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

Omer Antverg and **Eyal Ben-David** and **Yonatan Belinkov***
Technion - Israel Institute of Technology
{omer.antverg@cs.|eyalbd12@campus.|belinkov@}technion.ac.il

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. [1]

1 Introduction

A common assumption in NLP, and in machine learning in general, is that the training set and the test set are sampled from the same underlying distribution. However, this assumption does not always hold in real-world applications since test data may arrive from many (target) domains, often not seen during training. Indeed, when applied to such unseen target domains, the trained model typically encounters significant degradation in performance.

DA algorithms aim to address this challenge by improving models' generalization to new domains, and algorithms for various DA scenarios have been developed (Daume III and Marcu, 2006; Reichart and Rappoport, 2007; Ben-David et al., 2007; Schnabel and Schütze, 2014). This work focuses on unsupervised domain adaptation (UDA), the most explored DA setup in recent years, which assumes access to labeled data from the source domain and unlabeled data from both source and target domains. Algorithms for this setup typically use the target domain knowledge during *training*, attempting to bridge the gap between domains through representation learning (Blitzer et al., 2007; Ganin et al., 2016; Ziser and Reichart, 2018; Han and Eisenstein, 2019; David et al., 2020). Recently, Ben-David et al. (2021) and Volk et al. (2022) introduced an approach for *inference*-time DA, assuming no prior knowledge regarding the test domains but still modifying the training process to their gain.

In contrast to this line of work, we assume a more realistic scenario, in which the model was already trained on a source domain, and encounters unlabeled data from the target domain during inference time.

Given an example from a target domain, we would have liked to change it to a source domain example, so that the model would be more likely to perform well on it. Since this is difficult to achieve, we aim to change its representation in a fine-grained manner, such that we modify only information about the domain of the representation, without hurting other information. To do so, we take inspiration from work analyzing language models, which showed that linguistic properties are localized in certain neurons (dimensions in model representations) (Dalvi et al., 2019; Durrani et al., 2020; Torroba Hennigen et al., 2020; Antverg and Belinkov, 2022; Sajjad et al., 2021). We first rank the neurons by their importance for identifying the domain (source or target) of each example. Then, we modify target-domain representations only in the highest-ranked neurons, to change their domain to the source domain. Since the model was trained on examples from the source domain, we expect it to perform better on the modified representations. We name this method as Inference-time Domain Adaptation via Neuron-level Interventions (IDANI).

We follow a large body of previous work, testing

* Supported by the Viterbi Fellowship in the Center for Computer Engineering at the Technion.

[1] Our code is available at https://github.com/technion-cs-nlp/idani.

Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, pages 21 - 29
July 14, 2022 ©2022 Association for Computational Linguistics

