



HAL
open science

On the hidden negative transfer in sequential Transfer Learning for domain adaptation from news to tweets

Sara Meftah, Nasredine Semmar, Youssef Tamaazousti, Hassane Essafi, Fatiha Sadat

► To cite this version:

Sara Meftah, Nasredine Semmar, Youssef Tamaazousti, Hassane Essafi, Fatiha Sadat. On the hidden negative transfer in sequential Transfer Learning for domain adaptation from news to tweets. The Second Workshop on Domain Adaptation for NLP, Association for Computational Linguistics, Apr 2021, Kiev, Ukraine. cea-04572482

HAL Id: cea-04572482

<https://cea.hal.science/cea-04572482v1>

Submitted on 10 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

On the *Hidden* Negative Transfer in Sequential Transfer Learning for Domain Adaptation from News to Tweets

Sara Meftah*, Nasredine Semmar*, Youssef Tamaazousti*, Hassane Essafi*, Fatiha Sadat⁺

*CEA-List, Université Paris-Saclay, F-91120, Palaiseau, France

⁺UQÀM, Montréal, Canada

{firstname.lastname}@cea.fr, sadat.fatiha@uqam.ca

Abstract

Transfer Learning has been shown to be a powerful tool for Natural Language Processing (NLP) and has outperformed the standard supervised learning paradigm, as it takes benefit from the pre-learned knowledge. Nevertheless, when transfer is performed between less related domains, it brings a negative transfer, *i.e.* it hurts the transfer performance. In this research, we shed light on the *hidden* negative transfer occurring when transferring from the News domain to the Tweets domain, through quantitative and qualitative analysis. Our experiments on three NLP tasks: Part-Of-Speech tagging, Chunking and Named Entity recognition reveal interesting insights.

1 Introduction

High performing NLP neural tools often require huge volumes of annotated data to produce powerful models and prevent over-fitting. Consequently, in the case of social media content (informal texts) such as Tweets, it is difficult to achieve the performance of state-of-the-art neural models on News (formal texts).

The last few years have witnessed an escalated interest in studying Transfer Learning (TL) for neural networks to overcome the problem of the lack of annotated data. TL aims at performing a task on a target dataset using features learned from a source dataset (Pan and Yang, 2009). TL has been proven to be effective for a wide range of applications (Zamir et al., 2018; Long et al., 2015; Moon and Carbonell, 2017), especially for low-resourced domains.

However, it has been shown in many works in the literature (Rosenstein et al., 2005; Ge et al., 2014; Ruder, 2019; Gui et al., 2018a; Cao et al., 2018; Chen et al., 2019; Wang et al., 2019; O’Neill, 2019) that, when source and target domains are less

related (*e.g.* languages from different families), sequential transfer learning may lead to a negative effect on the performance, instead of improving it. This phenomenon is referred to as *negative transfer*. Precisely, negative transfer is considered when transfer learning is harmful for the target task/dataset, *i.e.* the performance when using transfer learning algorithm is lower than that with a solely supervised training on in-target data (Torrey and Shavlik, 2010).

Several works (Gui et al., 2017, 2018b; Meftah et al., 2018a,b; März et al., 2019) have shown that sequential transfer learning from the News resource-rich domain to the Tweets low-resource domain enhances the performance of sequence labelling of Tweets. Hence, following the above definition of negative transfer, transfer learning from News to Tweets does not beget a negative transfer. Contrariwise, in this work, we rather consider the *hidden negative transfer*, *i.e.* the percentage of predictions which were correctly tagged by random initialisation, but using transfer learning falsified.

In this work, we take a step towards identifying and analysing the impact of transfer from News to Tweets. Precisely, we perform an empirical analysis to investigate the *hidden* negative transfer. First, we show in section.5.1 that, the final gain brought by TL can be separated into two categories: *positive transfer* and *negative transfer*. We define *positive transfer* as the percentage of tokens that were wrongly predicted by random initialisation, but the TL changed to the correct ones. In comparison, *negative transfer* represents the percentage of words which were tagged correctly by random initialisation, but using TL gives wrong predictions. Then, in section.5.2, we study the impact of pretraining state on negative and positive transfer. Finally, in section.5.3, we provide some qualitative examples of negative transfer. Our experiments on three NLP tasks (Part-Of-Speech tagging (POS), Chunking

(CK) and Named Entity recognition (NER)) reveal interesting insights.

