IDANI: Inference-time Domain Adaptation via Neuron-level Interventions

TECHNION Israel Institute of Technology

Omer Antverg Eyal Ben-David Yonatan Belinkov

omer.antverg@cs.technion.ac.il eyalbd12@campus.technion.ac.il belinkov@technion.ac.il

Technion – Israel Institute of Technology

Abstract

Large pre-trained models are usually fine-tuned on downstream task data, and tested on unseen data. When the train and test data come from different domains, the model is likely to struggle, as it is not adapted to the test domain. We propose a new approach for domain adaptation (DA), using neuron-level interventions: We modify the representation of each test example in specific neurons, resulting in a counterfactual example from the source domain, which the model is more familiar with. The modified example is then fed back into the model. While most other DA methods are applied during training time, ours is applied during inference only, making it more efficient and applicable. Our experiments show that our method improves performance on unseen domains.

Method

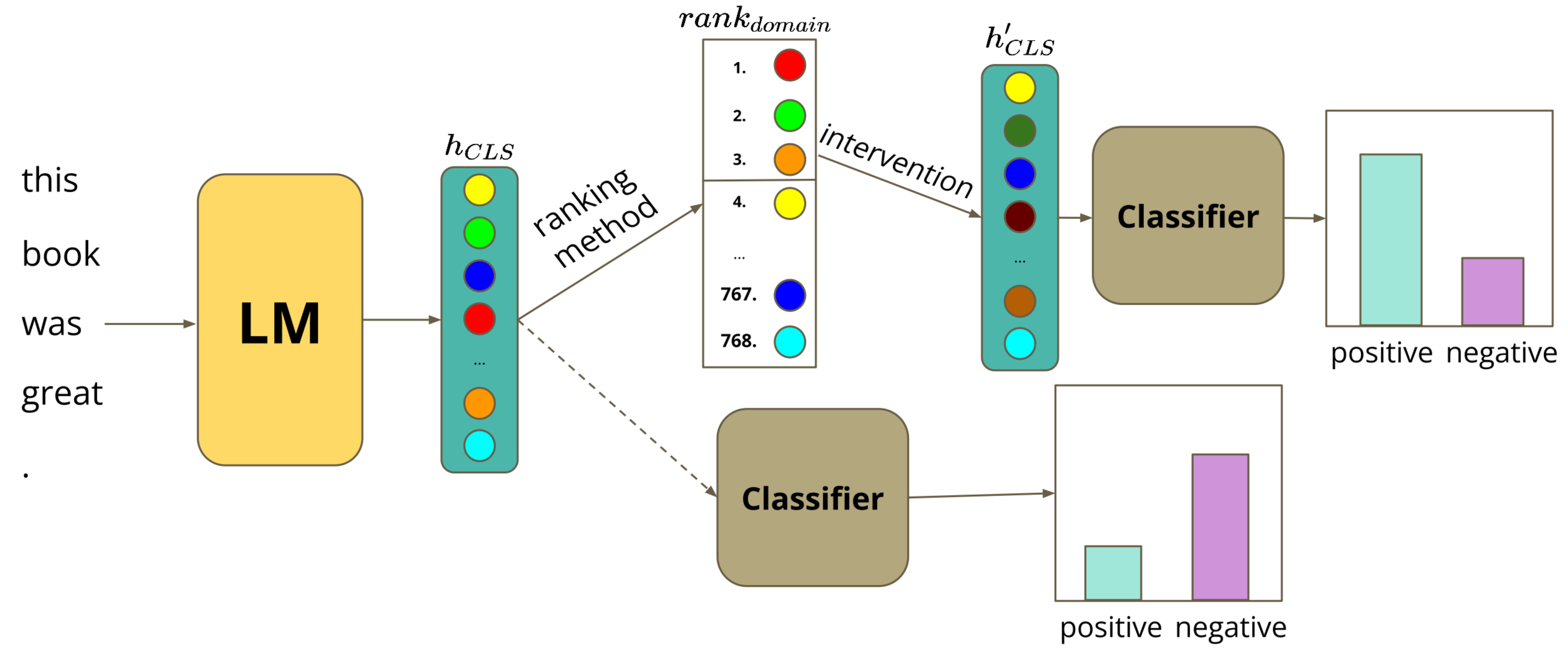

Figure 1. Illustration of our method.

Given:

- Model (M) with a classifier (f), fine-tuned on data from a source domain, $D_s = \{X_s\}$.
- Unlabeled data, $D_t = \{X_t\}$, from a target domain, only used for inference.