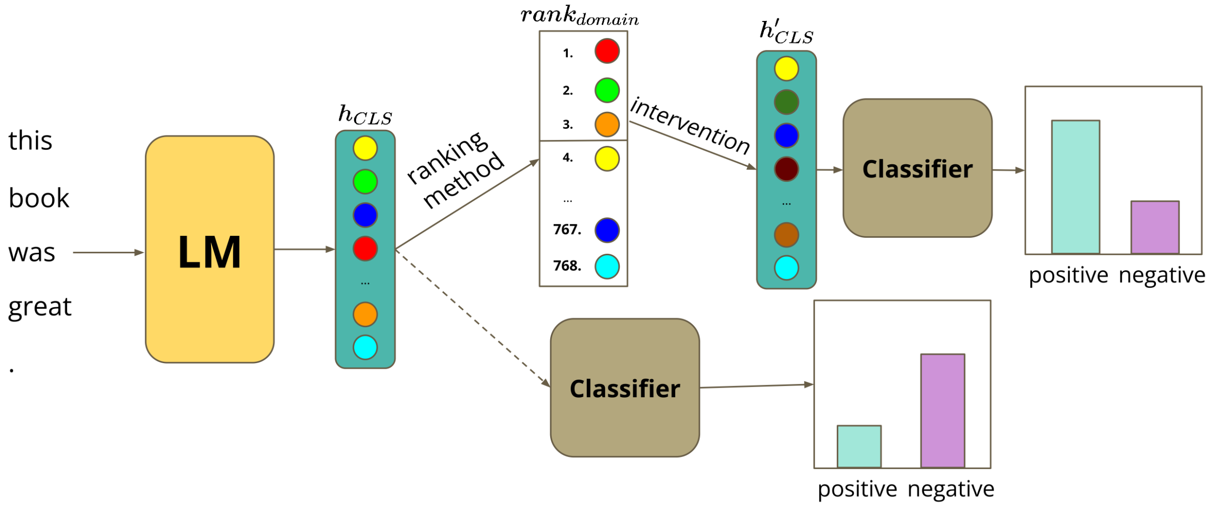

Figure 1: The language model—which was trained on some source domain, e.g., airline—creates a representation (CLS) for the review. Since the review is from a domain on which it was not trained, the model's classifier mistakenly classifies it as negative (bottom). In IDANI (top), the representation is fed into a neuron-ranking method. The k-highest ranked neurons are modified by an intervention, to change the domain of the review, and the new representation is fed into the classifier, which correctly classifies it as positive.

IDANI on a variety of well known DA benchmarks, for a total of two text classification tasks (sentiment analysis, natural language inference) and one sequence tagging task (aspect identification), across 52 source–target domain pairs. We demonstrate that IDANI can improve results in many of these cases, with some significant gains.

2 Method

Given a model M with a classification module f and hidden dimensionality d, which was fine-tuned on data from a source domain $D_s = \{X_s\}$, we receive unlabeled task data $D_t = \{X_t\}$ from a target domain for inference. As $s \neq t$, M's performance is likely to deteriorate when processing X_t compared to X_s. Thus, we would like to make the representation of X_t more similar to that of X_s (regardless of the labels). To do so, we apply the IDANI intervention method:

1. We process X_s and X_t through M, producing representations $H^s, H^t \subseteq \mathbb{R}^d$. We also compute $\bar{v}^s$ and $\bar{v}^t$, the element-wise mean representations of X_s and X_t.

2. We apply existing ranking methods to 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 (§ 2.1).[2]

3. For each $h^t \in H^t$, we would ideally like to have h^s, its source domain counterpart. Since h^s is impossible to get, we create a counterfactual $\tilde{h}^s$ that simulates it by modifying h^t only in the k-highest ranked neurons $\{n_1, ..., n_k\}$, such that $\forall i \in \{1, ..., k\}$,

$$\tilde{h}^s_{n_i} = h^t_{n_i} + \alpha_{n_i}(\bar{v}^s_{n_i} - \bar{v}^t_{n_i}) \qquad (1)$$

   To allow stronger intervention on neurons that are ranked higher, 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 (Antverg and Belinkov, 2022). We denote the new set of representations as $\tilde{H}^s$.

4. Representations from $\tilde{H}^s$ are fed into the classifier f—without re-training f—to predict the labels. Since $\tilde{H}^s$ is more similar to H^s than H^t is to H^s, we expect performance to improve. That is, for some scoring metric γ, we expect to have $\gamma(f(\tilde{H}^s)) > \gamma(f(H^t))$.

The process is illustrated in Fig. 1.

2.1 Ranking Methods

We consider two ranking methods for ranking the representations' neurons (step 2):

[2]Following previous work (Antverg and Belinkov, 2022), our method assumes that neurons with the same index carry similar information. While this is not necessarily true, we perform extrinsic (Table 1) and intrinsic evaluations (Table 2) that support this assumption.

LINEAR (**Dalvi et al., 2019**) This method trains a linear classifier on H^s and H^t to learn to predict the domain, using standard cross-entropy loss regularized by elastic net regularization (Zou and Hastie, 2005). Then, it uses the classifier's weights to rank the neurons according to their importance for domain information. Intuitively, neurons with a higher magnitude of absolute weights should be more important for predicting the domain.