The remainder of the paper is organised as follows. We first present, briefly, the sequence tagging neural model (§2). Then, we describe the sequential transfer learning method (§3), followed by a short presentation of the involved datasets and tasks (§4.1). Then, we report the results of our analysis to highlight the hidden negative transfer occurring when transferring from News to Tweets (§5). Finally, we wrap up with a conclusion and future work (§6).

2 Sequence Tagging Neural Architecture

We perform experiments on 3 Sequence Tagging (ST) tasks: Part-Of-Speech tagging (**POS**), Chunking (**CK**) and Named Entity Recognition (**NER**). Given an input sentence of n successive tokens $S = [w_1, \dots, w_n]$, the goal of a ST model is to predict the tag $c_i \in \mathcal{C}$ of every w_i , with \mathcal{C} being the tag-set. We use a common ST neural model. It includes three main components. First, we have a WRE (Word Representation Extractor) to build, for each word w_i , a final representation \mathbf{x}_i combining two hybrid representations; a word-level embedding (denoted Υ^{word}) and a character-level embedding based on a bidirectional-Long Short-Term Memory (biLSTMs) encoder (denoted Υ^{char}). Second, the \mathbf{x}_i representation is fed into a Features Extractor (FE) (denoted Φ) based on a single-layer BiLSTMs network, to produce a hidden representation h_i which constitutes the input of the Classifier (denoted Ψ): a fully-connected (FC) layer used for classification. Formally, given w_i , the predictions are obtained using the following equation: $w_i: \hat{y}_i = (\Psi \circ \Phi \circ \Upsilon)(w_i)$. With Υ ensuring the concatenation of Υ^{char} and Υ^{word} .¹

3 Transfer Learning Method

We use a simple sequential TL method to transfer knowledge from the News domain to the Tweets-domain. It consists in learning a source model on the source task with enough data from the News domain, then transferring a part of the learned parameters to initialise the target model, which is further fine-tuned on the target task with few training examples from the Tweets domain.

¹Note that – for simplicity –, we define \hat{y}_i only as a function of w_i , but in reality \hat{y}_i is a function of all words in the sentence, thanks to the biLSTMs component.

Specifically, in this work we perform the TL following three simple yet effective steps: 1) The source model is learnt using a large annotated dataset from the source domain. 2) We transfer to the target model the first set of parameters (Υ and Φ) of the source model, while the second set of parameters (Ψ) of the target model is randomly initialised. Then, 3) the target model is further fine-tuned on the small target data-set.

4 Experimental Settings

4.1 Data-sets

We conduct experiments on TL from English News (source-domain) to English Tweets (target-domain) on three tasks (Datasets statistics are given in Table.1):

- **POS tagging:** we use the Wall Street Journal (**WSJ**) part of Penn-Tree-Bank (PTB) as a source-dataset. Regarding the target-datasets, we used three Tweets datasets: **TPoS** (Ritter et al., 2011), **ARK** (Owoputi et al., 2013) and **TweeBank** (Liu et al., 2018).
- **CK:** for the source dataset, we use the **CONLL2000** shared task’s English data-set (Tjong Kim Sang and Buchholz, 2000). Regarding the target dataset, we use **TChunk** Tweets data-set (Ritter et al., 2011) (the same corpus as TPoS).
- **NER:** regarding the source domain, we make use of the English newswire dataset **CONLL-03** from the CONLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003). target domain, we conduct our experiments on **WNUT2017** dataset (Derczynski et al., 2017).

4.2 Implementation Details

In the standard word-level embeddings, tokens are converted to lower-case while the character-level component still retains access to the capitalisation information. We set the randomly initialised character embedding dimension at 50, the dimension of hidden states of the character-level biLSTM at 100 and used 300-dimensional word-level embeddings. Word-level embeddings were pre-loaded from publicly available GloVe vectors pre-trained on 42 billions words collected through web crawling and containing 1.9M different words (Pennington et al., 2014). These embeddings are also updated during training. For the **FE** component, we

Task	Classes	Sources	Eval. Metrics	Splits (train - val - test)
POS: POS Tagging	36	WSJ	Top-1 Acc.	912,344 - 131,768 - 129,654
CK: Chunking	22	CONLL-2000	Top-1 Exact-match F1.	211,727 - n/a - 47,377
NER: Named Entity Recognition	4	CONLL-2003	Top-1 Exact-match F1.	203,621 - 51,362 - 46,435
POS: POS Tagging	17	TweeBank	Top-1 Acc.	24,753 - 11,742 - 19,112
CK: Chunking	18	TChunk	Top-1 Exact-match F1.	10,652 - 2,242 - 2,291
NER: Named Entity Recognition	6	WNUT	Top-1 Exact-match F1.	62,729 - 15,734 - 23,394

Table 1: Statistics of the datasets we used to train our models. **Top:** datasets of the source domain. **Bottom:** datasets of the target domain.

