

NLytics at CheckThat! 2021: Detecting Previously Fact-Checked Claims by Measuring Semantic Similarity

Albert Pritzkau¹

¹*Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, Fraunhoferstraße 20, 53343 Wachtberg, Germany*

Abstract

The following system description presents our approach to the detection of previously fact-checked claims. Given a claim originating from a tweet or a political debate, we specified the similarity to a collection of previously fact-checked claims. In line with the origin of the claims, the collection of previously fact-checked claims is composed of tweets and political debates respectively. The given task has been framed as a sequence similarity problem. Relevance scoring is based on semantic similarity. Similarity is calculated by distance metrics on representation vectors at paragraph level.

Keywords

Information Retrieval, Semantic Similarity, Deep Learning, Transformers, RoBERTa

1. Introduction

Social networks provide opportunities to conduct disinformation campaigns for organizations as well as individual actors. The proliferation of disinformation online, has given rise to a lot of research on automatic fake news detection. CLEF 2021 - CheckThat! Lab [1, 2] considers disinformation as a communication phenomenon. By detecting the use various claims in (political) communication, it takes into account not only the content but also how a subject matter is communicated by specific actors, in particular, by repetition of the same claims.

Task definition: Detect Previously Fact-Checked Claims Given a check-worthy claim, and a set of previously fact-checked claims, determine whether the claim has been previously fact-checked. Based on the source of the considered claims the shared task [3] defines the following subtasks both of which are framed as ranking tasks:

- Subtask A: Detect Previously Fact-Checked Claims in Tweets
- Subtask B: Detect Previously Fact-Checked Claims in Political Debates/Speeches

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania


✉ albert.pritzkau@fkie.fraunhofer.de (A. Pritzkau)

🌐 <https://www.fkie.fraunhofer.de/> (A. Pritzkau)

🆔 0000-0001-7985-0822 (A. Pritzkau)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

In this work, we covered our approach. Below, we describe the systems built for these two subtasks. At the core of our systems is RoBERTa [4], a pre-trained model based on the Transformer architecture [5].

2. Related Work

The goal of the shared task is to investigate automatic techniques for Information Retrieval (IR) obtaining information resources relevant to an information need from a larger collection of information resources. In particular, we want to find the most similar paragraphs from a large set of documents given a query document.

2.1. Relevancy scoring

Relevancy scoring is a process to determine the relevance of retrieved documents based on user queries, term frequencies, and other important parameters. In the simplest form, documents are ranked on how many words of the document match the terms in the query. In the given task the performance is evaluated with an Elastic Search baseline. Elastic Search uses two kinds of scoring function – TF-IDF and Okapi BM25 – both of which follow the same principal of lexical similarity. In particular, to determine a relevancy score TF-IDF as in equation(3), short for term frequency–inverse document frequency, is used as a statistical measure to evaluate the importance of a query term t to a document d form a larger collection of documents D . The importance increases proportionally to the number of times a query term appears in the document as shown in equation (1) but is offset by the frequency of the query term in the whole collection as shown in equation (2).

$$TF(t, d) = \frac{\text{term frequency in document}}{\text{total words in document}} \quad (1)$$

$$IDF(t, D) = \log_2 \left(\frac{\text{total documents in corpus}}{\text{documents with term}} \right) \quad (2)$$

$$TF-IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (3)$$

Based on TF-IDF, Okapi BM25 as in 4 handles some of it's shortcomings to make the function result more relevant to the user's query. Same as TF-IDF, relevance is calculated as a result of multiplying TF and IDF with the difference of how these values are calculated. With $|D|$ as the number of words in a document it takes into account the document's length. Furthermore, k_1 normalizes the impact of the frequent occurrences of common words on the relevance score.

$$BM25(D, Q) = \sum_{i=1}^n IDF(q_i, D) \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i) + k_1 \cdot (1 - b + b \cdot |D|/d_{avg})} \quad (4)$$

Beyond word matching, our approach considers semantic similarity for relevance estimation. Neural network based language models are receiving significant attention in the field of natural language processing due to their capability to effectively capture semantic information

representing words, sentences or even larger text elements in low-dimensional vector space. Exploiting language models for the task of ad-hoc retrieval has already been demonstrated by [4] and [6]. In this study, we investigated Sentence-BERT [7] to derive semantically meaningful sentence embeddings, that to some extent recognizes synonyms, acronyms or spelling variations. From this point of view, the meaningfulness of a comparison of semantic similarity with lexical similarity as a baseline could be debated.

