Machine Translation Evaluation Using Semantic Structures and Language-Aligned Sentence Embeddings

1st Semester of 2021-2022

Ioan-Bogdan Iordache
ioan.iordache@s.unibuc.ro

Teodor-Florin Manghiuc
teodor.manghiuc@s.unibuc.ro

Abstract

In this paper we look at two different approaches for implementing automated metrics for machine translation. The first one involves incorporating semantic information to traditional metrics (like BLEU, METEOR, CHRF) through Universal Conceptual Cognitive Annotations of reference and system translations. The second one is directed towards the use of pre-trained monolingual and multilingual models. We also try to exemplify how such models can be used to assess the quality of translation systems without the need of references.

1 Introduction

Automatic metrics are an essential part of Machine Translation tasks. Most of the time, they are used to declare the superiority of one system over another. What metrics are chosen by the community may guide the direction in which research develops. Because of that, it is of paramount importance to find metrics that correlate as much as possible with human judgment.

Building parallel datasets for machine translation is hard, but also assessing the performance of multiple translation systems on these datasets, using human evaluators, is even harder. The amount of manual work needed for such a task increases linearly with the number of systems that are evaluated. This is partly why the Conference on Machine Translation has initiated the "Metrics Shared Task", where they provide datasets containing human evaluation for the outputs of participating systems in the main translation tasks. Using such datasets, teams compete each year on developing automatic metrics for evaluating machine translation, that correlate as much as possible with the human scores.

Some traditional metrics (that we will also use in our experiments as baselines) are:

- BLEU (Papineni et al., 2002): computed as the precision of word n-grams of the output generated by a machine translation system, compared to the reference translation; in order to punish overly short translations, a brevity penalty is added to the metric computation.
- CHRF (Popović, 2015): similar definition to BLEU, with the exception that character n-grams are counted instead of word n-grams.
- METEOR (Lavie and Agarwal, 2007): computes the harmonic mean of precision and recall (with recall weighing higher) determined for uni-gram occurrences in the translation output and the reference. Its unique aspect compared to other metrics is the incorporation of stemming and synonymy when comparing various word occurrences.

Designing an automatic metric comes with two challenges. The first one is obviously linked to its correlation to human judgment. The second challenge is the computational one, traditional metrics have low costs in terms of time and resources needed for evaluation. Highest correlation scores were usually obtained using deep learning approaches, fine-tuning large scale models using human judgment annotations (Thompson and Post, 2020; Rei et al., 2021; Sellam et al., 2020), but competitive metrics that incorporate less expensive computations were also submitted (usually building on top of traditional metrics or incorporating additional features) (Wang et al., 2016; Xu et al., 2020; Mukherjee et al., 2020).

Our approach involves designing automated metrics for two language pairs (English-German and German-English), and it is two-fold:

- firstly, we start from the technique described by (Xu et al., 2020): semantic features are computed by finding the overlap of words that carry important semantic meanings (core words) between output and reference translation. Core words are extracted using a semantic representation framework, called Universal Conceptual Cognitive Annotation. We further build on top of this approach by incorporating synonymy information into the computation.

- secondly, we explore siamese neural networks based on pre-trained transformer architectures, used to extract sentence embeddings that incorporate contextual and semantic information. The embedding generator models are fine-tuned in order to provide similar embeddings for good output-reference pairs and distant ones for bad translations. We also experiment with using multilingual models to evaluate translation quality using only the output translation and the source text.

2 Dataset

We used the dataset provided for the metrics shared task at the 2020 Conference on Machine Translation (Mathur et al., 2020). Human annotators scored the translations of various translation systems on the news translation shared task.

The sources, references and system outputs for each language pair are organized in a hierarchical structure. Each text file is split into larger units (called documents) containing various numbers of segments. Segments are the smallest unit of textual information (sentences or small groups of sentences), while the documents encompass related segments extracted from the same source.

As stated previously, we look at two language pairs (English-to-German and German-to-English).

For the **English-German** translation pair, a total of $11,360$ of segment translation pairs were scored by humans (these represent pairs of segments in the source language and their translation by a competing system). These segments are coming from a total of 130 documents, amounting for about 47% of the total number of segments from the news translation task for this language pair. The translations come from 17 systems. For each source segment, up to 3 reference translations are provided.

