CSC2528 TERM PAPER

DOMAIN ADAPTATION IN NATURAL LANGUAGE PROCESSING

January 4, 2019

Krishnapriya Vishnubhotla
Department of Computer Science
University of Toronto

Contents

Abstract

Statistical learning methods rely on having large amounts of training data, either labelled or unlabelled. Traditional supervised methods assume the training and test data come from the same underlying probability distribution, but this assumption rarely holds in real world scenarios. Transfer learning is the study of methods that allow us to transfer knowledge across different but related tasks, domains and distributions. It is an extensively studied topic in machine learning, and specifically for natural language processing. This work looks at the major approaches proposed for transfer learning tasks, how they've been adapted in the context of neural networks and deep learning, and compares their performance for a specific application: part-of-speech tagging. We'll also take a brief look at a few related tasks that have gained a lot of attention recently, such as multi-task learning and few-shot learning.

1 Introduction

Natural Language Processing (NLP) systems, when used out in the wild, often result in performance below what is expected or observed during experimental testing. This is because statistical classifiers assume that both the training and test data come from a common underlying distribution (Li, 2012), but oftentimes the training data is too restricted/specialised to provide an accurate estimate of this. The problem relates to one of the core challenges in designing a computational language understanding system: natural language is highly variable and sparse (Goldberg, 2017). The number of sentences that can be framed using the English language could never possibly be enumerated, and it therefore becomes extremely challenging to assemble a representative "training set for the English language".

Nevertheless, we can try. Generalizability of a statistical model is a highly desirable property, and much research has been done on preventing overfitting to the train set. The property of a model that allows it to draw conclusions on data it hasn't encountered before is called it's *inductive bias* (Baxter, 2000). For example, the inductive bias of a linear regression model is that there exists a linear relationship between the input and the output. A rote learner, therefore, lacks any inductive bias. It is a term closely related to the above discussion on the distribution of data - (Torrey and Shavlik, 2010) define inductive bias of a model as "the set of assumptions about the true distribution of the training data". Clearly, the better the inductive bias of a model, the better it's generalizability.

Consider this: most part-of-speech tagging systems were trained on the Wall Street Journal (WSJ) corpus, containing news articles from the financial domain. A commonly cited example in literature is that of the word "monitor" (Li, 2012). In the WSJ corpus, the word most probably is used as a verb, but in another domain, say Amazon electronic product reviews, it would be used as a noun. How would a PoS tagger know this, if it has only seen "monitor" as a verb? Intuitively, we expect the tagger to make decisions based on sentence structure and context words, and not just surface features. Thus, achieving a good inductive bias for NLP tasks implies a good understanding of syntax and semantics. This has been the focus of recent work on improving transferability of deep neural networks for NLP - obtaining generalizable, dense representations for words and sentences that work well across a range of tasks (Peters et al., 2018). Unfortunately, this transferability hasn't moved beyond the first layer, and we still need quite a bit of labelled data to achieve good performance on a particular task.

Rather than aim for the lofty goal of building a general natural language understanding model, techniques have been studied that facilitate transfer of knowledge and parameters from a source domain (our training set) to a target domain (our test set). Formally, the study of methods

that allow learning across domains, tasks and distributions is called transfer learning (Pan et al., 2010). The key motivation is that one can transfer knowledge from a source domain for which we have annotated data, to a target domain where such data is scarce. Another way of looking at this is that the source data affects the inductive bias of the model in the target space. A related idea is multi-task learning (Collobert and Weston, 2008), where one simultaneously optimises a model for several different, but related, objectives. Transfer learning, by contrast, aims to achieve a high performance only on the target task. Some other related techniques that apply to the low-data scenario include zero-shot/few-shot learning, data augmentation methods, and building task-independant architectures.

A plethora of methods have been proposed for transfer learning in text applications. Many of these focus and evaluate on certain tasks: sentiment analysis (Blitzer et al., 2007) (Glorot et al., 2011), Named Entity Recognition (NER) (Lee et al., 2017), part-of-speech tagging (PoS) (Blitzer et al., 2006), machine translation (Chu et al., 2017) etc. The recent surge in neural methods for NLP has also led to a surge in corresponding transfer learning methods. Before we look at these, let us first formally define a few terms that will be used throughout the rest of the paper.

1.1 Formal Definitions

We will use here the notation defined in (Pan et al., 2010) that is also followed in most transfer learning papers since.

- **Domain**: A domain **D** has two components: a feature space χ, and a marginal probability distribution $P(X)$. $P(X)$ here is the underlying distribution of our training data. The feature space varies based on how we define our features - for a binary bag-of-words representation of a text document, the feature space would be the set of all possible binary term vectors.

