Train, Sort, Explain: Learning to Diagnose Translation Models

Robert Schwarzenberg[1], **David Harbecke**[1], **Vivien Macketanz**[1],
Eleftherios Avramidis[1], **Sebastian Möller**[1,2]
[1]German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
[2]Technische Universität Berlin, Berlin, Germany
{firstname.lastname}@dfki.de

Abstract

Evaluating translation models is a trade-off between effort and detail. On the one end of the spectrum there are automatic count-based methods such as BLEU, on the other end linguistic evaluations by humans, which arguably are more informative but also require a disproportionately high effort. To narrow the spectrum, we propose a general approach on how to automatically expose systematic differences between human and machine translations to human experts. Inspired by adversarial settings, we train a neural text classifier to distinguish human from machine translations. A classifier that performs and generalizes well after training should recognize systematic differences between the two classes, which we uncover with neural explainability methods. Our proof-of-concept implementation, DiaMaT, is open source. Applied to a dataset translated by a state-of-the-art neural Transformer model, DiaMaT achieves a classification accuracy of 75% and exposes meaningful differences between humans and the Transformer, amidst the current discussion about human parity.

1 Introduction

A multi-dimensional diagnostic evaluation of performance or quality often turns out to be more helpful for system improvement than just considering a one-dimensional utilitarian metric, such as BLEU (Papineni et al., 2002). This is exemplified by, for instance, the pioneering work of Bahdanau et al. (2014). The authors introduced the attention mechanism responding to the findings of Cho et al. (2014) who reported that neural translation quality degraded with sentence length. The attention mechanism was later picked up by Vaswani et al. (2017) for their attention-only Transformer model, which still is state of the art in machine translation (MT) (Bojar et al., 2018). Furthermore, while MT output approaches human translation quality and the claims for ”human parity” (Wu et al., 2016; Hassan et al., 2018) increase, multi-dimensional diagnostic evaluations can be useful to spot the thin line between the machine and the human.

Diagnostic (linguistic) evaluations require human-expert feedback, which, however, is very time-consuming to collect. For this reason, there is a need for tools that mitigate the effort, such as the ones developed by Madnani (2011); Popović (2011); Berka et al. (2012); Klejch et al. (2015).

In this paper we propose a novel approach for developing evaluation tools. Contrary to the above tools that employ string comparison methods such as BLEU, implementations of the new approach derive annotations based on a neural model of explainability. This allows both capturing of semantics as well as focusing on the particular tendencies of MT errors. Using neural methods for the evaluation and juxtaposition of translations has already been done by Rikters et al. (2017). Their method, however, can only be applied to attention-based models and their translations. In contrast, our approach generalizes to arbitrary machine and even human translations. After first discussing the abstract approach in the next section, we present a concrete open-source implementation, “DiaMaT” (from *Dia*gnose *Ma*chine *T*ranslations).

2 Approach

The proposed approach consists of the three steps (1) *train*, (2) *sort*, and (3) *explain*.

2.1 Step 1: Train

In a first step, inspired by generative adversarial networks (Goodfellow et al., 2014; Wu et al., 2017; Yang et al., 2017) we propose to train a model to distinguish machine from human translations. The premise is that if the classifier generalizes well after training, it has learned to recognize

Proceedings of NAACL-HLT 2019: Demonstrations, pages 29–34
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

systematic or frequent differences between the two classes (herinafter also referred to as “class evidence”). Class evidence may be, for instance, style differences, overused n-grams but also errors. The text classifier can be implemented through various architectures, ranging from deep CNNs (Conneau et al., 2017) to recurrent classifiers built on top of pre-trained language models (Howard and Ruder, 2018).

2.2 Step 2: Sort

In a second step, we suggest letting the trained classifier predict the labels of a test set which contains human and machine translations and then sort them by classification confidence. This is based on the assumption that if the classifier is very certain that a given translation was produced by a machine (translation moved to the top of the list in this step), then the translation should contain strong evidence for a class, i.e. errors typical for only the machine. Furthermore, even if we are dealing with a very human-like MT output, which means that our classifier may only slightly perform above chance, sorting by classification confidence should still move the few systematic differences that the classifier identified to the top.

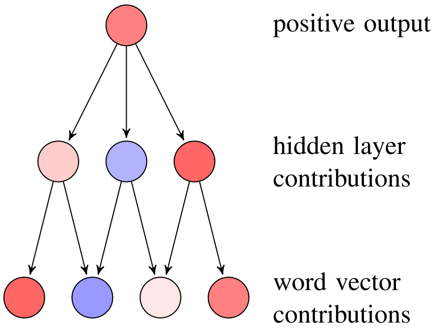


Figure 1: Contributions propagated from output to input space. Colors represent positive (red) and negative (blue) contributions. The Figure is adapted from Kindermans et al. (2018).

