Knowledge Infused Representations Through Combination of Expert Knowledge and Original Input

Daniel Biermann(✉), Morten Goodwin, and Ole-Christoffer Granmo

Centre for Artificial Intelligence Research (CAIR), Department of ICT, University of Agder, Grimstad, Norway
daniel.biermann@uia.no

Abstract. Sophisticated applications in natural language processing, such as conversational agents, often need to be able to generalize across a range of different tasks to generate natural-feeling language. In this paper, we introduce a model that aims to improve generalizability with regard to different tasks by combining the original input with the output of a task-specific expert. Through a combination mechanism, we create a new representation that has been enriched with the information given by the expert. These enriched representations then serve as input to a downstream model. We test three different combination mechanisms in two combination paradigms and evaluate the performance of the new enriched representation in a simple encoder-decoder model. We show that even very simple combination mechanisms are able to significantly improve performance of the downstream model. This means that the encoded expert information is transported through the new enriched input representation, leading to a beneficial impact on performance within the task domain. This opens the way for exciting future endeavors such as testing performance on different task domains and the combination of multiple experts.

Keywords: Artificial neural networks · Natural language processing · Knowledge representation · Knowledge transfer

1 Introduction

In the field of natural language processing (NLP), conversational agents or chatbots are of ongoing interest. Challenges like the Amazon Alexa prize challenge[1] further incentivise research on chatbots in open-domain settings such as day-to-day conversation. A significant challenge in open-domain settings is the wide field of tasks these conversational agents encounter. For example, in a day-to-day conversation, a chatbot might need to simultaneously generate grammatically correct sentences while identifying different types of sentences (dialogue act

[1] https://developer.amazon.com/alexaprize.

E. Zouganeli et al. (Eds.): NAIS 2022, CCIS 1650, pp. 3–15, 2022.
https://doi.org/10.1007/978-3-031-17030-0_1

classification), recognizing intent (intent classification) and answering questions (question answering).

Transfer learning is the field of using the knowledge of an intelligent agent trained in one task for another task. It is of natural interest to the field of NLP as all tasks share the underlying concept of language. This mainly shows in the practice of pre-training models on large text corpora to generate contextualized word representations, i.e. ELMo [12]. Since the inception of the Transformer model [18], the Transformer's efficiency prompted a trend in research to improve performance by pre-training Transformer-based models of rapidly increasing size on vast sets of unlabeled data and fine-tuning them for a specific task. Prominent examples are the GPT architectures [1,13,14] as well as BERT architectures (e.g. [3,9,16]) and XLNet [20]. The problem with these architectures are the massive costs of pretraining. The costs have already reached regions in which only corporations like Google, Facebook, etc. can afford to train these large models from the ground up.

Next to the pretraining-finetuning approaches, Mixture-of-Expert (MoE) and other ensemble methods are of particular interest for transfer learning. The idea behind ensemble models is to combine an ensemble of distinct experts in a way that the different experts offset the weaknesses of the other experts and elevate the overall architecture to a better and more robust performance, possibly across different tasks.

In this paper we propose a new, ensemble-based architecture that combines task-specific expert output with the initial input representation to form a new expert-information-enriched representation to serve as input for a downstream task model. Meaning, we combine the output of an expert solving a specific task with the original input word embeddings. Our model utilizes, in contrast to other ensemble models, an already trained expert whose output shape differs significantly from the original input shape. Furthermore, we explore in our proposed architecture different combination methods that are based on attention and RNNs. Additionally, we explore these methods in a dimensional- and sequential combination paradigm.

2 Related Work

The idea to combine seperate experts has been explored since the 90's [7,8]. Early renditions of MoE models used a gating function to decide which expert output is further propagated. Recent MoE research pushed the concept of sparsely-activated models such as the Switch-Transformers [5], enabling efficient models with trillions of parameters. MoE models mainly aim at creating sparse models where each incoming example is processed by different parameters, thus, possibly training different parameter sets for different tasks. This is in contrast to dense networks in which the parameters are shared for each input. Our approach differs from these MoE models in that the experts are already trained and can have different architectures and output shapes. In MoE models, the experts often have the same architecture and output shape and have to be trained.