- **Task**: Given a domain $\mathbf{D} = \{\chi, P(X)\}$, a task T defines a label space γ and a predictive function $f : \chi \rightarrow \gamma$. This predictive function is learnt from the data and constitutes our model.

For a supervised learning task, we have a set of n training examples, denoted by $D = \{(x_i, y_i) \in \chi \times \gamma : i \in (1, .., n)\}$. Our predictive function $f(.)$ is then equivalent to $P(y|x)$ for an $(x, y) \in D$. We will denote our source domain data and target domain data with D_s and D_t respectively. Similarly, the source and target tasks will be referred to by T_s and T_t.

With the above notations, we can refine our definition of transfer learning as techniques that utilise the knowledge in D_s and T_s to improve the learning of T_t in D_t, where either $D_t \neq D_s$ or $T_t \neq T_s$. Note that, if both D and T are the same for the source and target domains, then it reduces to a general supervised learning problem.

1.2 Categories of Transfer Learning

Depending on the relationship between D_s, D_t, T_t and T_s, we end up with slightly different scenarios of transfer learning, with a different set of algorithms proposed for each:

1. $\chi_s \neq \chi_t$
   This is the case when the feature spaces of our domains are different. In NLP, this commonly occurs when we're dealing with documents written in different languages (cross-lingual adaptation).

2. $P(X_s) \neq P(X_T)$
   Here, the feature spaces are the same but the underlying data distributions are different. Our initial example of PoS tagging with the WSJ and bio-medical texts falls into this category, and is referred to as *domain adaptation*. Thus, our source and target tasks are the same, but we're dealing with datasets from different domains.

3. $P(Y_s|X_s) \neq P(Y_t|X_t)$
   Both the source and target distributions share the same output labels, but the conditional distributions of these classes is different, i.e, the classes are unbalanced. Several class balancing techniques like SMOTE and random undersampling exist to deal with these, though they are far from perfect.

4. $P(Y_s) \neq P(Y_t)$
   This case usually occurs alongisde case 3, where the supervised task itself is different for the source and target domains. Most of the recent research on neural transfer learning falls under this scenario. An application in NLP is where the source task involves binary classification, whereas the target task is a 10-way, perhaps more fine-grained, classification.

The focus of this paper will be mostly constrained to case 2, domain adaptation (DA). Algorithms proposed for this fall under one of three categories: feature representation based (Daume III, 2007) (Collobert and Weston, 2008) (Blitzer et al., 2006), instance based (Xu et al., 2011) (Jiang and Zhai, 2007) (Ruder and Plank, 2017) , and prior based (Chelba and Acero, 2006) (Finkel and Manning, 2009). In the context of neural NLP, we have work on learning general, off-the-shelf

vector representations of words, such as word2vec (Mikolov et al., 2013). Language modelling is seen as a task that utilises most of the syntactic and semantic features of words, and is therefore used to achieve more informative representations (Devlin et al., 2018) (Peters et al., 2018). More specifically for DA, autoencoders have emerged as a popular method for achieving domain invariant representations (Yu and Jiang, 2016) (Chen et al., 2012).

In the following sections, we will first look at popular DA algorithms proposed before the explosion of deep learning based NLP methods. We will then look more closely at how domain adaptation functions in the context of neural network approaches to NLP. Due it's recent popularity, we'll also take a brief look at multi-task learning (Changpinyo et al., 2018) and few-shot learning in NLP (Srivastava et al., 2018). In Section 8, we present performance numbers for popular DA algorithms proposed for part-of-speech (PoS) tagging, and Section 9 concludes the survey.

2 Domain Adaptation in NLP

2.1 What do we mean by different domains?

In the previous section, we looked at formal definitions for a domain and task. Intuitively, in NLP applications, different domains means that our datasets are generated by different sources, i.e, written by different people, relating to different topics, or are in different languages. For example, two different people may use different adjectives to praise the same product, or the same person may use different adjectives to praise two different products.

2.1.1 Multilingual corpora

Adapting algorithms for multiple languages has been a longstanding focus in natural language processing. Much of the initial work was done in the context of Machine Translation (MT). The M1 model, the first of IBM's statistical alignment models, is basically defined as a statistical bilingual dictionary that captures word correlation across languages (Pinto et al., 2009). This model was subsequently adapted for cross-lingual text classification, information retrieval and inference. It is a field that has achieved notable results in the last few years, with the development of cross-lingual embeddings that can be inferred even with no parallel data.