2.3 Step 3: Explain

Arras et al. (2016, 2017a,b) demonstrated the data exploratory power of explainability methods in Natural Language Processing (NLP). This is why in a third step, we propose to apply an explainability method to uncover and visualize the class evidence on which the classifier based its decisions. Our definition of an explanation follows Montavon et al. (2018), who define it as “the collection of features of the interpretable domain, that have contributed for a given example to produce a decision (e.g. classification or regression).”[1] In our case the interpretable domain is the plain text space. There exist several candidate explainability methods, one of which we present in the following as an example.

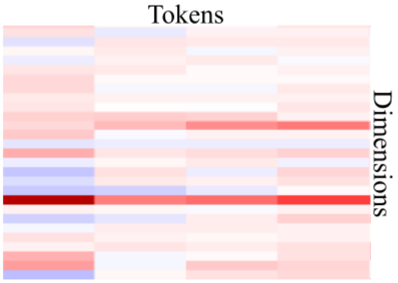


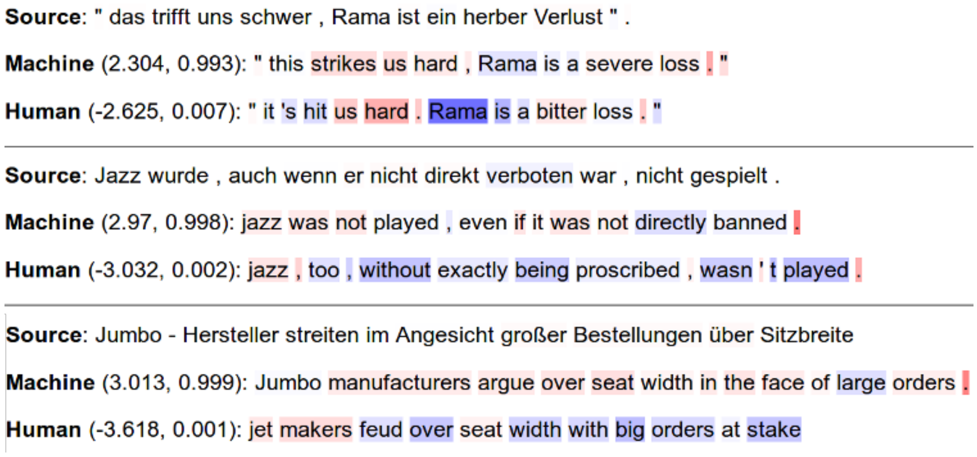
Figure 2: A heatmap of contribution scores in word vector space over a sequence of tokens. Red means positive contribution (score > 0), blue means negative contribution (score < 0).

2.3.1 Explainability and Interpretability Methods for Data Exploration

In their tutorial paper, Montavon et al. (2018) discuss several groups of explainability methods. One group, for instance, identifies how sensitively a model reacts to a change in the input, others extract patterns typical for a certain class. Here, we discuss methods that propagate back contributions.

The contribution flow is illustrated in Fig. 1. At the top, the depicted binary classifier produced a positive output (input classified as class one). The classification decision is based on the fact that in the previous layer, the evidence for class one exceeded the evidence for class zero: The left and the right neuron contributed positively to the decision (reddish), whereas the middle neuron contributed negatively (blueish). Several explainability methods, such as Layerwise Relevance Propagation (Bach et al., 2015) or PatternAttribution (Kindermans et al., 2018), backtrack contributions layer-wise. The methods have to preserve coherence over highly non-linear activation functions. Eventually, contributions are projected into the input space where they reveal what the model considers emblematic for a class. This is what we exploit in step 3.

[1] Montavon et al. (2018) distinguish between explainability and interpretability. Interpretability methods also hold potential for the approach. For brevity, we limit ourselves to explainability methods here.



Source: " das trifft uns schwer , Rama ist ein herber Verlust " .

Machine (2.304, 0.993): " this strikes us hard , Rama is a severe loss . "

Human (-2.625, 0.007): " it 's hit us hard . Rama is a bitter loss . "

Source: Jazz wurde , auch wenn er nicht direkt verboten war , nicht gespielt .

Machine (2.97, 0.998): jazz was not played , even if it was not directly banned .

Human (-3.032, 0.002): jazz , too , without exactly being proscribed , wasn ' t played .