For the **German-English** pair, $9,389$ human-scored segment pairs are available, coming from 118 documents and amounting for 92% of the total number of segments from the news translation task. The outputs of 13 systems were scored, while for each segment up to 2 reference translations are available.

Using this data, we build our datasets by collecting segment triplets containing the source text of a segment, a reference translation of the segment and a system translation. These triplets are scored using the human annotations and are used in our experiments. The human score assigned for a translation does not depend on the reference translation, so we will consider it the same regardless the reference. We thus obtain $34,080$ triplets for English-German and $18,778$ triplets for German-English.

3 Incorporating Semantic Information from UCCA

Our first approach involves extracting semantic structural information from Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013). UCCA provides multi-layered semantic representations of text by building directed acyclic graphs with the words of the text being represented as leaves.

UCCA is appealing for extracting these kind of features since it includes a lot of fundamental semantic phenomena such as verbal, nominal and adjectival argument structures and the relationships between them. The graph structure is anchored in the words of the text (the terminal vertices), while non-terminal nodes correspond to the combination of meanings of their child nodes. Directed edges connecting parents to children mark the semantic role of the children in the overall meaning of the parent node.

An essential concept of UCCA is the notion of scenes. Scenes describe moves, actions or states in a sentence. Scene nodes may be directly connected to the root of the representation or they may be embedded in other scenes. Figure 1 contains the UCCA representation of the sentence "John and Mary bought the sofa I sold together". We can see that in this sentence there are two scenes: the first corresponding to the whole sentence, and the other embedded in the first corresponding to the secondary action of selling the sofa ("I sold"). Scene nodes are easy to recognize, since a scene has one main relation for *Process* or *State*, along with other argument relations.

The intuition behind this approach is the fact that words carrying high semantic information should be translated correctly by systems. Using this idea, we can compute the overlap of such words between the reference and the system output.

Going forward we will use the same terminology as the authors of this initial approach (Xu et al., 2020), and call these semantically important words

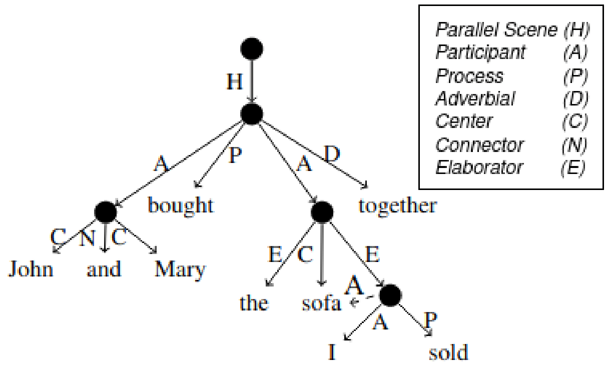

Figure 1: UCCA graph example. Dashed lines correspond to secondary semantic relations. (Xu et al., 2020)

as **core words**. Core words are identified in the graph by the relation preceding their corresponding leaf node. For this, we take into account *Process*, *State*, *Participant*, and *Center* relations.

In order to compute the metric for a segment translation we look at the reference and the system output. UCCA representations are generated for these segments using TUPA (Hershcovich et al., 2017, 2018) and the sets of core words are then computed. The overlap of core words is computed as an F1 score, or a harmonic mean of the precision and recall of the core words that appear in the system output compared to the reference. The matching of core words is determined by checking for identical stems. Our addition to this process is further checking for matches based on synonymy relationships (i.e. core words can math if they are synonyms).

For stemming we employed a Snowball stemmer (Porter, 2001) from the NLTK Python library (Loper and Bird, 2002), while for synonymy we checked for common synsets as defined in the English WordNet (Miller, 1995), while for German we used an online dictionary of synonyms[1].

Four penalty coefficients are computed along with the core words overlap:

- the ratio between the number of scenes in the reference and in the system output representations (P_1)
- the ratio between counts of nodes in the two representations (P_2)
- ratio between number of edges corresponding to critical semantic roles, i.e. *State*, *Process*, *Participant* (P_3)
- the arithmetic mean between number of words from the two representations (P_4).