Using ensemble models to create new word embeddings has been the subject of previous research. [10] combined different word embeddings by ordinary least squares regression and by solving the orthogonal Procrustes problem while [21] creates word meta-embeddings by combining different word embeddings via different ensemble methods. Recently, [4] employed an attention network to combine semantic lexical information of knowledge graphs and pre-trained word embeddings in an ensemble method. The method proposed in our work differs from these previous approaches. The biggest difference is that the mentioned works aimed at creating general word embeddings instead of task specific embeddings. By task-specific embeddings we mean a vector representation that is infused with the output of an expert solving a specific task. Thus, the representations generated in this work are created with specific tasks is mind. Creating task specific embeddings allows for a more flexible use of the architecture as we can tailor the experts that we choose to combine to the downstream task. Additionally, we use Transformer-base attention mechanisms to combine the original input with the expert output. Rather than creating new general word embeddings, we infuse the original word embedding with focused task-specific information in form of the output of task-specific experts.

3 Methods

3.1 Model

Fundamentally, our architecture resembles a classic encoder-decoder model. The encoder consists of the pre-trained expert and the combination mechanism, and generates the new enriched word-knowledge representation. The decoder consists of a downstream task model that is to be trained to perform its downstream task.

In the encoder, we present the input embedding to the expert which subsequently calculates the output. The original input embedding and expert output are then concatenated and passed towards the combination mechanism. The combination mechanism calculates the expert-knowledge-enriched representation that has the same dimensionality as the original input embedding. The idea of enforcing the same dimensionality is to further support the modular structure of the architecture. This way, the expert combination process can be easily interjected between the original word embedding and the downstream model without having to change the downstream model. This input embedding is then used as input for the decoder. The general structure is outlined in Fig. 1.

In general, the expert and downstream model can be arbitrary models of arbitrary tasks with the experts already trained. The expert is regarded a finished model and is NOT trained in our architecture. The idea is to be able to make use of old already trained models and available pre-trained models to improve performance of the downstream model either in the same or a different task.

In this paper, we explore the simplest case of combining 1 expert that has the same task domain as the downstream model. We choose the Context-Aware Self Attention dialogue act classifier model (CASA) [15] as an expert. Compared to the original CASA model, we only use pre-trained Glove vectors [11] as word

embeddings for the expert and replace the CRF classifier with a softmax classifier with 1000 hidden units. We test different combination methods and paradigms that are described in more detail below.

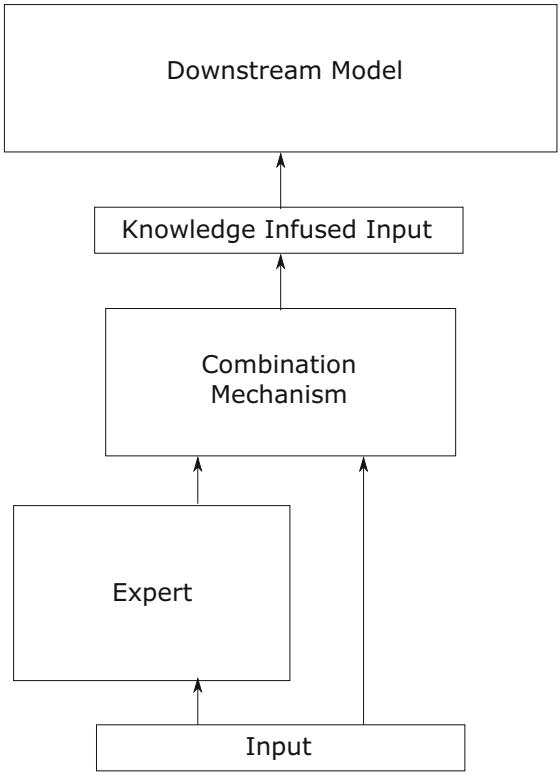


Fig. 1. Model architecture. Experts are pre-trained task-specific models. Downstream models are arbitrary, to-be-trained models. The combination mechanism combines the expert output and original input into a new enriched representation.