Method	POS (Accuracy %)			CK (Accuracy %)	NER (F1 %)
	TPoS	ArK	TweeBank	TChunk	WNUT
Random Initialisation	86.82	91.10	91.66	85.96	40.36
Transfer Learning	89.57	92.09	93.23	88.86	41.92

Table 2: Results on POS, CK and NER of Tweets using Transfer Learning vs Random initialisation.

use a single layer biLSTM (token-level feature extractor) and set the number of units to 200. In all of our experiments, both pretraining and fine-tuning were performed using the same training settings, *i.e.* SGD with momentum and early stopping, and mini-batches of 16 sentences, and a fixed learning rate of 1.5×10^{-2} . Throughout this thesis, all our models are implemented with the PyTorch library (Paszke et al., 2017).

5 Analysis

First, in order to have an idea about the final impact of TL compared to randomly initialised models, we provide in Table.2 the performance of Random Initialisation and Transfer Learning. Clearly, TL enhances the performance across all data-sets and tasks. In the following sub-sections, we attempt to analyse thoroughly these results by showing that the impact of TL is two fold, *positive transfer* and *negative transfer*.

5.1 Quantifying Negative Transfer

Let us consider the gain \mathcal{G}_i brought by transfer learning compared to random initialisation for the dataset i . \mathcal{G}_i is defined as the difference between positive transfer \mathcal{PT}_i and negative transfer \mathcal{NT}_i : $\mathcal{G}_i = \mathcal{PT}_i - \mathcal{NT}_i$. Where positive transfer \mathcal{PT}_i represents the percentage of tokens that were wrongly predicted by random initialisation, but transfer learning changed to the correct ones. *negative transfer* \mathcal{NT}_i represents the percentage of words which were tagged correctly by random initialisation, but using transfer learning gives wrong predictions.

\mathcal{PT}_i and \mathcal{NT}_i are defined as follows: $\mathcal{PT}_i = \frac{N_i^{corrected}}{N_i}$ and $\mathcal{NT}_i = \frac{N_i^{falsified}}{N_i}$. Where

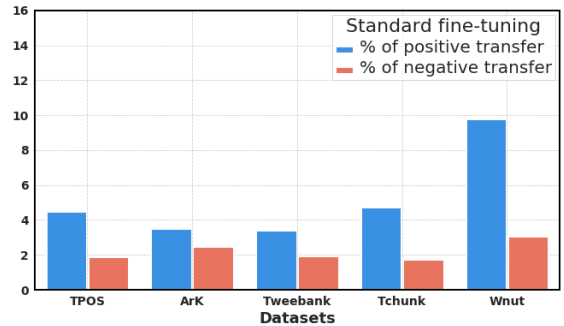


Figure 1: Impact on predictions made by TL compared to Random initialisation. Positive Transfer stands for the percentage of predictions that were wrong in the training from scratch scheme but the TL changed to the correct ones, and Negative Transfer stands for the percentage of predictions which the random model tagged correctly, but the TL falsified.

N_i the total number of tokens in the validation-set of the dataset i . $N_i^{corrected}$ is the number of tokens from the validation-set of the dataset i , that were wrongly tagged by the the model trained from scratch but are correctly predicted by the model using transfer learning. And $N_i^{falsified}$ is the number of tokens from the validation-set of the dataset i , that were correctly tagged by the the model trained from scratch but are wrongly predicted by the model using transfer learning.

We show in Figure.1 the results on English Tweets datasets TpoS, ArK and TweeBank for POS; WNUT for NER; and Tchunk for CK. First tagged with the classic training scheme (Random) and then *using TL*. Blue bars show the percentage of *positive transfer*, and red bars give the percentage of *negative transfer*. We observe that even though TL approach is effective, since the resulting *positive transfer* is higher than *negative transfer* in all cases,

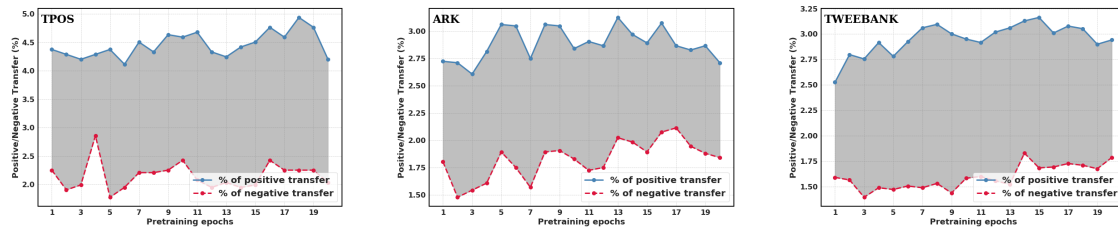


Figure 2: Positive transfer curves (blue) and negative transfer curves (red) on Tweets data-sets, according to different pretraining epochs. Transparent Gray highlights the final gain brought by TL.

