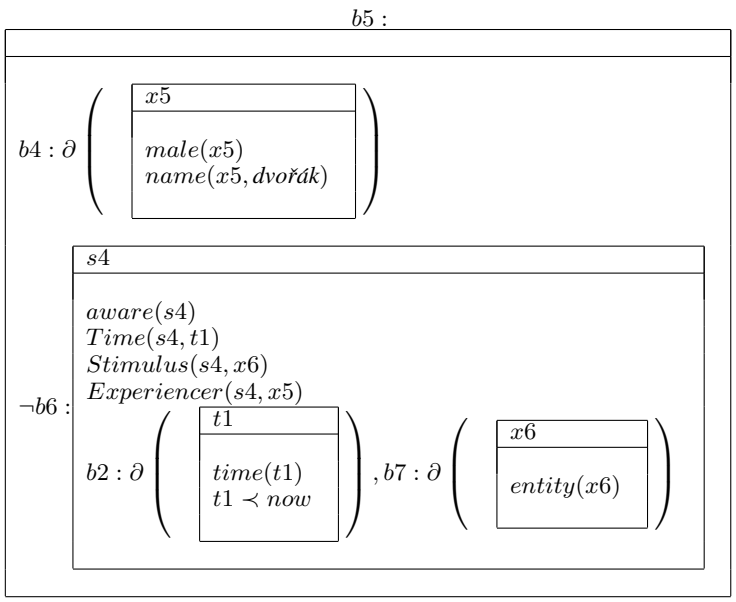
Recent work has explored a variety of neural network architectures for this task, but curiously, little work has been done using the otherwise widely utilized transformer-based encoder-decoder architecture. In this paper, we report on such experiments using Wordpiece (Wu et al., 2016) to tokenize the input and output, and train a sequence-to-sequence model where the encoder is a pre-trained BERT model (Devlin et al., 2018) and the decoder consists of randomly initialized transformer layers with cross attention. We experiment with different hyperparameter settings and achieve higher performance than in previous work.

In the remainder of this paper, we briefly introduce DRSs and the PMB dataset in Section 2. We then survey previous work on DRS parsing in Section 3. Section 4 provides the machine learning configurations we used. Section 5 presents the results and a comparison with prior work. Section 6 remarks on our overall takeaways from this work.

2 Data

Historically, DRSs are represented in a box notation designed for human readability. The left-hand side of Figure 1 shows the representation of *Dvořák was not aware of it*. The negated content *was not aware of it* is represented as a separate embedded box labeled $b6$. Moreover, the sentence contains three presuppositions that must be resolved: these are the boxes $b2$, $b4$, $b7$ (shown inside a presupposition operator ∂), corresponding to the referents $t1$ (time at which the sentence holds), $x5$ (the referent of the proper name *Dvořák*), and $x6$ (the entity to which *it* refers). The latter two referents appear inside the negated box, because they are syntactically in the scope of negation, but they must in fact be interpreted in a wider context (i.e. the text entails that there exists a reference for *it* and a time at which the state of Dvořák not being aware of it held). For more details about DRSs, we refer to (Kamp and Reyle, 1993; Kamp et al., 2011).

The release of the PMB offered for the first time relatively large amounts of text annotated with deep



b4 REF x5
b4 Name x5 "dvořák"
b4 PRESUPPOSITION b5
b4 male "n.02" x5
b2 PRESUPPOSITION b6
b6 Time s4 t1
b2 REF t1
b2 TPR t1 "now"
b2 time "n.08" t1
b5 NEGATION b6
b6 REF s4
b6 Experiencer s4 x5
b6 aware "a.01" s4
b6 Stimulus s4 x6
b7 REF x6
b7 PRESUPPOSITION b6
b7 entity "n.01" x6

Figure 1: Box and clause notation of the DRS for *Dvořák was not aware of it*

semantic representations in the form of DRSs. In PMB, DRSs are given in a more machine-friendly clause format shown on the right-hand side of Figure 1. We refer to Liu et al. (2021) for more details on the conversion. Notice that the clause format also contains references to WordNet synsets ("n.02" etc.). Parsing to PMB representations therefore also involves word sense disambiguation.