2.2. Text embeddings

For a long time word embeddings such as Word2Vec [8], PLSI [9], GloVe [10] have been a cornerstone for deep learning (DL) NLP. The embeddings are often pre-trained on unlabeled text corpus from co-occurrence statistics. Due to this fact, these representations are context independent. To capture semantic and morpho-syntactic properties contextual representations sentence-level approaches such as Semi-Supervised Sequence Learning [11], ULMFit [12], ELMo [13], GPT [14], XLNet [15] and BERT [16] are gaining increasing importance. The ability of embeddings such as RoBERTa [4], a pre-trained model based on the Transformer architecture [5] to capture semantic and morpho-syntactic properties has been shown on various NLP tasks like sentiment analysis, question answering.

2.3. About BERT and RoBERTa

BERT stands for Bidirectional Encoder Representations from Transformers. It is based on the Transformer model architectures introduced by Vaswani et al. [5]. The general approach consists of two stages: first, BERT is pre-trained on vast amounts of text, with an unsupervised objective of masked language modeling and next-sentence prediction. Second, this pre-trained network is then fine-tuned on task specific, labeled data. The Transformer architecture is composed of two parts, an Encoder and a Decoder, for each of the two stages. The Encoder used in BERT is an attention-based architecture for NLP. It works by performing a small, constant number of steps. In each step, it applies an attention mechanism to understand relationships between all words in a sentence, regardless of their respective position. By pre-training language representations, the Encoder yields models that can either be used to extract high quality language features from text data, or fine-tune these models on specific NLP tasks (classification, entity recognition, question answering, etc.). We rely on RoBERTa [4], a pre-trained Encoder model which builds on BERT's language masking strategy. However, it modifies key hyperparameters in BERT such as removing BERT's next-sentence pre-training objective, and training with much larger mini-batches and learning rates. Furthermore, RoBERTa was also trained on an order of magnitude more data than BERT, for a longer amount of time. This allows RoBERTa representations to generalize even better to downstream tasks compared to BERT. In this study, RoBERTa is at the core of each solution of the given subtasks.

2.4. Sentence embeddings

Sentence embeddings can be described as a document processing method of mapping sentences to vectors as a means of representing text with real numbers suitable for machine learning. For RoBERTa, the representation vectors consisting of 768 numerical values are considered