The downstream model consists of a single GRU (one-directional) layer [2] followed by a softmax classifier with 64 hidden units. We train the downstream model on the same task and dataset as the CASA expert.

When training the downstream model on the same task and data as the expert, we technically do not perform transfer learning as the task domains are the same. Nevertheless, by using a sophisticated, well-performing expert and a worse-performing, simple classifier we can test whether the task-knowledge infused in the enriched knowledge representation translates to a better performance in a simple model.

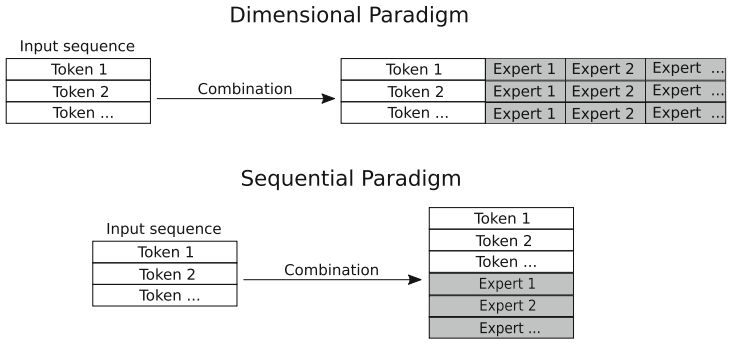


Fig. 2. Illustration of the dimensional and sequential combination paradigms.

3.2 Combination Paradigms

In our architecture we explore two different combination paradigms: Dimensional and sequential. These paradigms are illustrated in Fig. 2.

Dimensional Paradigm. In the dimensional paradigm, the expert output that has the number of classes as dimension is concatenated with the input embedding of each token in the input sequence, leading to the dimensionality $d_{emb} + d_{class}$. This concatenated vector is then presented to the combination mechanism as its input representation.

Sequential Paradigm. In the sequential paradigm, the expert output is appended to the list of tokens in the input sequence. For that, the output of the expert of dimension d_{class} is projected to the embedding dimension d_{emb} using a simple fully connected feedforward layer and added to the sequence. A sequence of length N becomes a sequence of length $N + 1$.

Thus, the combination mechanisms are presented with the challenge of reducing the dimensionality in the dimensional paradigm and reducing the sequence length in the sequential paradigm.

3.3 Combination Mechanisms

We test our model with three different combination methods. The first two mechanism are the scaled dot-product attention and multi-head attention introduced with the Transformer model [18] and the third consists of a simple recurrent network.

Mutli-head Attention. The first mechanism uses multi-head attention. Revisiting the attention definitions in [18] gives us:

$$\mathcal{A}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

$$\mathcal{M}(Q, K, V) = \text{Concat}(H_1, \ldots, H_h)W^O \tag{2}$$

$$H_i = \mathcal{A}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

where Q, K and V are query, key and value matrices with dimensionalities d_k, d_q and d_v, respectively. $\mathcal{A}$ and $\mathcal{M}$ denote the scaled-dot product and multi-head attention. The multi-head attention mechanism consists of multiple heads H_i that compute the scaled-dot product in parallel. Each head has their own Q, K and V matrices and produces outputs of dimension d_v/h with the number of heads h. The outputs are then concatenated and projected up to d_v via W^O.

In the dimensional paradigm we want d_v to be of the same dimension as the original input d_{emb} to reduce the concatenated dimensions back to the embedding dimension. While in principle the attention mechanism allows to rescale the dimension by choosing d_v, the multi-head attention requires that $d_k = d_q$ and d_v can be divided by the number of heads. This makes rescaling by d_v impracticable in our model as we can not always choose the output dimensions of our experts. For the dimensional paradigm, it is therefore beneficial to follow the general practice to set $d_k = d_q = d_v = d_{emb} + d_{class}$ and rescale by changing the dimension of W^O.

In case of the sequential paradigm, we do not want to change the dimension. We calculate the attention on the sequence $N + 1$ and drop the last sequence element.