There are several releases of the PMB, differing in size and also in some choices of representation. Previous work has focused on version 2.2.0, which contains 5929 DRSs for English sentences, and version 3.0.0, which has 8403 English DRSs. The latest release, version 4.0.0, has 10715 English DRSs. All versions also have data in Dutch, German, and Italian, which we ignore here. Each release has various data files available at the website (Parallel Meaning Bank, 2020), but also provides a separate download that contains only the data relevant for experiments in semantic parsing ("exp_data").

The annotations are done automatically and then manually corrected. The representations are labeled with bronze, silver, or gold status. Bronze sentences have no manual correction, silver sentences have a partial manual correction and gold sentences have a full manual correction. The dev, test, and eval datasets consist of gold sentences only.

3 Related work

Before the advancement of machine learning systems, rule-based approaches were proposed as a solution for the DRS parsing task. Work within this research track mainly tried to resolve anaphora (Johnson and Klein, 1986; Wada and Asher, 1986), scope ambiguities, and presuppositions (Bos, 2001) on short English text. Later, the Boxer Software (Bos, 2008) used syntactic parses from a Combinatory Categorial Grammar (Clark and Curran, 2004) to produce DRSs. In another line of work, DRSs were represented as graphs obtained from dependency structures of sentences (Le and Zuidema, 2012) and ranked according to their probabilities of representing the sentence where the probabilities are obtained from a corpus by computing word-to-word alignments using an external tool (Och and Ney, 2003).

System	Model	Input
Liu et al. (2019)	transformer	characters
van Noord et al. (2018)	seq2seq	characters
van Noord (2019)	seq2seq	characters
Evang (2019)	stack LSTMs	word embeddings
Fancellu et al.[1]	bi-LSTM	word embeddings

Table 1: Systems in the shared task on DRS parsing

With the advent of language models and a data set like the PMB which is large enough for fine-tuning such models, it became possible to employ neural nets for DRS parsing. All systems in the recent shared task on DRS parsing (Abzianidze et al., 2019b) used neural architectures, as shown in Table 1 adapted from Abzianidze et al. (2019a).

Most systems used a variety of a sequence-to-

[1]No system description was submitted to the proceedings but the system is described in Abzianidze et al. (2019a).

sequence LSTM (Hochreiter and Schmidhuber, 1997), though Liu et al. (2019) used a transformer model (Vaswani et al., 2017). The system input was either character-level representations or word embeddings obtained from one of the widely utilized BERT language models. Later, van Noord et al. (2020) combined these two inputs to their LSTM sequence-to-sequence system, which also used an attention mechanism (Vaswani et al., 2017), arguing that this improved results even when added to the rich BERT embeddings. They also report results using a transformer model but were unable to beat the LSTM sequence-to-sequence model in this way. Their work reported state-of-the-art results for PMB 2.2.0 and PMB 3.0.0 English datasets. Later, Liu et al. (2021) used BERT word embeddings and position embeddings as input and expression of DRSs as trees as output to train a transformer sequence-to-sequence model. They reported a slight improvement (0.4%) upon the state of the art for PMB 2.2.0 dataset. As far as we know these are the only attempts at using the transformers architecture which is the default approach across many NLP tasks today.