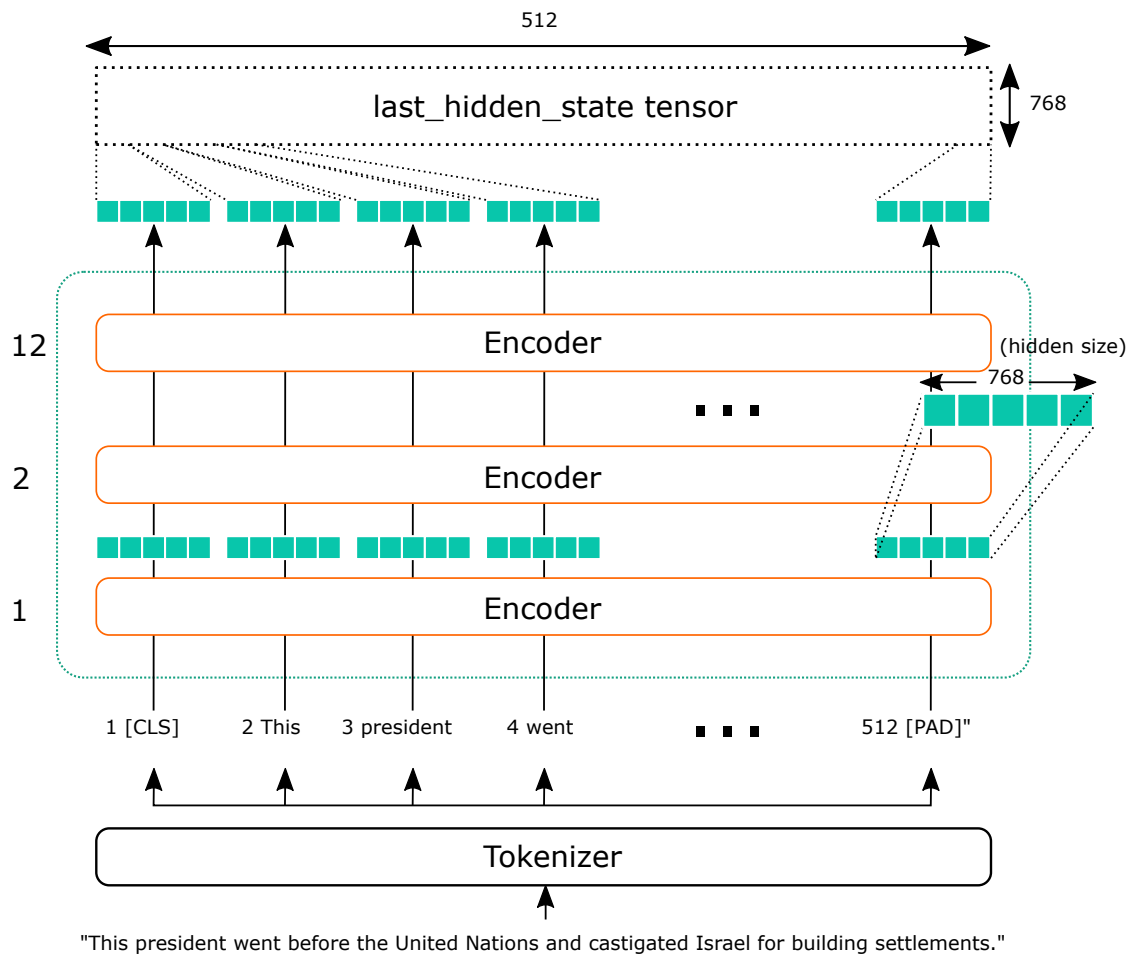


Figure 1: 12-layer transformer network — with the hidden layer representations highlighted in green.

as contextual word embeddings of a single token. Because there is one of these vectors for representing each token (output by each encoder), we are actually looking at a tensor of size 768 by the number of tokens (in our case 512). This tensor is transformed into a semantic representations of the whole input sequence. The simplest and most commonly extracted tensor is the `last_hidden_state` tensor — which is conveniently output by the BERT model. To convert the `last_hidden_states` tensor into a sequence vector, a mean pooling operation is used. This pooling operation takes the mean of all token embeddings.

In this study, we investigated Sentence-BERT [7] to derive semantically meaningful sentence embeddings. Reimers and Gurevych [7] by modified the original BERT using Siamese networks. With Sentence-BERT we can now take advantage of BERT embeddings for the tasks like semantic similarity comparison and information retrieval via semantic search. Similarity metrics are applied to the resulting sequence vectors to calculate the respective similarity between different sequences.

Table 1

Statistical summary of token counts on the collection of verified claims.

| Source | Tweets (A) | Debates/Speeches (B) |
|--------|------------|----------------------|
| count | 13825 | 19250 |
| mean | 16.86 | 18.01 |
| std | 6.38 | 8.12 |
| min | 1 | 2 |
| 25% | 13 | 12 |
| 50% | 16 | 16 |
| 75% | 20 | 22 |
| max | 110 | 79 |

2.5. Similarity metrics

To turn language into a machine-readable format, words and sentences are converted into high-dimensional vectors — organized so that each vector’s geometric position can attribute meaning. We expect that similar meaning corresponds with proximity/orientation between those vectors. Similarity measurements such as cosine similarity or Manhattan/Euclidean distance, evaluate semantic textual similarity so that the scores can be exploited for a variety of helpful NLP tasks, including information retrieval. For Semantic Textual Similarity (STS) tasks Reimers and Gurevych [7] suggest to compute the Spearman’s rank correlation between the cosine-similarity of the sentence embeddings and the gold labels, if they exist.

2.6. Evaluation measures

For both tasks the submitted ranked lists per claim have been evaluated using ranking evaluation measures $MAP@k$ for $k \in \{1, 3, 5, 10, all\}$ (Mean Average Precision for the top-k vClaims), MRR (Mean Reciprocal Rank) and $Precision@k$ for $k \in \{3, 5, 10\}$ (Precision for the top-k vClaims). $MAP@5$ has been defined as the official measure.

3. Dataset

The data for the task was developed during the CLEF-2021 CheckThat! campaign [1, 2, 3].

As presented in Table 1, the given collection contain 13825 and 19250 verified claims, respectively. Additional information is provided for each vClaim: the title of the entry, (the subtitle), the author/speaker, and the date of verification, (link to the justification). Additionally, there are 1000 and 563 positively labeled $\langle iClaim, vClaim \rangle$ pairs in the training set to possibly fine-tune the language model. However, to obtain sentence embeddings, in this study we assume a model without any task-specific fine-tuning. Thus, we postponed the fine-tuning to future work.