PROBELESS The second ranking method is a simple one and does not rely on an external probe, and thus is very fast to obtain: it only depends on computing the mean representation of each domain ($\bar{v}^s$ and $\bar{v}^t$), and sorting the difference between them. For each neuron $i \in \{1, ..., d\}$, we calculate the absolute difference between the means:

$$r_i = |\bar{v}_i^s - \bar{v}_i^t| \tag{2}$$

and obtain a ranking by arg-sorting r, i.e., the first neuron in the ranking corresponds to the highest value in r. Antverg and Belinkov (2022) showed that for interventions for morphology information, this method outperforms LINEAR and another ranking method (Torroba Hennigen et al., 2020).

3 Experiments

3.1 Datasets

We experiment with two text classification tasks: sentiment analysis (classifying reviews to positive or negative (Blitzer et al., 2007)) and natural language inference (NLI; classifying whether two sentences entail or contradict each other (Bowman et al., 2015)), and a sequence tagging task: aspect prediction (identifying aspect terms within reviews (Hu and Liu, 2004; Toprak et al., 2010; Pontiki et al., 2014)). For each task, the model is trained on a single source domain and tested on different target domains. We explore a low-resource scenario, thus we use 2000–3000 examples from the source domain to form the training set.[3] For test, we use equivalent size data from the corresponding target domain. Further data details are in Appendix A.

3.2 Experiments

For each task and pair of source and target domains, we fine-tune a pre-trained BERT-base-cased model (Devlin et al., 2019) on the training set of the source domain and evaluate its in-domain performance on the dev set of the source domain.[4] We intervene on representations from the last layer of the model: word representations for the aspect prediction task, and CLS token representation for the other tasks. We then test the model's out-of-distribution (OOD) performance on the test set of the target domain, for different k (number of modified neurons) and β (magnitude of the intervention) values: We perform grid search where k is in the range $[0, d]$ ($d = 768$) and β is in the range $[1, 10]$. We experiment with both ranking methods described in § 2.1.

We consider the model's performance at $k = 0$ as its initial (unchanged) OOD performance (INIT), and report the difference between initial performance and performance using IDANI, with either PROBELESS (Δ^P) or LINEAR (Δ^L) rankings. A limitation of IDANI (which we further discuss later) is the inability to choose the best β and k for each domain pair. Following Antverg and Belinkov (2022) we report results for $\beta = 8, k = 50$ ($\Delta_{8,50}$), as well as oracle results (the best performance across all values, Δ_O). We consider the model's performance when fine-tuned on the target domain as an upper bound (UB). For all pairs, we repeat experiments using 5 different random seeds, and report mean INIT, $\Delta_{8,50}$, Δ_O and UB across seeds, alongside the standard error of the mean.

Since we assume that the model is exposed to target domain data only during inference, we cannot experiment with UDA methods, as they require access to the data during training. Furthermore, experimenting with *inference-time DA* approaches (Ben-David et al., 2021; Volk et al., 2022) is also not possible since they assume multiple source domains for training.

4 Results

Overall, we have 52 source to target domain adaptation experiments. Table 1 aggregates results across all experiments in three different categories: experiments where we can be confident that we improved the initial performance (i.e., the mean result across seeds is greater than the standard error), damaged it (mean lower than the negative standard error) or did not significantly affect it. Detailed results per each source–target domain pair are in Appendix B.

[3]For development data we split our training set in a ratio of 80:20, where the smaller portion is used for development.

[4]For all experimented models, we define a maximum sequence length value of 256 and use a training batch size of 16.

	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.

As seen, IDANI provides decent performance, improving results much more than damaging even with default hyperparameters ($\Delta^P_{8,50}$ and $\Delta^L_{8,50}$). With oracle hyperparameters (Δ^P_O and Δ^L_O) it improves performance in almost all experiments.

Some of these gains are quite impressive: In the aspect prediction task, we gain 18.8 and 14.4 F1 points when adapting the Restaurants source domain to the target domains Laptops and Service, respectively. In other domain pairs, the gain is marginal. On average we gain 4 points with Δ^P_O.

In sentiment analysis, the airline domain (A) is quite different from the others, leading to lower INIT (initial performance) scores when it is the source domain. Adapting from A using IDANI results in a gain of up to 4.9 accuracy points. When other domains are used as source domains, we see mostly marginal gains, as the upper bound is closer to the initial performance, leaving less room for improvement in this task (UB $-$ INIT is low).