Source: Jumbo - Hersteller streiten im Angesicht großer Bestellungen über Sitzbreite

Machine (3.013, 0.999): Jumbo manufacturers argue over seat width in the face of large orders .

Human (-3.618, 0.001): jet makers feud over seat width with big orders at stake

Figure 3: Screenshot of demonstrative results in DiaMaT. Filters that allow the user to analyse the corpus are not shown. The bold label is the true label. The activations of the machine neuron are shown in brackets; on the left the unnormalized logit activation, on the right the softmax activation. Positive logits and softmax probabilities greater 0.5 indicate machine evidence, as do tokens highlighted in red. Consequently, blue indicates evidence for the human. The more intense the colour, the stronger the evidence.

Explainability methods in NLP (Arras et al., 2016, 2017a,b; Harbecke et al., 2018) are typically used to first project scores into word-vector space resulting in heat maps as shown in Fig. 2. To interpret them in plain text space, the scores are summed over the word vector dimensions to compute RGB values for each token, resulting in plain text heatmaps as shown in Fig. 3.

3 Implementation

For step 1 (training phase), DiaMaT[2] deploys a CNN text classifier, the architecture of which is depicted in Fig. 4. The classifier consumes three embeddings: the embedding of a source and two translations of the source, one by a machine and one by a human. It then separately convolves over the embeddings and subsequently applies max pooling to the filter activations. The concatenated max features are then passed to the last layer, a fully connected layer with two output neurons. The left neuron fires if the machine translation was passed to the left input layer, the right neuron fires if the machine translation was passed to the right input layer. Note that this layer allows the model to combine features from all three inputs for its classification decision.

For step 2 (sorting phase), DiaMaT offers to sort by unnormalized logit activations or by softmax activations. Furthermore, one can choose to use the machine neuron activation or the human neuron activation as the sorting key.

For step 3 (explaining phase), DiaMaT employs the iNNvestigate toolbox (Alber et al., 2018) in the back-end that offers more than ten explainability methods: Replacing one method with another only requires to change one configuration value in DiaMaT, before repeating step 3 again. In step 3, DiaMaT produces explanations in the form of $(token,\ score)$ tuple lists that are consumed by a front-end server which visualizes the scores as class evidence (see Fig. 3).[3]

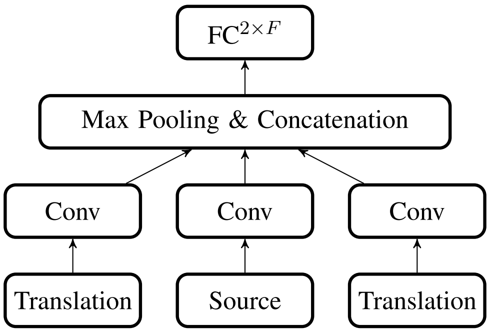


Figure 4: Architecture of the text classifier.

4 Datasets and Experiments

We tested DiaMaT on a corpus translated by an NMT Transformer engine (Vaswani et al., 2017)

[2] Source code, data and experiments are available at https://github.com/dfki-nlp/diamat.

[3] The front-end was inspired by the demo LRP server of the Fraunhofer HHI insitute https://lrpserver.hhi.fraunhofer.de/text-classification, last accessed 2019-01-31.

conforming to the WMT14 data setup (Bojar et al., 2014). The NMT model was optimized on the test-set of WMT13 and an ensemble of 5 best models was used. It was trained using OpenNMT (Klein et al., 2017), including Byte Pair Encoding (Sennrich et al., 2015) but no back-translation, achieving 32.68 BLEU on the test-set of WMT14.

Next, we trained the CNN text classifier sketched in Fig. 4 for which we randomly drew 1M training samples (human references and machine translations alongside their sources) from the WMT18 training data (Bojar et al., 2018), excluding the WMT14 training data. The validation set consisted of 100k randomly drawn samples from the same set and we drew another 100k samples randomly for training the explainability method of choice, PatternAttribution, which learns explanations from data. All texts were embedded using pre-trained fastText word vectors (Grave et al., 2018).

We evaluated the classifier on around 20k samples drawn from the official test sets, excluding WMT13. On this test set, the classifier achieved an accuracy of 75%, which is remarkable, considering the ongoing discussion about human parity (Wu et al., 2016; Hassan et al., 2018). We also used this test set for steps 2 and 3. Thus, neither the translation model, nor the text classifier, nor the explainability method encountered this split during training. For step 2, the machine translation was always passed to the right input layer and contributions to the right output neuron were retrieved with PatternAttribution.[4] We then sorted the inputs by the softmax activation of the machine neuron, which moved inputs for which the classifier is certain that it has identified the machine correctly to the top.