Table 2

Statistical summary of token counts on the collection of input claims.

| Source | Tweets (A) | Debates/Speeches (B) |
|--------|------------|----------------------|
| count | 1196 | 702 |
| mean | 32.19 | 20.25 |
| std | 12.97 | 13.94 |
| min | 11 | 1 |
| 25% | 23 | 10 |
| 50% | 29 | 17 |
| 75% | 40 | 26 |
| max | 108 | 91 |

4. Our approach

Problem Definition. Suppose we have a set C of claims and a set V of previously verified claims. Each claim $c \in C$ and verified claim $v \in V$ can be represented as (c, v, y) , where y is a variable indicating the distance between c and v . Therefore, the solution of sentence-level retrieval task could be considered as a text similarity problem. Given a claim c and a list of candidates of verified claims $Candidate(c) \subset V$, our goal is to predict $p(y|c, v)$ of each input claim c with each candidate $v \in Candidate(c)$.

4.1. Experimental setup

Word-Level Sentence Embeddings. A sentence is split into words w_1, \dots, w_n with length of n by the WordPiece tokenizer [17]. The word w_i and its index i (w_i 's absolute position in the sentence) are projected to vectors by embedding sub-layers, and then added to the index-aware word embeddings:

$$\hat{w}_i = WordEmbed(w_i)$$

$$\hat{u}_i = IdxEmbed(i)$$

$$h_i = LayerNorm(\hat{w}_i + \hat{u}_i)$$

We use Sentence-BERT [7] to compute dense vector representations for sentences and paragraphs. The embedding model is trained on paraphrases which is available from the model repository at huggingface.co¹. We determine the similarity at the paragraph level. Although the model can handle up to 512 tokens, we decided to split the documents into paragraphs with a target length of 15. Based on the length of the input claims (see Table 2) we try to adapt the length of the examined fragments of the verified claim to keep the comparison roughly balanced. We are aware that with this restriction we have made a rather conservative choice. This parameter can be adjusted in the future.

¹<https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v1>

Table 3

Results on the test set on subtask A.

| Rank | Team | MAP@5 | MAP@1 | RR | P@3 | P@5 |
|----------|----------------|--------------|--------------|--------------|--------------|--------------|
| 1 | aschern | 0.883 | 0.861 | 0.884 | 0.300 | 0.182 |
| 2 | NLytics | 0.799 | 0.738 | 0.807 | 0.289 | 0.179 |
| 3 | DIPS | 0.787 | 0.728 | 0.795 | 0.282 | 0.177 |
| 4 | shaar | 0.749 | 0.703 | 0.761 | 0.262 | 0.164 |

Table 4

Results on the test set on subtask B.

| Rank | Team | MAP@5 | MAP@1 | MAP@3 | MAP_AIIRR | P@3 | P@5 | |
|----------|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 1 | sshaar | 0.346 | 0.304 | 0.339 | 0.355 | 0.350 | 0.143 | 0.091 |
| 2 | DIPS | 0.328 | 0.278 | 0.313 | 0.346 | 0.336 | 0.143 | 0.099 |
| 3 | NLytics | 0.215 | 0.171 | 0.210 | 0.223 | 0.216 | 0.101 | 0.068 |

Similarity metrics. We use Euclidean distance of normalized vectors, which for two vectors \vec{c}, \vec{v} is equal to $distance(\vec{c}, \vec{v}) = \sqrt{2 * (1 - \cos(\hat{c}, \hat{v}))}$. The submitted scores represent the multiplicative inverse of the *distance*. In case of duplicate scores due to chunking of large documents, we only consider top scores.

4.2. Results and Discussion

We participated in both subtasks. Official evaluation results on the test set are presented in Table 3 and Table 4 for each subtask, respectively. shaar is a baseline submission (Elastic Search) of the competition organizers.

For the subtask A our system was ranked 2nd. However, a large gap (MAP@5: 0.883 vs. 0.799) with the superior approach should be mentioned. The result reflects the performance of the pre-trained language model used without task-specific fine-tuning. The submitted relevance scores are based on semantic similarity only, resulting from distances of the vector representations at paragraph level. As already mentioned in section 2.1, the comparison of the ranking to the given baseline seems problematic, since these are based on semantic similarity on the one hand and on lexical similarity on the other hand. Thus, this approach can potentially be used as a baseline for comparing other ranking methods based on semantic similarity.