4 Machine learning configurations

We use sequence-to-sequence modeling with two main components: an encoder and a decoder. HuggingFace transformers library (Wolf et al., 2020) provides the class EncoderDecoderModel to configure such models. The models are trained with various configurations of this class to test the performance.[2] For the encoder side, 7 configurations are tested. The first two options test different sizes of random initialization (No-PT). One configuration is 6 layers of 768 hidden layer size (No-PT, 6x768), and the other is 8 layers of 512 hidden layer size (No-PT, 8x512). The rest of the encoders are pre-trained models: bert_base_cased, bert_base_uncased, bert_large_cased, and bert_large_uncased. For the decoder side, we use the size of 6x768 with the 6x768 sized No-PT encoder, and 8x512 with both a No-PT encoder setup and the pre-trained encoders. All decoder side weights are randomly initialized. The 12x768 networks have 12 and 8x512 networks have 8 attention heads per layer. For the 6-layer setups, two configurations are used: 6 and 12 attention heads per layer. All decoders include cross-attention layers as it is effective in sequence-to-sequence training (Gheini et al., 2021).

The work by van Noord et al. (2020) reports that updating pre-trained encoder weights always resulted in poor performance. Therefore, a similar approach is followed and the encoder side weights are frozen whenever we use the pre-trained encoders. When the 12x768 decoder is used with No-PT encoders, the number of parameters to be trained gets too high and a model cannot be trained. Thus, the 12x768 decoder configuration is only used together with the frozen pre-trained encoders.

Our configurations get inputs as sub-word tokens derived from the widely utilized Wordpiece tokenizer (Wu et al., 2016). With the pre-trained encoders, the tokenizer used to train that pre-trained model is used as the input tokenizer. For No-PT encoders and for the decoder side output, we train custom Wordpiece tokenizers for each dataset. Since the output of DRS parsing is a DRS, the serializations of DRSs are tokenized using the relative clause notation introduced in van Noord (2021). All custom tokenizers are trained with: a vocabulary size of 25000, the minimum frequency for consideration of a token is set to 3, and the maximum tokenization length (maximum number of tokens for one sentence) is set to 512 tokens.

To test the effect of using different parameters introduced in this section, the other hyperparameters are fixed such as the optimizer, learning rate, and loss function. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0001, and use the negative log-likelihood loss (Yao et al., 2020) to compute the loss in each batch between the model output and the expected output. We set the batch size to 16 sentences as this is the amount the graphic cards could handle. For any other parameter, the default value defined by version 4.17.0 of the Transformers library is used for the objects of types BertConfig, EncoderDecoderModel, EncoderDecoderConfig, and BertModel.

We use four Nvidia V100 32GB GPUs to train the models. The training time depends on the number of parameters and the number of attention heads. For one configuration, training for PMB 2.2.0 sets takes around one day, and training for PMB 3.0.0 and PMB 4.0.0 sets takes around 2 days. When four GPUs are used, it takes around one week to train all models in all configurations. We train for each configuration only once.

[2]The replication code is published under GitHub: https://github.com/textlab/seq2seqDRSparser

number of parameters	Encoder	Decoder	PMB 2.2.0 dev	PMB 2.2.0 test	PMB 3.0.0 dev	PMB 3.0.0 test	PMB 4.0.0 dev	PMB 4.0.0 test	PMB 4.0.0 eval
139,636,648	No-PT, 6x768-6	6x768-6	86.65	87.65	89.78	89.16	89.05	89.48	87.32
139,636,648	No-PT, 6x768-12	6x768-12	86.45	87.8	89.64	89.48	89.03	89.45	87.51
102,389,160	No-PT, 8x512-8	8x512-8	86.87	87.26	89.46	89.64	89.06	89.61	87.38
55,775,144	bert_base_uncased	8x512-8	87.17	88.45	89.69	89.78	89.1	89.79	87.2
55,775,144	bert_base_cased	8x512-8	87.51	88.23	**89.96**	89.89	89.19	89.9	**88.18**
133,633,960	bert_base_uncased	12x768-12	87.41	88.18	89.57	89.66	**89.4**	**90.26**	87.36
133,633,960	bert_base_cased	12x768-12	**87.53**	**89.23**	89.78	**90.32**	88.07	89.04	86.9
134,421,160	bert_large_uncased	12x768-12	86.93	88.56	89.08	88.65	88.71	89.6	87.29
134,421,160	bert_large_cased	12x768-12	86.9	88.27	89.39	90.03	88.81	90.12	87.42
≈106 million	van Noord et al. (2020)		86.1	88.3	88.4	89.3			
≈106 million	Liu et al. (2021)			88.7					