Scaled Dot-Product Attention. Setting the number of heads in multi-head attention to $h = 1$ yields the scaled-dot product.

RNN. The third mechanism consist of a simple bi-directional GRU layer with its concatenated last hidden dimensions equaling the original embedding dimension. The hidden state after the last token in the sequence serves as the new knowledge infused representation. For the sequential paradigm, we require the RNN to be bi-directional as we have to drop the last hidden state. If the RNN were one-directional, dropping the last hidden state would also drop all the expert information.

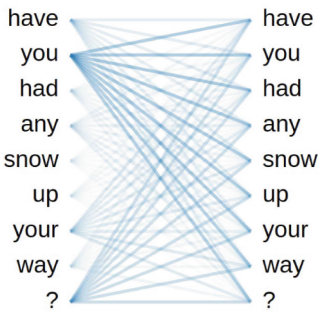
4 Experiments

We train the downstream DA classifier model for each combination method and paradigm. The results are shown in Table 2. As baseline, we have the simple classifier and CASA model that were each trained and evaluated with the unaltered GloVe embeddings as input. Additionally, we trained combination mechanism baseline models by removing the expert from the model. The purpose of this is

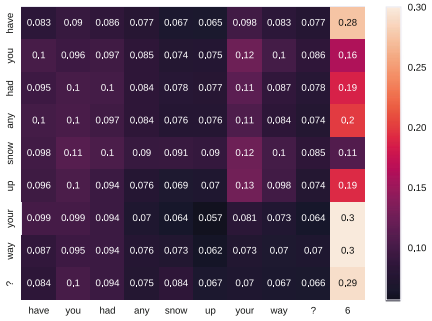
to get a better understanding whether any performance improvement is due to additional parameters the combination mechanism introduces to the model or the information of the expert.

Each model was trained until convergence with a patience of 30. The 5 best model iterations with regard to validation accuracy were saved. The results given in Table 2 show the averaged test accuracies.

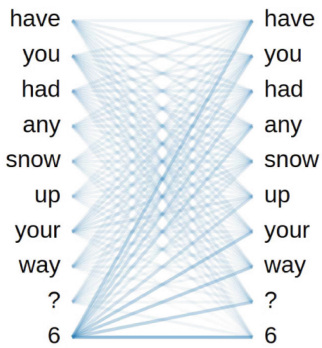
(a) Heatmap Dimensional



(b) Attention Visualization Dimensional



(c) Heatmap Sequential



(d) Attention Visualization Sequential

Fig. 3. Heatmap and attention visualization for the multi-head attention weights in both combination paradigms. The attention weights depicted have been averaged over all heads. Attention visualization created via *BertViz* [19]

4.1 Data

We train all models on the Switch-Board dialect corpus (SwDA) [6,17][2]. The dataset consists of conversations which contain sequences of sentences. We follow the train, validation and test splits given in the official paper.

After removing the non-verbal instances from the dataset, the corpus consists of $n_{class} = 41$ classes. The class frequency across the whole dataset is significantly imbalanced. To improve training, we calculate the cross-entropy loss with class weights. The class weights are inversely proportional to the frequency of the class.

We load the data in conversations. This means that the sentences within a conversation are always presented in the same order, thus retaining their contextual information. During training, we load the conversations in random order.

For the word embedding, we choose the $d_{emb} = 300$ dimensional GloVe vector trained on Wikipedia 2014 + Gigaword: 'glove.6B'.

4.2 Hyperparameters

The used hyperparameters are summarised in Table 1. The combination mechanism models share the same hyperparameters as the simple classifier as the combination mechanism itself is defined by d_{emb}. The learning rate was kept constant until epoch = 50 after which it was scaled by a factor $\frac{1}{\sqrt{epoch}}$. For the combination mechanism baseline models the learning was kept constant at 0.00001. No hyperparameter tuning was performed. The hyperparameters were chosen to represent standard values used in machine learning. The hyperparameters for the CASA classifier follow [15]

Table 1. List of hyperparameters.