this last mitigates the final gain brought by TL. For instance, on Tchunk dataset, TL corrected $\sim 4.7\%$ of predictions but falsified $\sim 1.7\%$, which reduces the final gain to $\sim 3\%$.

5.2 The impact of pretraining state on Negative Transfer

So far in our experiments we used the pretrained parameters from the best model trained on the source dataset. In simple words, we picked the model at the epoch with the highest performance on the source validation-set. In this analysis, we study when pretrained parameters are ready to be transferred. Specifically, we pick the pretrained weights at different pretraining epochs; that we call the pretraining states. Then, we assess the performance when transferring each. In Figure.2, we plot for each target dataset, the curves of positive transfer (blue curves) and negative transfer (red curves) brought by initialisation with pretrained weights from different pretraining epochs compared to random initialisation. Clearly, both negative and positive transfer increase with pretraining epochs. More important, we can observe that for *TweeBank* and *ArK* datasets the negative transfer increases rapidly in the last pretraining epochs. However, for *TPoS* dataset, the negative transfer stays almost stable throughout pretraining epochs. This phenomenon could be explained by the fact that *TPoS* shares the same PTB tag-set as *WSJ*, whereas *TweeBank* and *ArK* use different tag-sets. Consequently, in the last states of pretraining, the pretrained parameters become well-tuned to the source dataset and specific to the source tag-set, leading to an increase of negative transfer and thus a drop in transfer performance.

5.3 Qualitative Examples of Negative Transfer:

We illustrate in Table 3² concrete examples of words whose predictions were falsified when using transfer learning compared to random initialisation. Among mistakes we have observed:

- **Tokens with an upper-cased first letter:** In news (formal English), only proper nouns start with an upper-case letter inside sentences. Consequently, in the transfer learning scheme, the pre-trained units fail to slough this pattern which is not always respected in social media. Hence, we found that most of the tokens with an upper-cased first letter are mistakenly predicted as proper nouns (PROPN) in POS, *e.g.* Award, Charity, Night, etc. and as entities in NER, *e.g.* Father, Hey, etc., which is consistent with the findings of Seah et al. (2012); negative transfer is mainly due to conditional distribution differences between source and target domains.
- **Contractions** are frequently used in social media to shorten a set of words. For instance, in *TPoS* dataset, we found that “s” is in most cases predicted as a “possessive ending (pos)” instead of “Verb, 3rd person singular present (vbz)”. Indeed, in formal English, “s” is used in most cases to express the possessive form, *e.g.* “company’s decision”, but rarely in contractions that are frequently used in social media, *e.g.* “How’s it going with you?”. Similarly, “wont” is a frequent contraction for

²Classes significations: nn=N=noun=common noun / nnp=pnoun=proprn=proper noun / vbz=Verb, 3rd person singular present / pos=possessive ending / prp=personal pronoun / prp\\$=possessive pronoun / md=modal / VBP=Verb, non-3rd person singular present / uh!=intj=interjection / rb=R=adverb / L=nominal + verbal or verbal + nominal / E=emoticon / \\$=numerical / P=pre- or postposition, or subordinating conjunction / Z=proper noun + possessive ending / V=verb / adj=adjective / adp=adposition

DataSet							
TPoS	Award [◊] nn nnp	's vbz pos	its* prp prp\$	Mum nn uh	wont* MD VBP	id* prp nn	Exactly uh rb
ArK	Charity [◊] noun pnoun	I'M* L E	2pac [×] pnoun \$	2 [×] P \$	Titans* Z N	wth [×] ! P	nvr [×] R V
TweeBank	amazin [•] adj noun	Night [◊] noun propn	Angry [◊] adj propn	stangs propn noun	#Trump propn X	awsome [•] adj intj	bout [•] adp verb
TChunk	luv [×] b-vp i-intj	**ROCKSTAR** THURSDAY	b-np O	ONLY i-np b-np	Just [◊] b-advp b-np	wyd [×] b-np b-intj	id* b-np i-np
Wnut	Hey [◊] O b-person	Father [◊] O b-person	& [×] O i-group	IMO [×] O b-group	UN O b-group	Glasgow b-location b-group	Supreme b-person b-corporation

Table 3: **Examples of falsified predictions by standard fine-tuning scheme when transferring from News-domain to Tweets-domain.** Line 1: Some words from the validation-set of each data-set. Line 2: Correct labels predicted by the classic supervised setting (Random-200). Line 3: Wrong labels predicted by TL setting. Mistake type: ◊ for words with first capital letter, • for misspelling, * for contractions, × for abbreviations.