Table 2: F1% scores of various models. Prior works by van Noord et al. (2020) and Liu et al. (2021) use similar hyperparameter settings. No-PT: No pre-training. AxB-C: A hidden layers of size B and C attention heads per layer.

5 Results

The performance scores are computed for *dev*, *test*, and *eval*[3] sets for each dataset. To compute the scores, we used the Counter tool provided by van Noord (2022). To make the results comparable with the previous work, the version of Counter with the same version tag for each release of the datasets is used. For the 4.0.0 release of the datasets, we use the latest version of the code as 4.0.0 is the newest release. The models are trained for at least 80 epochs for all datasets and stopped if there is no increase in performance for the last five epochs. Table 2 presents the results obtained for the configurations mentioned in the previous section.

Previous work used gold and silver data for fine-tuning. Our work uses the train sets as is and does not prioritize gold, silver, or bronze sentences. Therefore, one training epoch consists of using each sentence only once, and, the learning rate is not changed throughout the training. Even with this setup, we observe that two configurations with randomly initialized encoders and decoders (No-PT) outperform the previous state of the art for PMB 3.0.0. Moreover, using pre-trained encoders performed even better. For the PMB 2.2.0 test set, our setup slightly improved upon the previous state-of-the-art. For PMB 4.0.0, to the best of our knowledge, this is the first time model performances are reported using Counter.[4] For all configurations, using the larger BERT pre-trained models bert_large cased and uncased do not perform better than the smaller bert_base cased and uncased. We observe that using cased pre-trained models generally performed better.

Table 3 presents detailed performances for different kinds of DRS clauses in the clause notation. The results are in line with what van Noord et al. (2020, Table 10) report. *DRS operators* have the highest performance which indicates that structural features of a DRS is captured better than the other features. One reason may be that the test set of all releases of PMB represent relatively short sentences that have structurally simple DRSs. Roles (i.e. binary predicates like `Agent`, `Theme`, `MadeOf` etc.) and concepts (which includes word sense disambiguation because each concept is a WordNet synset) are harder to capture, especially verbal concepts. Performance for adjective and adverbs increase with each release of the datasets, probably reflecting improving standards of annotation.

van Noord et al. (2020) observe that parsing performance decreases with sentence length. In Haug et al. (2023) we show that the same holds for our system. Nevertheless, the PMB test set with its uniformly quite short sentences (the large majority is ¡ 10 tokens) does not lend itself to study the effect of sentence length, and in Haug et al. (2023) we test the system on more realistic sentence lengths.

6 Conclusions

Our work presents the effect of using various sizes of transformer-based encoders and decoders in sequence-to-sequence neural networks with the

[3] PMB publishes the *eval* dataset only for the 4.0.0 release

[4] Poelman et al. (2022) reports using the SMATCH (Cai and Knight, 2013) tool by comparing Discourse Representation Graphs (DRG), a simpler form of DRSs, on PMB 4.0.0.

	PMB 2.2.0	PMB 3.0.0	PMB 4.0.0
Operators	95.58	96.55	95.78
Roles	88.2	89.88	89.01
Concepts	85.35	86.99	87.95
Nouns	90.68	91.35	92.28
Verbs	73.45	75.81	73.83
Adjectives	67.43	78.98	82.53
Adverbs	50.0	73.85	85.5
Events	72.37	76.46	75.96

Table 3: F1% scores in different PMB versions' test sets for different types of DRS clauses in the clause notation. The configuration is bert_base_cased encoder with 12x768-12 decoder that is trained for each PMB version separately.

subword tokenizer Wordpiece on the task of DRS parsing. The performances of the use of various sizes and pre-trained encoder configurations are reported. This work shows that the performance of DRS parsing increases with some of these configurations. We believe that applying our setup could improve the performance of other related tasks. For example, Liu et al. (2021) explores multilingual DRS parsing based on transfer from English translations which, as we have shown here, could be better parsed with our approach.