We make the representation of X_t **more similar** to that of X_s (regardless of the labels):

1. Process X_s and X_t through M, producing representations $H^s, H^t \subseteq \mathbb{R}^d$. Also compute $\bar{v}^s$ and $\bar{v}^t$, the element-wise mean representations of X_s and X_t.
2. Rank the representation's neurons by their relevance for domain information, i.e., the highest-ranked neuron holds the most information about the representation's domain.
3. For each $h^t \in H^t$, create a counterfactual $\tilde{h}^s$ by modifying h^t only in the k-highest ranked neurons $\{n_1, ..., n_k\}$, such that $\forall i \in \{1, ..., k\}$,
$$\tilde{h}^s_{n_i} = h^t_{n_i} + \alpha_{n_i}(\bar{v}^s_{n_i} - \bar{v}^t_{n_i}) \tag{1}$$
To allow stronger intervention on higher-ranked neurons, we scale the intervention with $\alpha \in \mathbb{R}^d$, a log-scaled sorted coefficients vector in the range $[0, \beta]$ such that $\alpha_{n_1} = \beta$ and $\alpha_{n_d} = 0$, where β is a hyperparameter [1]. Denote the new set of representations as $\tilde{H}^s$.
4. Feed representations from $\tilde{H}^s$ to the classifier f—without re-training f—to predict the labels.

References

[1] Omer Antverg and Yonatan Belinkov. On the pitfalls of analyzing individual neurons in language models. In *International Conference on Learning Representations*, 2022.

[2] Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James R. Glass. What is one grain of sand in the desert? analyzing individual neurons in deep NLP models. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 6309–6317. AAAI Press, 2019.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

Rankings

We work with two neuron-ranking methods:

- **Linear** [2]: Train a linear classifier on word representations to learn some task F. Then, use the trained classifier's weights to rank the neurons according to their importance for F.
- **Probeless** [1]: For every attribute label $z \in Z$, calculate $q(z)$, the mean vector of all representations of words that possess the attribute and the value z. Then, calculate the element-wise difference between the mean vectors,
$$r = \sum_{z,z' \in Z} |q(z) - q(z')| \tag{2}$$
and obtain a ranking by arg-sorting r, i.e., the first neuron in the ranking corresponds to the highest value in r.

Experiments

Datasets

- Binary sentiment analysis.
  - Airline, Books, DVD, Electronics, Kitchen.
- Natural language inference (NLI) - contradiction, entailement, neutral.
  - Fiction, Government, Slate, Telephone, Travel.
- Aspect prediction - binary token classification.
  - Device, Laptops, Restaurants, Service.

Setup - unsupervised domain adaptation (UDA)

- Train an algorithm on a single source domain.
- Test the algorithm on a (different) target domain.
- Low-resource scenario: 2000–3000 training examples from the source domain.

Model

- BERT-base-cased [3] fine-tuned on the training set of the source domain.
- We experiment with different k (number of modified neurons) and β (magnitude of the intervention) values.

Results

- IDANI generally improves results with default hyperparameters.
- With oracle hyperparameters, IDANI improves performance in almost all experiments.
- Probeless provides better performance than Linear.
- Improvements in aspect prediction and some source–target domain pairs in sentiment analysis are substantial (over 10 points; improvements in NLI are more modest). Details in the paper.

	Improved	Damaged	Neither	AVG Δ
$\Delta^P_{8,50}$	21	9	22	0.25
$\Delta^L_{8,50}$	23	7	22	0.25
Δ^P_O	51	0	1	1.77
Δ^L_O	50	0	2	0.93

Table 1. Number of experiments in which IDANI improved, damaged, or did not significantly affect the initial performance. Δ^P and Δ^L refer to Probeless and Linear respectively, while $\Delta_{8,50}$ and Δ_O refer to $\beta = 8, k = 50$ and oracle values.

Code

Code available at `https://github.com/technion-cs-nlp/idani`.

Analysis

Qualitative Analysis

- For each word in the dataset we record the change in results when classifying sentences containing the word (sentiment analysis) or when classifying the word itself (aspect prediction).
- When switching from the Airline domain to the DVD domain in the sentiment analysis task, those are mostly words that sound negative in an airline context, but may not imply a sentiment towards a movie (*terrorist*, *kidnapped*).
- In the aspect prediction task, those are mostly target domain related terms that are not likely to appear in the source domain.

Airline → DVD (Sentiment)	*immortal, insanely, terrorist, crossing, obsessive, buzz, kidnapped*
Laptops → Restaurant (Aspect)	*Food, soup, selection, sushi, food, atmosphere, menu, staff*
Restaurant → Laptops (Aspect)	*time, user, slot, speed, MAC, Acer, system, size, SSD, design*

Table 2. Words that are part of sentences for which accuracy has improved the most (sentiment analysis), and words for which F1 score has improved the most (aspect prediction), using IDANI.