In NLI, it seems harder to improve: the room for improvement is lower (3.3 F1 points on average), which may imply that domain information is not crucial for this task. Still, we do see some significant gains, e.g., an improvement of 2 F1 points when adapting from Slate to the Telephone domain.

Generally, across all tasks and domain pairs, PROBELESS provides better performance than LINEAR as $\Delta^P_O > \Delta^L_O$ in 47 of the 52 experiments (Appendix B). This is in line with the insights from Antverg and Belinkov (2022), who observed that PROBELESS was better than LINEAR when used for intervening on morphological attributes.

4.1 Qualitative Analysis

To analyze the benefits of IDANI, 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). We report the words with the greatest improvement in Table 2. 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.

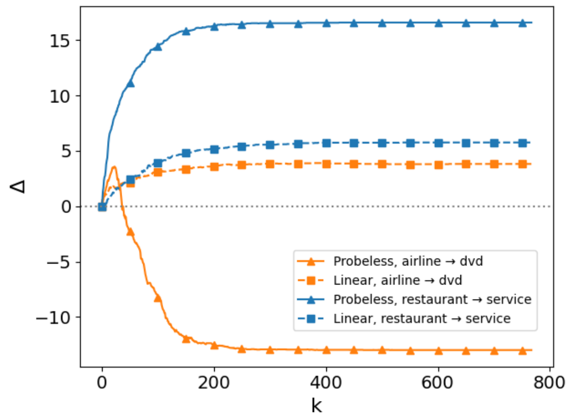

Figure 2: Results for different k values, using $\beta = 8$.

4.2 Default β and k are Not Optimal

While the potential for performance improvement with PROBELESS is high, the selection of $\beta = 8, k = 50$ turns out as non optimal, as $\Delta^P_{8,50}$ is well below Δ^P_O across our experiments. This is also true for $\Delta^L_{8,50}$ compared to Δ^L_O, but to a lesser degree.

Fig. 2 shows that a milder intervention—lower k value—would have been more ideal for the Airline $\rightarrow$ DVD scenario. Modifying too many neurons probably affects other encoded information—besides domain information—damaging the task performance. Thus, we might lean towards smaller k values. However, this is not always the case: Fig. 2 also shows that for the Restaurant $\rightarrow$ Service scenario in the aspect prediction task, PROBELESS' performance reaches a saturation point around the value of $k = 100$ neurons. Thus there is no ideal value of k across all domain pairs. A similar phenomenon with β is shown in Appendix C.

Therefore, hyperparameters should be task- and domain-dependent, but it is unclear how to define them for each domain pair. Yet, in most real-world cases some labeled data should be available or could be manually created. In such cases, the best approach would be to grid-search over the hyperparameters on the available labeled data, and use the selected values for the (unlabeled) test data.

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.

5 Conclusion

In this work, 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

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

References

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

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. Pada: A prompt-based autoregressive approach for adaptation to unseen domains. *arXiv preprint arXiv:2102.12206*.

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. 2007. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James R. Glass. 2019. 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, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6309–6317. AAAI Press.

Hal Daume III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126.

Eyal Ben David, Carmel Rabinovitz, and Roi Reichart. 2020. Perl: Pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Transactions of the Association for Computational Linguistics*, 8(0):504–521.

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

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

Chenggong Gong, Jianfei Yu, and Rui Xia. 2020. Unified feature and instance based domain adaptation for aspect-based sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online,* *November 16-20, 2020*, pages 7035–7045. Association for Computational Linguistics.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4237–4247. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM.

Quang Nguyen. 2015. The airline review dataset.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 27–35. The Association for Computer Linguistics.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623, Prague, Czech Republic. Association for Computational Linguistics.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2021. Neuron-level interpretation of deep nlp models: A survey. *ArXiv*, abs/2108.13138.

Tobias Schnabel and Hinrich Schütze. 2014. Flors: Fast and simple domain adaptation for part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 2(0):15–26.

Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. 2010. Sentence and expression level annotation of opinions in user-generated discourse. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 575–584. The Association for Computer Linguistics.

Lucas Torroba Hennigen, Adina Williams, and Ryan Cotterell. 2020. Intrinsic probing through dimension selection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 197–216, Online. Association for Computational Linguistics.