2.2 Main Approaches

For a general supervised classification problem, let us assume χ to be our feature space and γ to be our set of labels. We also have a set of training instances $\{(x_i, y_i) \in \chi \times \gamma : i \in (1, .., n)\}$, where (x_i, y_i) are drawn from the underlying probability distribution $p(x, y)$. We are trying to recover this unknown distribution so we can predict the label for unlabelled input instances. Discriminative models directly model the distribution $p(y|x)$. Assuming a fixed set of parameters θ for the model family, our aim now becomes finding the following:

$$\theta^* = argmax_\theta \int_\chi \sum_{y \in \gamma} p(x, y) log p(y|x; \theta) dx$$

Since we do not know the true underlying probability $p(x, y)$, we estimate it using the training data. This observed, empirical probability distribution is denoted by $p_s(x, y)$. When our test data now comes from a different distribution $p_t(x, y)$, our estimation of the optimal parameters θ^* is no longer optimal.
Using Bayes rule, one can factorize $p(x, y)$ as $p(x, y) = p(y|x)p(x)$ or as $p(x, y) = p(x|y)p(y)$. Domain adaptation methods can be broadly classififed into three categories based on which terms they modify in the above equations in order to bring $p_s(x, y)$ closer to $p_t(x, y)$.

- **Prior-based methods**
  These methods place different priors over the parameters or the labels to make the estimated $p_s(y|x; \theta)$ closer to $p_t(y|x)$. A prior over the label space $p(y)$ is usually used in generative models such as the naive Bayes.

- **Feature Representation based methods**
  These methods change the underlying feature representation space χ such that $p_t(y|x)$ is similar to $p_s(y|x)$.

- **Instance-based methods**
  These set of methods revolve around selecting and/or weighting instances in the source domain that are similar to those in the target domain. Thus, they focus on the distribution $p(x)$, or selecting x such that $p(y|x)$ is roughly the same in both the source and target domains.

The adaptation algorithms in all three categories also vary in terms of the amount of labelled target data they require, dependence on the underlying classification model, and computational complexity. These facets of the algorithms will become clearer as we look at them in individual detail in the following sections.

3 Prior-based methods

Let us now look at domain adaptation methods that work by placing priors over parameters θ of the model $p(y|x;\theta)$. Discriminative classifiers usually assume a gaussian prior with zero mean over the parameters. This prior acts as a regularizer for the model, preventing overfitting. A distribution over parameters can be used to control how features behave in the target and source domains - those that have a similar effect in both domains will share the same prior value, whereas they will be pushed apart for features that behave differently. These methods generally assume a small amount of labelled data in the target domain.

1. (Chelba and Acero, 2006) proposed a prior-based adaptation method for the automatic capitalisation task. Their classifier was a Maximum Entropy Markov Model (MEMM), and involves training two models sequentially on the source and target data. First, the source model is trained with a zero-mean gaussian mean over the parameters. The target model parameters are then initialised with a gaussian prior centered at the corresponding value estimated by the source model. New features in the target data are initialised with a zero-mean prior as usual. The target model is then trained to maximise likelihood of data as usual.

2. (Finkel and Manning, 2009) extended the above model to work with multiple source and target domains. Their algorithm utilises a hierarchical structure of priors, with a high-level one that is domain-independent, and a lower level of domain-specific priors. The evidence present in each domain for a feature influences the domain-specific prior, which in turn pushes the top-level prior to the same value. This top-level prior is the default value for all the domains. Thus, in every domain, for a particular feature, the prior value defaults to that of the top-level, and conversely, domain-level priors affect the value of the top-level priors. The hierachical structure of this algorithm results in a large number of parameters that need to be estimated, and the computational complexity is therefore quite high.

3.1 Prior over label space

(Chan and Ng, 2006) took a different approach towards adapting classifiers for word sense disambiguation in different domains. The authors assume that $p(x|y)$ remains the same across both domains, but the prior probabilities of the labels $p(y)$ (here, word senses) differ. In other words, the occurrence proportions of different word senses will vary depending on the corpus in consideration. Assuming no labelled data in the target domain, the Expectation-Maximization

(EM) algorithm is applied to estimate prior probabilities of the labels for both naive Bayes and logistic regression classifiers.

Learning under the above assumption has also been studied in machine learning as the class imbalance problem. A popular solution is to re-sample instances from the soure data such that this re-sampled data has the same class distribution as the target dataset.

3.2 Fine-tuning classifiers

A popular supervised domain adaptation method, especially in computer vision, is fine-tuning of classifiers. Starting from support vector machines (Yang et al., 2007) to deep convolutional neural networks (Sharif Razavian et al., 2014), both transfer learning and domain adaptation have been hugly succesful for vision aplications. A classifier is first trained on a related, auxiliary task that has lablled data, and it's parameters are fine-tuned from those values using labelled data from the target domain. Specifically in the context of neural networks, the use of pre-trained ResNet models (He et al., 2016) has become pervasive in almost every application. A new classification layer is added on top of the pre-trained block, and, depending on the amount of labelled data available, the top-k layers of the network are trained (Yosinski et al., 2014).