Our results provide a new state-of-the-art of what can be achieved in a vanilla setup of transformer networks with raw text input and clause format DRS output. While it is likely that the results can be improved with better language models, or by fine-tuning strategies similar to those of van Noord et al. (2020) (prioritizing gold data over silver and bronze), we think more substantial improvements can come from working on the input and output representations. On the output side, we plan to experiment with other ways of expressing DRSs such as the format introduced by Liu et al. (2021). On the input side, we believe that syntactic dependency parses contain much information that is useful to DRS parsing, such as predicate argument structures. We are currently experimenting with rule-based extraction of relevant information from UD trees and ways of adding this information to the input.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos. 2019a. The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos, editors. 2019b. *Proceedings of the IWCS Shared Task on Semantic Parsing*. Association for Computational Linguistics, Gothenburg, Sweden.

Johan Bos. 2001. Doris 2001: Underspecification, resolution and inference for discourse representation structures. *ICoS-3, Inference in Computational Semantics*, pages 117–124.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 277–286. College Publications.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 103–es, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Kilian Evang. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained Transformers for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1765, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dag Trygve Truslew Haug, Jamie Y. Findlay, and Ahmet Yıldırım. 2023. Ethe long and the short of it: DRASTIC, a semantically annotated dataset containing sentences of more natural length. In *Proceedings of the 4th International Workshop on Designing Meaning Representation(DMR 2023)*, Nancy, France. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mark Johnson and Ewan Klein. 1986. Discourse, anaphora and parsing. In *Coling 1986 Volume 1: The 11th International Conference on Computational Linguistics*.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer Academic Publishers.

Hans Kamp, Josef van Genabith, and Uwe Reyle. 2011. Discourse Representation Theory. In D. M. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 15, pages 125–394. Springer.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Phong Le and Willem Zuidema. 2012. Learning compositional semantics for open domain semantic parsing. In *Proceedings of COLING 2012*, pages 1535–1552, Mumbai, India. The COLING 2012 Organizing Committee.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019. Discourse representation structure parsing with recurrent neural networks and the transformer model. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, Mirella Lapata, and Johan Bos. 2021. Universal discourse representation structure parsing. *Computational Linguistics*, 47(2):445–476.

Ri van Noord. 2019. Neural boxer at the IWCS shared task on DRS parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Rik van Noord. 2022. Rikvn/drs_parsing: Scripts to evaluate scoped meaning representations. https://github.com/RikVN/DRS_parsing. Accessed: 2022-07-19.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring Neural Methods for Parsing Discourse Representation Structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. Character-level representations improve DRS-based semantic parsing even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51.

Parallel Meaning Bank. 2020. Index of /releases. https://pmb.let.rug.nl/releases/. Accessed: 2022-07-19.

Wessel Poelman, Rik van Noord, and Johan Bos. 2022. Transparent semantic parsing with Universal Dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4186–4192, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Rik van Noord. 2021. *Character-based Neural Semantic Parsing*. Ph.D. thesis, University of Groningen.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hajime Wada and Nicholas Asher. 1986. BUILDRS: An implementation of DR theory and LFG. In *Proceedings of the 11th Coference on Computational Linguistics*, COLING '86, page 540–545, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Hengshuai Yao, Dong-lai Zhu, Bei Jiang, and Peng Yu. 2020. Negative log likelihood ratio loss for deep neural network classification. In *Proceedings of the Future Technologies Conference (FTC) 2019*, pages 276–282. Springer International Publishing.