5 Demonstration and Observations

We observed that the top inputs frequently contained sentences in which DiaMaT considered the token after a sentence-ending full stop strong evidence for the human (Fig. 3, top segment). We take this as evidence that DiaMaT correctly recognized that the human generated multiple sentences instead of a single one more often than the machine did. At this point, we cannot, however, offer an explanation for why the token preceding the punctuation mark is frequently considered evidence for the machine.

Furthermore, DiaMaT also regarded reduced negations (“n’t”) as evidence for the human (see Fig. 3, middle segment) which again is reflected in the statistics. The machine tends to use the unreduced negation more frequently.

The last segment in Fig. 3 shows how DiaMaT points to the fact that the machine more often produced sentence end markers than the human in cases where the source contained no end marker. The claims above are all statistically significant in the test set, according to a χ^2 test with $\alpha = 0.001$.

6 Future Work

The inputs in Fig. 3 contain easily readable evidence. There is, however, also much evidence that is hard to read. In general, we can assume that with increasing architectural complexity, more complex class evidence can be uncovered, which may come at the cost of harder readability.

In the future, it is worth exploring how different architectures and model choices affect the quality, complexity and readability of the uncovered evidence. For instance, one direction would be to to train the classifier on top of a pretrained language model (Howard and Ruder, 2018; Devlin et al., 2019) which could improve the classification performance. Furthermore, other explainability methods should also be tested.

7 Conclusion

We presented a new approach to analyse and juxtapose translations. Furthermore, we also presented an implementation of the approach, DiaMaT. DiaMaT exploits the generalization power of neural networks to learn systematic differences between human and machine translations and then takes advantage of neural explainability methods to uncover these. It learns from corpora containing millions of translations but offers explanations on sentence level. In a stress test, DiaMaT was capable of exposing systematic differences between a state-of-the-art translation model output and human translations.

Acknowledgements

This research was partially supported by the German Federal Ministry of Education and Research through the project DEEPLEE (01IW17001). We would also like to thank the anonymous reviewers and Leonhard Hennig for their helpful feedback.

[4] In order to visualize evidence for the human (blue), positive contributions in the left input needed to be inverted.

References

Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. 2018. iNNvestigate neural networks! *arXiv preprint arXiv:1808.04260*.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining Predictions of Non-Linear Classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017a. "What is relevant in a text document?": An interpretable machine learning approach. *PLOS ONE*, 12(8).

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017b. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the EMNLP'17 Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA)*, pages 159–168. Association for Computational Linguistics.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Jan Berka, Ondřej Bojar, Mark Fishel, Maja Popović, and Daniel Zeman. 2012. Automatic MT Error Analysis: Hjerson Helping Addicter. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, pages 2158–2163, Istanbul, Turkey. European Language Resources Association.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 272–307, Belgium, Brussels. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the European Chapter of the Association for Computational Linguistic (EACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

David Harbecke, Robert Schwarzenberg, and Christoph Alt. 2018. Learning explanations from language data. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 316–318.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. *CoRR*, abs/1803.05567.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Pieter-Jan Kindermans, Kristof T. Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. 2018. Learning how to explain neural networks: PatternNet and PatternAttribution. In *International Conference on Learning Representations (ICLR)*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. OpenNMT:

Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Ondrej Klejch, Eleftherios Avramidis, Aljoscha Burchardt, and Martin Popel. 2015. MT-ComparEval: Graphical evaluation interface for Machine Translation development. *The Prague Bulletin of Mathematical Linguistics*, 104(1):63–74.

Nitin Madnani. 2011. iBLEU: interactively debugging and scoring statistical machine translation systems. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 213–214.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

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.

Maja Popović. 2011. Hjerson: An Open Source Tool for Automatic Error Classification of Machine Translation Output. *The Prague Bulletin of Mathematical Linguistics*, 96(-1):59–68.

Matīss Rikters, Mark Fishel, and Ondřej Bojar. 2017. Visualizing Neural Machine Translation Attention and Confidence. *The Prague Bulletin of Mathematical Linguistics*, 109(1):39–50.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural Machine Translation of Rare Words with Subword Units. *CoRR*, abs/1508.0.

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.

Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2017. Adversarial Neural Machine Translation. *CoRR*, abs/1704.06933.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *Computer Research Repository*, abs/1609.0.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2017. Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets. *CoRR*, abs/1703.04887.