The final formula for the defined metric is:

$$S = F_1 \cdot exp(-\alpha_1 \cdot P_1 - \alpha_2 \cdot P_2 - \alpha_3 \cdot P_3 - \alpha_4 \cdot P_4)$$

Where α_1, α_2, α_3, and α_4 are tunable coefficients. We will further assess the performance of this metric when combined with the baseline metric scores mentioned in the introduction. Combining a baseline metric with this score can be done using another tunable hyperparameter:

$$Baseline\ score + \beta \cdot S$$

The best coefficients are found by splitting the dataset into a 50% train split and a 50% test split. This operation is done by making sure that all segments coming from the same document are in the same split. The coefficients are chosen such that the Pearson correlation between the computed scores and the human scores is as high as possible for the training dataset. Table 1 contains the comparison between the baseline metrics and their augmentation with the proposed method. Baseline metrics are computed using the SacreBLEU (Post, 2018) implementations of BLEU and CHRF, and the METEOR implementation from NLTK.

Metric	**De → En**	**En → De**
BLEU	0.3473	0.1601
BLEU + UCCA	0.3482	0.1603
BLEU + UCCA + SYN.	**0.3484**	**0.1604**
METEOR	0.5394	0.1719
METEOR + UCCA	0.5436	0.1782
METEOR + UCCA + SYN.	**0.5475**	**0.1784**
CHRF	0.5135	0.2036
CHRF + UCCA	0.5137	0.2037
CHRF + UCCA + SYN.	**0.5141**	**0.2037**

Table 1: Segment-level Pearson correlation computed on the test datasets of our targeted language pairs. We compare the performance of baseline metrics, baselines augmented with the UCCA-based scores, and baselines augmented with UCCA-based scores computed with synonymy-sensitive matches.

We can see that some improvements in correlation scores emerge from augmenting the traditional baseline metrics with the proposed semantic-based scores. In table 2 we provide the best hyperparameters found for each corresponding baseline metric. These coefficients were found by random searching inside pre-defined intervals for all 5 of them.

4 Transformer-based Sentence Encoders

Our second approach is based on using transformer-based models (Vaswani et al., 2017) pre-trained on

[1] https://pypi.org/project/PyMultiDictionary/

	BLEU	METEOR	CHRF
α_1	0.1	0.2	0.1
α_2	0.8	0.5	0.5
α_3	0.2	0.3	0.3
α_4	0.01	0.01	0.01
β	0.5	0.3	0.4

Table 2: List of best coefficients for computing UCCA-based scores and augmenting each of the baseline metrics. These parameters where tuned using the training splits.

large monolingual and multilingual corpora.

The used models are from the BERT family of architectures (Devlin et al., 2018). These models are large-scale transformer encoders trained in a self-supervised manner on large text corpora, using techniques such as masked language modeling and next sentence prediction. The common method is to take these pre-trained architectures and fine-tune them on downstream tasks.

For classification/regression the common practice is to use the output embedding generated by the encoder for the initial token (usually denoted as `[CLS]`). Without pre-training though this embedding is not that useful, being only trained for the task of next sentence prediction. That is why, in order to obtain embeddings for a whole sentence/sequence it is better to take the mean of the embeddings generated for all tokens in the sequence.

The method we employed is inspired by the strategy defined by (Reimers and Gurevych, 2019) for textual semantic similarity. For a language pair, we take the dataset of scored segment translations and use the references and system outputs to train a BERT-like encoder. The training is done in a siamese network setting, by encoding separately pairs of reference and output segments and then computing the cosine similarity between them (see figure 2), this similarity is then compared to the normalized human scores and a mean squared error loss is computed. Using this strategy, good translations should be very close together in the embedding space.

This approach uses the reference translations for scoring, so the underling BERT encoder can be monolingual. We thus employ the standard English BERT model (Devlin et al., 2018) and a German BERT model (Chan et al., 2020), both provided by the Huggingface Python library (Wolf et al., 2019). The two models are 12-layer transformer encoders with a hidden size of 768.

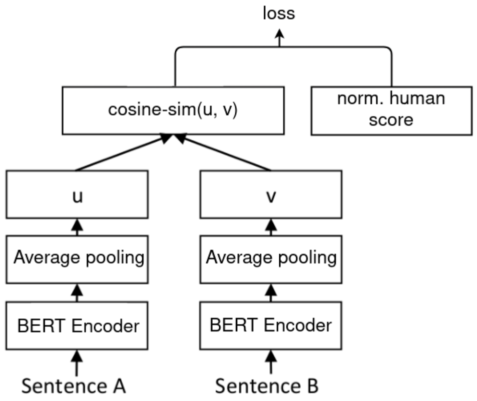

Figure 2: Siamese network architecture used for semantic similarity training. Note that the same encoder network is used to encode both sentences in the pair.