Hyperparameters	Simple classifier
d_{emb}	300
Hidden GRU	8
Hidden softmax	64
Learning rate	0.0001*
	Multi-head attention
No. heads h	10/11

4.3 Results

As shown in Table 2, all combination models show a significant improvement in performance compared to the simple classifier. In addition, the combination models also show a significant improvement when compared to their baseline performance.

[2] This work uses the pre-cleaned dataset files provided in https://github.com/NathanDuran/Switchboard-Corpus.

Table 2. Dialogue act classification accuracies

Experiments	Test accuracy/%		
	No expert	Dimensional paradigm	Sequential paradigm
Baseline:			
Simple classifier	69.25		
CASA	75.03		
Combination mechanisms:			
Multi-head attention	70.80	75.03	74.78
Scaled dot-product attention	68.20	74.86	74.99
RNN	71.74	74.99	74.97

The simple classifier is able to reach an accuracy of 69.25. This low accuracy is expected as we chose a deliberately simple downstream model. We can also observe that the combination baseline models reach similar accuracies to the simple classifier. This solidifies that the significant performance improvement is not an artifact of the additional trainable parameters that the combination mechanism introduces. For the multi-head attention and RNN we only see small improvements to the accuracy. The performance worsens for the scaled dot-product. This suggests that a single application of the scaled dot-product might be too simple and has a detrimental effect on the information present in the pre-trained GloVe embeddings.

Nevertheless, when given outputs from an expert, all combination models in both combination paradigms significantly increase the performance and push the accuracy into the regime of the expert of ∼75%. This means that the information present in expert output is successfully infused into the new representation that we pass onto the downstream model. In case of the Multi-head attention mechanism in the dimensional paradigm, the performance equals the CASA baseline performance of 75.03%. This might indicate that the new representation has incorporated all information from the expert and carried it over to the downstream model so that it reaches equal performance. Whether the performance of the simple downstream model fed with the expert infused representations can exceed the performance of the expert or if the expert baseline represents a performance ceiling for the downstream model is subject of future work.

Figure 3 shows the visualization of the multi-head attention weights for an example sentence for both combination paradigms. The weights are visualized as a heatmap and using the *BertViz* visualization tool.

For the dimensional paradigm, the influence of the expert output can not be made visible by attention as we infuse every token with the expert knowledge. Thus, every token carries the same expert information. Nevertheless, it can be seen that for a question, a significant part of the attention is put on the '?' token as well as the 'you' token. In attention models, we usually see more variation in

the weights of single words instead of entire columns. This means that certain words carry over strongly into all new token representations. We suspect that this behavior is due to using only a single layer in the attention mechanism. Infusing all token representations with the same expert information might emphasize this effect as the combination of expert information and original token could combine into a 'universally good' or 'bad' representations. Thus, 'universally good' representations carry large weights for all new representations. The sequential attention weight heatmap does not show such a pronounced column wise attention proclivity. While the heatmap shows the significant influence of the expert output, it offers slightly more variation in weights across distinct words instead of columns (with the exception of the expert output column). This indicates that we have successfully created new word embeddings that have been infused with knowledge by paying attention to the relevant expert token. While the expert token dominates the attention weights, it can be seen that some tokens also pay attention to other tokens than the expert token. This means that the original word embedding also contributes to the new word embedding. Comparing the visualizations of the two paradigms makes the advantage of the sequential paradigms on explainability immediately obvious. While we have to speculate on what the effects of the expert are on the combination process in the dimensional paradigm, in the sequential paradigm, we can immediately see the effect of the expert output through attention itself.

Across the different paradigms the combination models perform similarly well and no clear paradigm or model outperforms the others. The multi-head attention reaches the best performance in the dimensional paradigm with an accuracy of 75.03 which is equal to the expert performance. Though, no sensible conclusion or insight can be gained from comparing the combination model accuracies as the differences between them are negligible. Apart from retaining the explainability of attention in the sequential paradigm, no clear preference of paradigms can be made with regard to performance.

4.4 Future Potential of Model

We expect that the performances will start to diverge once more sophisticated combination mechanism are employed. In our exploration, we deliberately limited our models to the simplest possible variants of the presented combination mechanisms. If the performance increase can be seen for the simplest models, it is a reasonable expectation that it will also work for more sophisticated models.