Tomer Volk, Eyal Ben-David, Ohad Amosy, Gal Chechik, and Roi Reichart. 2022. Example-based hypernetworks for out-of-distribution generalization. *arXiv preprint arXiv:2203.14276*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 1112–1122.

Yftah Ziser and Roi Reichart. 2018. Pivot based language modeling for improved neural domain adaptation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1241–1251, New Orleans, Louisiana. Association for Computational Linguistics.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.

A Data Details

We test IDANI on three different tasks: sentiment analysis, natural language inference, and aspect prediction. Further details of the training, development, and test sets of each domain are provided in Table 3.

Sentiment Analysis We follow a large body of prior DA work to focus on the task of binary sentiment classification. We experiment with the four legacy product review domains of Blitzer et al. (2007): Books (B), DVDs (D), Electronic items (E) and Kitchen appliances (K). We also experiment in a more challenging setup, considering an airline review dataset (A) (Nguyen, 2015; Ziser and Reichart, 2018). This setup is more challenging because of the differences between the product and service domains.

Natural Language Inference (Williams et al., 2018) This corpus is an extension of the SNLI dataset (Bowman et al., 2015). Each example consists of a pair of sentences, a premise and a hypothesis. The relationship between the two may be entailment, contradiction, or neutral. The corpus includes data from 10 domains: 5 are matched, with training, development and test sets, and 5 are mismatched, without a training set. Following Ben-David et al. (2021), we experiment only with the five matched domains: Fiction (F), Government (G), Slate (SL), Telephone (TL) and Travel (TR).

Since the test sets of the MNLI dataset are not publicly available, we use the original development sets as our test sets for each target domain, while source domains use these sets for development. Following prior work (Ben-David et al., 2021; Volk et al., 2022) we explore a low-resource supervised scenario, which emphasizes the need for a DA algorithm. Thus, we randomly downsample each of the training sets by a factor of 30, resulting in 2,000–3,000 examples per set.

Aspect Prediction The aspect prediction dataset is based on aspect-based sentiment analysis (ABSA) corpora from four domains: Device (D), Laptops (L), Restaurant (R), and Service (SE). The D data consists of reviews from Toprak et al. (2010), the SE data includes web service reviews (Hu and Liu, 2004), and the L and R domains consist of reviews from the SemEval-2014 ABSA challenge (Pontiki et al., 2014). The task is to identify aspect terms within reviews. For example, given a sentence "The price is reasonable, although the service is poor", both "price" and "service" should be identified as aspect terms.

Sentiment Classification			
Domain	**Training (src)**	**Dev (src)**	**Test (trg)**
Airline (A)	$1,700$	300	$2,000$
Books (B)	$1,700$	300	$2,000$
DVD (D)	$1,700$	300	$2,000$
Electronics (E)	$1,700$	300	$2,000$
Kitchen (K)	$1,700$	300	$2,000$
MNLI			
Domain	**Training (src)**	**Dev (src)**	**Test (trg)**
Fiction (F)	$2,547$	$1,972$	$1,972$
Government (G)	$2,541$	$1,944$	$1,944$
Slate (SL)	$2,605$	$1,954$	$1,954$
Telephone(TL)	$2,754$	$1,965$	$1,965$
Travel (TR)	$2,541$	$1,975$	$1,975$
Aspect			
Domain	**Training (src)**	**Dev (src)**	**Test (trg)**
Device (D)	$2,302$	255	$1,279$
Laptops (L)	$2,726$	303	800
Restaurants (R)	$3,487$	388	800
Service(SE)	$1,343$	149	747

Table 3: The number of examples in each domain of our four tasks. We denote the examples used when a domain is the source domain (src), and when it is the target domain (trg).

We follow the training and test splits defined by Gong et al. (2020) for the D and SE domains, while the splits for the L and R domains are taken from the SemEval-2014 ABSA challenge. To establish our development set, we randomly sample 10% out of the training data.

B Detailed Results

Results for all domain pairs are shown in Tables 4, 5 and 6. As described in § 4, IDANI can potentially significantly improve performance, shown by the results of Δ_O^P. Current hyperparameter values do not fulfill this entire potential, but still improve performance in most cases ($\Delta_{8,50}^P$).