The particularly poor results of the subtask B are astonishing, since the same procedure was followed. Our system was ranked last not even passing the given baseline on this task. The comparison of the input data resulting from tables 1 and 2 do yield any useful clues for explanation, since they are similar in scope and length. On the contrary, we expected the phrase structure in political debates should significantly improve semantic representation. For this reason, the problem of comparing semantic and lexical similarity is considered responsible for the poor performance.

5. Conclusion and Future work

We described our approach for the CLEF 2021 - CheckThat! Lab: Detecting Previously Fact-Checked Claims. We employed RoBERTa-based neural architectures to encode text sequences into a dense vector space. Similarity scores are being calculated using geometric distances between representation vectors at paragraph level. In future work, we will examine the impact task-specific fine-tuning on relevance ranking. Furthermore, we plan to investigate more recent neural architectures for language representation such as T5 [18] and GPT-3 [19]. Finally, from probing experiments, the morpho-syntactic and semantic features captured by the embedding models could be extracted to be used in an elaborated weighting scheme for relevance scores.

References

- [1] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl, The CLEF-2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News, in: Proceedings of the 43rd European Conference on Information Retrieval, ECIR'21, Lucca, Italy, 2021, pp. 639–649. URL: https://link.springer.com/chapter/10.1007/978-3-030-72240-1_75.
- [2] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, W. Mansour, B. Hamdan, Z. S. Ali, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl, M. Kutlu, Y. S. Kartal, Overview of the CLEF-2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News, in: Proceedings of the 12th International Conference of the CLEF Association: Information Access Evaluation Meets Multilinguality, Multimodality, and Visualization, CLEF'2021, Bucharest, Romania (online), 2021.
- [3] S. Shaar, F. Haouari, W. Mansour, M. Hasanain, N. Babulkov, F. Alam, G. Da San Martino, T. Elsayed, P. Nakov, Overview of the CLEF-2021 CheckThat! Lab Task 2 on Detect Previously Fact-Checked Claims in Tweets and Political Debates, in: Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum, CLEF'2021, Bucharest, Romania (online), 2021.
- [4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019. [arXiv:1907.11692](https://arxiv.org/abs/1907.11692).
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, volume 2017-Decem, 2017, pp. 5999–6009. [arXiv:1706.03762](https://arxiv.org/abs/1706.03762).
- [6] R. Nogueira, W. Yang, K. Cho, J. Lin, Multi-Stage Document Ranking with BERT (2019). URL: <http://arxiv.org/abs/1910.14424>. [arXiv:1910.14424](https://arxiv.org/abs/1910.14424).
- [7] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, Association for Computational Linguistics,

- 2020, pp. 3982–3992. URL: <http://arxiv.org/abs/1908.10084>. doi:10.18653/v1/d19-1410. arXiv:1908.10084.
- [8] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, 2013. URL: <http://ronan.collobert.com/senna/>. arXiv:1301.3781.
- [9] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999, Association for Computing Machinery, Inc, 1999, pp. 50–57. doi:10.1145/312624.312649.
- [10] J. Pennington, R. Socher, C. D. Manning, GloVe: Global vectors for word representation, in: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014, pp. 1532–1543. doi:10.3115/v1/d14-1162.
- [11] A. M. Dai, Q. V. Le, Semi-supervised Sequence Learning, Advances in Neural Information Processing Systems 2015-Janua (2015) 3079–3087. URL: <http://arxiv.org/abs/1511.01432>. arXiv:1511.01432.
- [12] J. Howard, S. Ruder, Universal Language Model Fine-tuning for Text Classification, ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers) 1 (2018) 328–339. URL: <http://arxiv.org/abs/1801.06146>. arXiv:1801.06146.
- [13] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, Association for Computational Linguistics (ACL), 2018, pp. 2227–2237. doi:10.18653/v1/n18-1202. arXiv:1802.05365.
- [14] A. Radford, T. Salimans, Improving Language Understanding by Generative Pre-Training, OpenAI (2018) 1–12. URL: https://gluebenchmark.com/leaderboardhttps://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.
- [15] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding, Technical Report, 2019. arXiv:1906.08237.
- [16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018). arXiv:1810.04805.
- [17] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation (2016). arXiv:1609.08144.
- [18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, arXiv 21 (2019) 1–67. arXiv:1910.10683.
- [19] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei,

Language models are few-shot learners, 2020. [arXiv:2005.14165](https://arxiv.org/abs/2005.14165).