Default Hyper-parameters are Not Optimal

- The selection of $\beta = 8, k = 50$ turns out as non optimal.
- There is no ideal value of k across all domain pairs (Fig. 2).
- Similarly, no one value of β works best in all cases (Fig. 3).

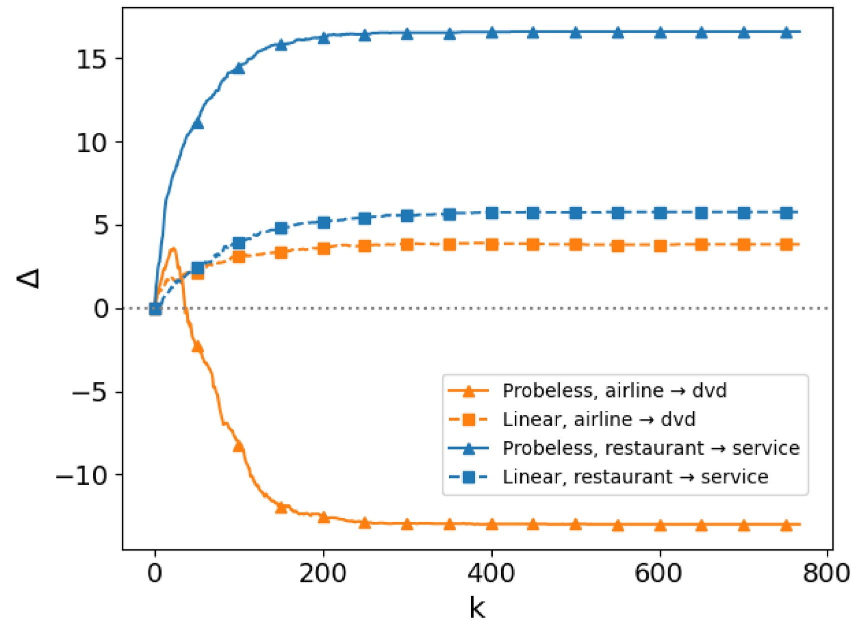

Figure 2. Results for different k values, using $\beta = 8$.

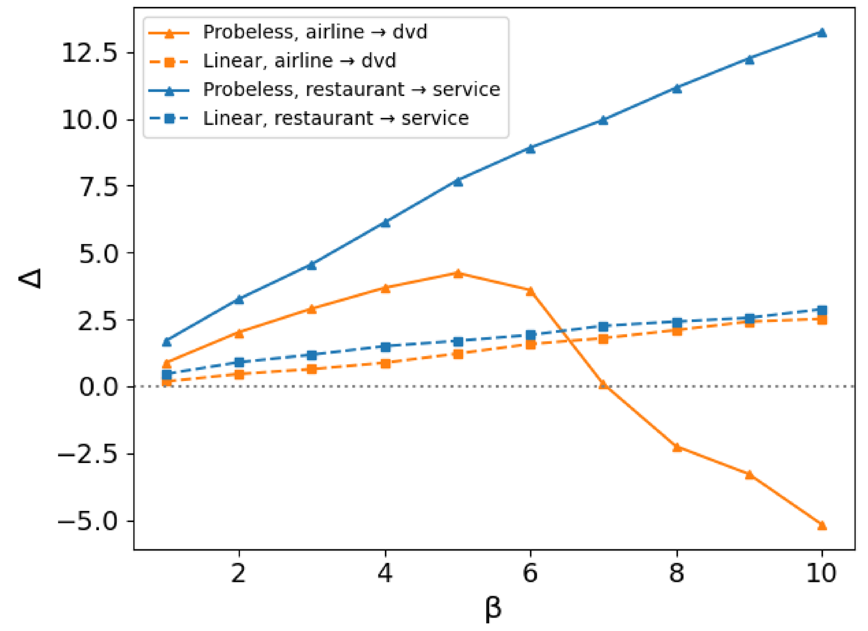

Figure 3. Results for different β values, using $k = 50$.

Conclusion

- We demonstrated the ability to leverage neuron-intervention methods to improve OOD performance.
- We showed that in some cases, IDANI can significantly help models to adapt to new domains.
- IDANI performs best with oracle hyperparameters, but even with the default ones we see overall positive results.
- We showed that IDANI indeed focuses on domain-related information, as the gains come mostly from domain-related information, such as domain-specific aspect terms.
- Importantly, IDANI is applied only during inference, unlike most other DA methods.

Acknowledgements

Research supported by the ISRAEL SCIENCE FOUNDATION (grant No. 448/20) and by an Azrieli Foundation Early Career Faculty Fellowship. We thank the anonymous reviewers for their insightful comments and suggestions.