C Performance for different β

While our default hyperparameter values, $\beta = 8$ and $k = 50$ improve performance in most cases, they are not optimal for all cases. Fig. 3 shows that when $k = 50$, the optimal β value for the Airline $\rightarrow$ DVD case is 5, whereas for Restaurants $\rightarrow$ Service it is actually better to use a greater β. Thus, it is not possible to find one value that would be optimal for all cases.

	A → B	A → D	A → E	A → K	B → A	B → D	B → E
INIT	77.4 ± 1.3	75.5 ± 2.2	85.2 ± 1.0	84.9 ± 0.9	83.7 ± 0.7	87.9 ± 0.3	90.4 ± 0.2
UB	88.0 ± 0.5	89.2 ± 0.5	92.4 ± 0.4	92.4 ± 0.2	88.0 ± 0.1	89.2 ± 0.5	92.4 ± 0.4
$\Delta^P_{8,50}$	-4.4 ± 4.8	-2.2 ± 5.4	-1.2 ± 2.4	-1.5 ± 1.9	0.5 ± 0.1	0.1 ± 0.1	-0.0 ± 0.0
$\Delta^L_{8,50}$	2.0 ± 1.0	2.1 ± 1.0	1.3 ± 0.4	1.1 ± 0.5	0.2 ± 0.1	0.1 ± 0.0	-0.0 ± 0.0
Δ^P_O	3.0 ± 1.3	4.9 ± 1.8	2.3 ± 0.8	2.3 ± 1.0	0.9 ± 0.2	0.3 ± 0.1	0.1 ± 0.0
Δ^L_O	2.9 ± 1.3	4.2 ± 1.8	2.3 ± 0.8	2.2 ± 0.9	0.3 ± 0.1	0.1 ± 0.0	0.0 ± 0.0

	B → K	D → A	D → B	D → E	D → K	E → A	E → B
INIT	87.8 ± 0.4	81.5 ± 0.3	89.4 ± 0.3	90.3 ± 0.2	88.1 ± 0.5	86.3 ± 0.4	86.8 ± 0.4
UB	92.4 ± 0.2	88.0 ± 0.1	88.0 ± 0.5	92.4 ± 0.4	92.4 ± 0.2	88.0 ± 0.1	88.0 ± 0.5
$\Delta^P_{8,50}$	0.1 ± 0.0	0.8 ± 0.2	0.1 ± 0.1	-0.0 ± 0.1	0.8 ± 0.3	0.0 ± 0.0	0.6 ± 0.2
$\Delta^L_{8,50}$	0.1 ± 0.0	0.5 ± 0.1	0.1 ± 0.0	0.1 ± 0.0	0.2 ± 0.1	0.0 ± 0.0	0.1 ± 0.1
Δ^P_O	0.4 ± 0.1	1.4 ± 0.3	0.3 ± 0.1	0.3 ± 0.1	1.4 ± 0.5	0.2 ± 0.0	1.0 ± 0.3
Δ^L_O	0.2 ± 0.0	0.8 ± 0.1	0.2 ± 0.1	0.1 ± 0.0	0.5 ± 0.2	0.1 ± 0.0	0.3 ± 0.1

	E → D	E → K	K → A	K → B	K → D	K → E	AVG
INIT	86.5 ± 0.2	93.2 ± 0.3	83.9 ± 0.4	87.0 ± 0.2	86.4 ± 0.1	92.2 ± 0.2	86.2 ± 0.7
UB	89.2 ± 0.5	92.4 ± 0.4	88.0 ± 0.1	88.0 ± 0.5	89.2 ± 0.5	92.4 ± 0.2	90.0 ± 0.4
$\Delta^P_{8,50}$	0.2 ± 0.1	0.2 ± 0.2	0.7 ± 0.2	0.1 ± 0.1	0.2 ± 0.1	0.1 ± 0.0	-0.2 ± 1.7
$\Delta^L_{8,50}$	-0.1 ± 0.1	-0.0 ± 0.0	0.1 ± 0.1	0.1 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.4 ± 0.4
Δ^P_O	0.4 ± 0.1	0.4 ± 0.2	1.2 ± 0.3	0.2 ± 0.0	0.5 ± 0.0	0.2 ± 0.0	1.1 ± 0.6
Δ^L_O	0.1 ± 0.0	0.2 ± 0.1	0.5 ± 0.2	0.1 ± 0.0	0.2 ± 0.0	0.0 ± 0.0	0.8 ± 0.6

Table 4: Sentiment analysis results (accuracy).