“will not”, e.g. “i wont get bday money lool”, predicted as “verb” instead of “modal (MD)” by transfer learning. The same for “id”, which stands for “I would”.

- **Abbreviations** are frequently used in social media to shorten the way a word is standardly written. We found that transfer learning scheme stumbles on abbreviations predictions, e.g. *2pac* (Tupac), *2* (to), *ur* (your), *wth* (what the hell) and *nvr* (never) in ArK dataset; and *luv* (love) and *wyd* (what you doing?) in TChunk dataset.
- **Misspellings:** Likewise, we found that the transfer learning scheme often gives wrong predictions for misspelt words, e.g. *awsome*, *bout*, *amazin*.

6 Conclusion

Our analysis on the *hidden* negative transfer from News-domain to Tweets-domain reveals interesting insights: 1) Even if using TL improves the performance on Tweets Sequence labelling, an inherent negative transfer may minimise the final gain; and 2) the negative transfer increases with the number of pretraining epochs. This study opens a set of promising directions. We plan to 1) Extend our experiments by investigating the impact of the model’s hyper-parameters (size, activation functions, learning rate, etc.). 2) Investigate the impact of the similarity between source and target datasets and source and target training datasets size on the negative transfer. 3) Tackle the negative transfer

problem, by identifying automatically biased neurons in the pretrained model and proceed to a pruning of the most biased ones before fine-tuning. 4) Explore negative transfer on Transformers-based pretrained models, such as BERT, XLNet, etc.

References

- Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2018. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2724–2732.
- Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. 2019. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In *Advances in Neural Information Processing Systems*, pages 1908–1918.
- Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.
- Liang Ge, Jing Gao, Hung Ngo, Kang Li, and Aidong Zhang. 2014. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4):254–271.
- Lin Gui, Ruifeng Xu, Qin Lu, Jiachen Du, and Yu Zhou. 2018a. Negative transfer detection in transductive transfer learning. *International Journal of Machine Learning and Cybernetics*, 9(2):185–197.
- Tao Gui, Qi Zhang, Jingjing Gong, Minlong Peng, Di Liang, Keyu Ding, and Xuan-Jing Huang. 2018b. Transferring from formal newswire domain with hypernet for twitter pos tagging. In *Proceedings of the*

- 2018 Conference on Empirical Methods in Natural Language Processing, pages 2540–2549.
- Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. 2017. Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2411–2420.
- Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A Smith. 2018. Parsing tweets into universal dependencies. 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 965–975.
- Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105.
- Luisa März, Dietrich Trautmann, and Benjamin Roth. 2019. Domain adaptation for part-of-speech tagging of noisy user-generated text. 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 3415–3420.
- Sara Meftah, Nasredine Semmar, and Fatiha Sadat. 2018a. A neural network model for part-of-speech tagging of social media texts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Sara Meftah, Nasredine Semmar, Fatiha Sadat, and Stephan Raaijmakers. 2018b. Using neural transfer learning for morpho-syntactic tagging of south-slavic languages tweets. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 235–243.
- Seungwhan Moon and Jaime G Carbonell. 2017. Completely heterogeneous transfer learning with attention-what and what not to transfer. In *IJCAI*, volume 1, pages 1–2.
- James O’Neill. 2019. Learning to avoid negative transfer in few shot transfer learning.
- Olutobi Owoputi, Brendan O’Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 380–390.
- Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics.
- Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. 2005. To transfer or not to transfer. In *In NIPS’05 Workshop, Inductive Transfer: 10 Years Later*. Citeseer.
- Sebastian Ruder. 2019. *Neural Transfer Learning for Natural Language Processing*. Ph.D. thesis, NATIONAL UNIVERSITY OF IRELAND, GALWAY.
- Chun-Wei Seah, Yew-Soon Ong, and Ivor W Tsang. 2012. Combating negative transfer from predictive distribution differences. *IEEE transactions on cybernetics*, 43(4):1153–1165.
- Erik F Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 127–132.
- Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147.
- Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.
- Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. 2019. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11293–11302.
- Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722.