A preference of paradigm might emerge regarding computational cost as parameter space scales differently with increasing expert numbers for each paradigm. The dimensional paradigm grows faster in the trainable parameter space due to the query, key and value weight matrices that grow with increasing expert output dimensions. The sequential paradigm does not affect the query, key and value weight matrices but adds additional feedforward layers and computation calls for each expert. Nevertheless, this is an additive cost in model size for each expert instead of a multiplicative one. Thus, it can be expected that the

sequential paradigm might gain an advantage when combining larger numbers of experts.

5 Conclusion

We developed a simple ensemble based architecture that creates knowledge infused representations by combining the original input with the output of a pre-trained task-specific expert. We tested this infusion process for different combination methods and paradigms. The proof of concept that this architecture is able to create knowledge infused representation opens up several exciting future research directions. We saw that knowledge infused representations improved the performance of deliberately simple downstream models. This opens exciting opportunities to simplify training of new models as we can use already trained or pre-trained models to improve the performance of simpler models. In a way, this method can be understood as a combination of an ensemble model and a pretraining-finetuning approach.

In future work, we would like to train the downstream model and expert on different tasks to investigate the architectures true transfer learning capabilities. A natural next step would be to increase the number of experts and explore the architectures ability to perform multitask learning as well as investigate the scaling behavior of the two different combination paradigms. The exploration of more sophisticated combination models is also of interest. Of particular interest is also the question whether the performance of this approach is fundamentally capped by the performance of the experts or if the combination process is able to elevate the performance beyond the experts baseline performance. In contrast to the proof of principle investigation presented in this paper, a next step is a more systematic investigation to achieve the best performance and compare it with other state-of-the-art models.

Overall, the approach of infusing already trained expert knowledge into original pre-trained representations has the potential to offer great benefits to the fields of transfer learning. The ability to combine distinct experts into expert-sets that have been selected with a specific task in mind could offer great task-specific performance gains.

References

1. Brown, T., et al.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020). https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
2. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

4. Fang, L., Luo, Y., Feng, K., Zhao, K., Hu, A.: A knowledge-enriched ensemble method for word embedding and multi-sense embedding. IEEE Trans. Knowl. Data Eng. 1 (2022). https://doi.org/10.1109/TKDE.2022.3159539
5. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. arXiv preprint arXiv:2101.03961 (2021)
6. Godfrey, J., Holliman, E., McDaniel, J.: Switchboard: telephone speech corpus for research and development. In: [Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 517–520 (1992). https://doi.org/10.1109/ICASSP.1992.225858
7. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**(1), 79–87 (1991). https://doi.org/10.1162/neco.1991.3.1.79
8. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Comput. **6**(2), 181–214 (1994)
9. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
10. Muromägi, A., Sirts, K., Laur, S.: Linear ensembles of word embedding models. arXiv preprint arXiv:1704.01419 (2017)
11. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the Conference on EMNLP 2014–2014 Conference on Empirical Methods in Natural Language Processing (2014). https://doi.org/10.3115/v1/d14-1162
12. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana, June 2018. https://doi.org/10.18653/v1/N18-1202, https://aclanthology.org/N18-1202
13. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
14. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
15. Raheja, V., Tetreault, J.: Dialogue act classification with context-aware self-attention. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 3727–3733. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. https://doi.org/10.18653/v1/N19-1373, https://aclanthology.org/N19-1373
16. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
17. Stolcke, A., et al.: Dialogue act modeling for automatic tagging and recognition of conversational speech. Comput. Linguist. **26**(3), 339–373 (2000)
18. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
19. Vig, J.: A multiscale visualization of attention in the transformer model. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 37–42. Association for Computational Linguistics, Florence, Italy, July 2019. https://doi.org/10.18653/v1/P19-3007, https://www.aclweb.org/anthology/P19-3007

20. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019). https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf
21. Yin, W., Schütze, H.: Learning word meta-embeddings. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1351–1360 (2016)