	F → G	F → SL	F → TL	F → TR	G → F	G → SL	G → TL
INIT	70.2 ± 0.8	63.7 ± 0.8	67.4 ± 1.3	65.6 ± 0.8	59.9 ± 0.8	62.1 ± 0.5	64.9 ± 0.9
UB	73.8 ± 0.4	62.6 ± 0.9	68.3 ± 0.4	69.9 ± 0.3	67.6 ± 0.9	62.6 ± 0.9	68.3 ± 0.4
$\Delta^P_{8,50}$	0.5 ± 0.5	0.4 ± 0.4	0.1 ± 0.4	-0.2 ± 0.4	0.8 ± 0.2	-0.2 ± 0.2	0.4 ± 0.3
$\Delta^L_{8,50}$	0.1 ± 0.2	0.0 ± 0.1	0.3 ± 0.2	0.1 ± 0.1	0.7 ± 0.4	-0.2 ± 0.1	0.1 ± 0.1
Δ^P_O	1.2 ± 0.4	0.9 ± 0.3	0.9 ± 0.3	0.7 ± 0.2	1.8 ± 0.6	0.4 ± 0.1	1.2 ± 0.2
Δ^L_O	0.6 ± 0.2	0.6 ± 0.2	0.8 ± 0.3	0.5 ± 0.2	1.5 ± 0.5	0.2 ± 0.0	0.9 ± 0.2

	G → TR	SL → F	SL → G	SL → TL	SL → TR	TL → F	TL → G
INIT	68.8 ± 0.2	62.0 ± 1.6	71.1 ± 1.4	63.7 ± 1.2	67.0 ± 1.2	63.6 ± 0.5	69.7 ± 0.4
UB	69.9 ± 0.3	67.6 ± 0.9	73.8 ± 0.4	68.3 ± 0.4	69.9 ± 0.3	67.6 ± 0.9	73.8 ± 0.4
$\Delta^P_{8,50}$	-0.0 ± 0.1	0.8 ± 0.4	-0.5 ± 0.2	1.1 ± 0.4	-0.1 ± 0.1	-0.6 ± 0.3	-1.1 ± 0.6
$\Delta^L_{8,50}$	-0.1 ± 0.1	0.4 ± 0.2	0.1 ± 0.1	0.7 ± 0.1	0.1 ± 0.2	0.2 ± 0.1	-0.2 ± 0.1
Δ^P_O	0.5 ± 0.1	1.5 ± 0.4	0.3 ± 0.1	2.0 ± 0.5	0.5 ± 0.1	0.7 ± 0.2	0.7 ± 0.2
Δ^L_O	0.2 ± 0.1	1.4 ± 0.4	0.3 ± 0.1	1.4 ± 0.2	0.6 ± 0.1	0.6 ± 0.1	0.3 ± 0.0

	TL → SL	TL → TR	TR → F	TR → G	TR → SL	TR → TL	AVG
INIT	61.6 ± 0.5	64.9 ± 0.5	60.0 ± 1.0	71.5 ± 0.7	61.3 ± 0.6	63.3 ± 1.1	65.1 ± 0.9
UB	62.6 ± 0.9	69.9 ± 0.3	67.6 ± 0.9	73.8 ± 0.4	62.6 ± 0.9	68.3 ± 0.4	68.4 ± 0.7
$\Delta^P_{8,50}$	-0.3 ± 0.4	-0.5 ± 0.4	-0.1 ± 0.5	-0.1 ± 0.2	0.1 ± 0.2	0.4 ± 0.3	0.0 ± 0.4
$\Delta^L_{8,50}$	0.5 ± 0.2	-0.4 ± 0.3	0.3 ± 0.5	0.3 ± 0.3	0.0 ± 0.1	0.3 ± 0.3	0.2 ± 0.2
Δ^P_O	1.2 ± 0.1	0.7 ± 0.1	1.7 ± 0.4	0.7 ± 0.2	0.8 ± 0.2	1.2 ± 0.3	1.0 ± 0.3
Δ^L_O	1.1 ± 0.2	0.6 ± 0.1	1.0 ± 0.4	0.7 ± 0.2	0.6 ± 0.2	0.8 ± 0.3	0.7 ± 0.2

Table 5: MNLI results (macro-F1).