We also wanted to build scoring models that ignore the translation references. Ideally this could be done by replacing the monolingual encoders with multilingual ones. The main issue with multilingual sentence embeddings is that they are usually not language-aligned, so training the model to generate close embeddings for the source and output segments may be hard. (Reimers and Gurevych, 2020) proposed a method of training multilingual models that generate language-aligned embeddings. This is achieved by using knowledge distillation from monolingual models and training the model using parallel corpora with sentences in multiple languages.

We use such a model provided by the authors through the Sentence Transformers library[2]. The checkpoint name is *distiluse-base-multilingual-cased-v1*.

Preprocessing: the segment pairs used for training are tokenized using a specific tokenizer for the chosen BERT-like encoder (usually Sentencepiece tokenizers (Kudo and Richardson, 2018)) and specific tokens for the beginning and the end of the segment are added. Figure 3 contains the distribution of number of tokens for the source, reference and output segments from both language pair datasets. In order to limit the demand of computational resources and allow for a larger batch size, token sequences were trimmed to a maximum length of 100 tokens.

Training details: In order to train the encoders using the siamese strategy we firstly split the datasets into 80% train and 20% test. This is needed since we wish to compare scoring perfor-

[2] https://www.sbert.net

Language pair	German to English		English to German	
Correlation level	segment	document	segment	document
BLEU	0.4631	0.6894	0.2884	0.3993
METEOR	0.6416	0.7837	0.3061	0.4314
CHRF	0.6425	0.7699	0.3944	0.5216
Monolingual BERT	**0.7120**	**0.7946**	**0.5388**	**0.6809**
Multilingual BERT	0.6630	0.7706	0.3694	0.5157

Table 3: Pearson correlation between scores computed using the transformer-based encoders and the human judgments from the test dataset. In this context we call "monolingual" approaches models trained using the references and the system outputs, while "multilingual" describes the harder task of scoring using only the source text and the corresponding system output. Correlation with baseline scores is also computed.

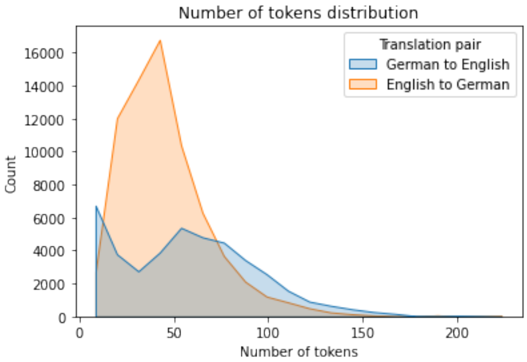

Figure 3: Distribution of number of tokens, computed for both English-to-German and German-to-English datasets.

mances on unseen data. As before, the segments coming from the same document are placed in the same split. For optimizing the model's weights, we used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of $2 \cdot 10^{-5}$ (used for fine-tuning) and 0.01 weight decay. A learning rate scheduler is employed for increasing the learning rate linearly to the specified value for 10% of the training steps (warm-up), then decreasing it linearly for the rest of the steps. Training was done in mini-batches of size 16, for a total of 5 epochs. The model weights are fine-tuned end-to-end. Training one model usually took between one and two hours on an 8GB VRAM GPU.

After training, each model was evaluated on its corresponding test dataset and Pearson correlations were computed between the predicted scores and the human annotations, for both segment-level and document-level translations. When dealing with multiple references for a segment translation, the predicted score is computed as the maximum of all scores computed for each reference. For document scoring, the models assign the average segment score as the overall score of that document. Table 3 contains the performance comparison between the monolingual models (trained using the references) and the multilingual ones (trained without any reference translation, only using the source texts and their corresponding system outputs) on the test dataset. As expected, models that take into account reference translation obtain better correlation with human annotations, but multilingual approaches are not that far off, having also the advantage of not needing reference translations.

Document correlations are usually higher than segment level, one reason for this might be the fact that averaging independent segment scores the overall evaluated quality of translation becomes less noisy. As stated by (Mathur et al., 2020) making observations on document-level scores is hard since one needs to take a closer look at examples where the document context is needed for a good translation. Usually it was seen that this does not happen for a large number of segments, and the errors in translation quality estimation for them can be averaged out when computing the document score.

Better correlations were found for the to-English dataset, this can be because of better annotations but also because of a somewhat skewed score distribution. We plot in figures 4 and 5 the score distributions predicted by the transformer-based architectures along with the normalized human judgments. We can see that both distributions contain more higher scoring translations.

5 Conclusions and Future Work

In this paper we have detailed two approaches to automatic evaluation of machine translation. The first one involved augmenting existing traditional metrics with semantic information extracted from

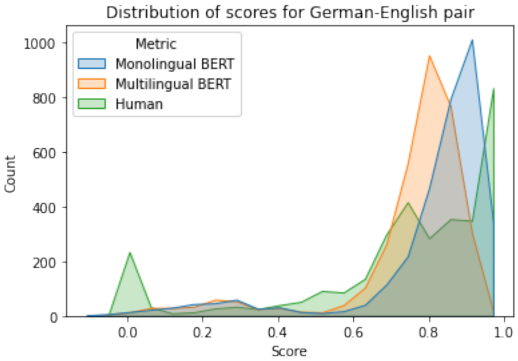

Figure 4: Distribution of segment-level human judgments and scores predicted by the transformer-based models on the test dataset for the German-to-English language pair

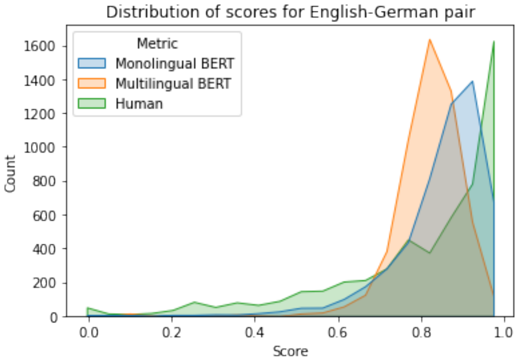

Figure 5: Distribution of segment-level human judgments and scores predicted by the transformer-based models on the test dataset for the English-to-German language pair

the reference translation and the system outputs, by employing UCCA representation graphs. In the second one, we attacked the problem with large-scale pre-trained transformer networks and tried to obtain embedding spaces where good translations are represented with close vectors. We also experimented with reference-free translation evaluation using multilingual approaches.

For UCCA-based features, there is still a lot of ideas to explore and not so many resources available. The parser we used is not being maintained at the moment and the only languages it supports are English, German and French. Building such representations for more languages can boost automated metrics for a large number of tasks, even for languages with lower resources. The lack of an open-sourced German WordNet was also a problem we faced and because of that we needed to fall back to exhaustive dictionaries of synonyms.

For transformer-based models, there is always the question of time and resources. These requirements limit us in terms of how many hyperparameters and training settings we can try. Ideally, looking more into language-aligned embedding representations may provide a good way of scoring translations without the need of wasting data from parallel corpora just to evaluate translation systems.

Finally, our implementation for the proposed models can be accessed at: https://github.com/eu3neuom/machine-translation

Acknowledgements

All team members contributed equally to this paper and had worked on each section in various proportions. Teodor was mainly responsible for the implementation of semantic structure-based approaches using UCCA, while Bogdan directed the high-level targets of the paper and worked on the transformer-based encoders implementation and experiments.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. *arXiv preprint arXiv:2010.10906*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proc. of ACL*, pages 1127–1138.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proc. of ACL*, pages 373–385.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Alon Lavie and Abhaya Agarwal. 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*, pages 228–231.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Ananya Mukherjee, Hema Ala, Manish Shrivastava, and Dipti Misra Sharma. 2020. Mee : An automatic metric for evaluation using embeddings for machine translation. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 292–299.

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 of the Association for Computational Linguistics*, pages 311–318.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Martin F Porter. 2001. Snowball: A language for stemming algorithms.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André FT Martins, and Alon Lavie. 2021. Are references really needed? unbabel-ist 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. *arXiv preprint arXiv:2004.14564*.

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*, pages 5998–6008.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jin Xu, Yinuo Guo, and Junfeng Hu. 2020. Incorporate semantic structures into machine translation evaluation via ucca. *arXiv preprint arXiv:2010.08728*.