	D → L	D → R	D → S	L → D	L → R	L → S	R → D
INIT	50.9 ± 0.8	36.9 ± 1.1	40.5 ± 0.9	47.6 ± 0.2	35.3 ± 0.8	36.3 ± 0.5	46.2 ± 0.9
UB	85.5 ± 0.3	83.4 ± 0.2	81.2 ± 0.2	67.1 ± 0.5	83.4 ± 0.2	81.2 ± 0.2	67.1 ± 0.5
$\Delta^P_{8,50}$	-1.2 ± 0.6	-3.0 ± 1.2	-2.2 ± 1.0	0.9 ± 0.1	3.6 ± 0.7	1.0 ± 0.3	1.3 ± 0.3
$\Delta^L_{8,50}$	-0.2 ± 0.1	-0.4 ± 0.3	-0.1 ± 0.2	0.2 ± 0.1	0.3 ± 0.2	0.2 ± 0.1	0.1 ± 0.1
Δ^P_O	0.3 ± 0.2	0.6 ± 0.3	0.2 ± 0.2	1.4 ± 0.1	6.7 ± 1.0	1.9 ± 0.4	2.1 ± 0.5
Δ^L_O	0.2 ± 0.1	0.3 ± 0.2	0.4 ± 0.1	0.7 ± 0.0	1.5 ± 0.3	0.5 ± 0.2	0.7 ± 0.2

	R → L	R → S	S → D	S → L	S → R	AVG
INIT	44.1 ± 1.1	33.2 ± 0.9	49.1 ± 0.3	44.9 ± 0.5	55.6 ± 0.6	43.4 ± 0.8
UB	85.5 ± 0.3	81.2 ± 0.2	67.1 ± 0.5	85.5 ± 0.3	83.4 ± 0.2	79.3 ± 0.4
$\Delta^P_{8,50}$	9.5 ± 0.8	11.2 ± 0.7	0.6 ± 0.2	-2.1 ± 0.4	-4.2 ± 0.7	1.3 ± 0.7
$\Delta^L_{8,50}$	2.2 ± 0.5	2.4 ± 0.6	0.0 ± 0.1	-0.5 ± 0.2	-0.7 ± 0.4	0.3 ± 0.3
Δ^P_O	14.4 ± 0.9	18.8 ± 0.9	0.9 ± 0.2	0.3 ± 0.2	0.3 ± 0.2	4.0 ± 0.5
Δ^L_O	5.7 ± 0.9	6.8 ± 0.7	0.3 ± 0.1	0.2 ± 0.1	0.2 ± 0.1	1.5 ± 0.4

Table 6: Aspect prediction results (binary-F1).

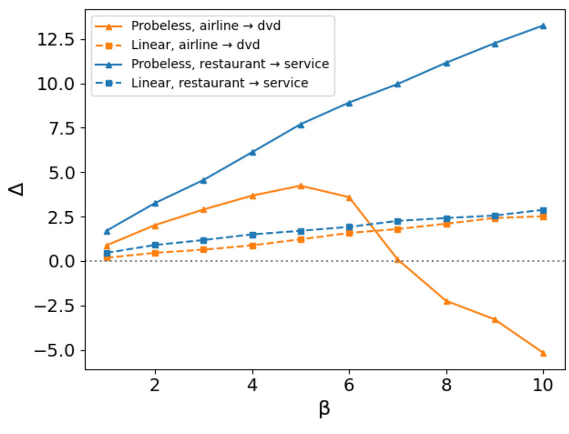

Figure 3: Results for different β values, using $k = 50$.