In recent months, a slew of methods have been proposed that aim to replicate the above methods for NLP applications. Universal Language Model for Fine-Tuning (ULMFiT) (Howard and Ruder, 2018) and Transformer networks (Vaswani et al., 2017) have been shown to achieve state-of-the-art (SoTA) results on sentiment classification tasks without large labelled datasets. We'll take a closer look at these deep learning methods in the feature-representation section of this survey.

4 Instance-based methods

Instance-weighting frameworks seek to bring either the distribution of the data $p(x)$ or the conditional label distributions $p(y|x)$ closer by giving greater importance to those instances that behave similarly in both domains. This approach has a straight-forward justification, explained below.

Let's assume a loss function for our classifier, $l(x,y,\theta)$. Our parameter estimation is now as follows:

$$\theta^* = argmin_\theta \sum_{(x,y)\in\chi\times\gamma} p_s(x,y)l(x,y,\theta)$$

In the target domain, we wish to find

$$\theta^* = argmin_\theta \sum_{(x,y)\in\chi\times\gamma} p_t(x,y)l(x,y,\theta)$$

But our training instances are sampled from the source domain. Therefore, we re-write the above equation as:

$$\begin{aligned}\theta^* &= argmin_\theta \sum_{(x,y)\in\chi\times\gamma} \frac{p_t(x,y)}{p_s(x,y)} p_s(x,y)l(x,y,\theta) \\ &\simeq argmin_\theta \sum_{i=1}^{N} \frac{p_t(x_i^s,y_i^s)}{p_s(x_i^s,y_i^s)} p_s(x_i^s,y_i^s)l(x_i^s,y_i^s,\theta) \\ &= argmin_\theta \sum_{i=1}^{N} \frac{p_t(x_i^s,y_i^s)}{p_s(x_i^s,y_i^s)} l(x_i^s,y_i^s,\theta)\end{aligned}$$

Thus, we weight each training instance in the source domain by the ratio $\frac{p_t(x_i^s,y_i^s)}{p_s(x_i^s,y_i^s)}$. However, with no labelled instances the target domain, the above ratio is hard to compute. Proposed approaches compute this by employing various heuristics to measure similarity between two data points.

1. (Jiang and Zhai, 2007) analyse the domain adaptation problem by splitting it into two types:

   - **Labelling Adaptation** : This is the case when $p(y|x)$ differs in the source and target domains.
   - **Instance Adaptation** : This is the case when $p(y|x)$ is mostly similar in both domains, but $p(x)$ differs.

   They propose the *instance pruning* approach for labelling adaptation, where one actively removes those instances from the source domain for which $p_s(y|x)$ is different from $p_t(y|x)$, and an *instance weighting* aprroach for the latter case. They also propose a general weighting scheme that combines the above two approaches, and estimate the weights for each instance by a set of heuristics inspired from self-training and semi-supervised learning methods. However, these methods require some labelled data in the target domain. They evaluate their method on PoS tagging, Named Entity Recognition (NER), and spam classification in different domains.

2. The problem of instance adaptation has been studied in machine learning under the term *covariate shift*. Solutions mainly revolve around matching first or second-order statistics of the source and target datasets (density ratio estimation) (Bickel et al., 2009).

3. Specifically for NLP, instance weighting methods were studied for Statistical Machine Translation (SMT). In (Axelrod et al., 2011), the authors train a language model on the target domain data, and rate source domain data based on their perplexity scores on the language model.

4. (Xia et al., 2013) assume that that the training data contains some samples that are drawn from, or are very close to, the target domain distribution. They apply a semi-supervised learning technique called PU learning to identify and weight these instances, and evaluate their approach on cross-domain sentiment classification.

5 Feature-based approaches

These methods attempt to find a feature-space χ that minimizes the different between $p_t(y,x)$ and $p_s(y,x)$. With the advent of deep learning methods in NLP, it has become the most popular approach towards building domain and task invariant representations. We'll begin this section with a look some of the earlier algorithms proposed in this area, and then switch our focus to advances in neural methods and a discussion of the current state-of-the-art.

1. **EasyAdapt**
   (Daume III, 2007) propose a "frustratingly easy" approach to supervised domain adaptation, where each original feature is replicated thrice - as a source-specific, target-specific and domain invariant version. Features that behave similarly across domains will have a large weight for the domian-invariant version, whereas domain-specific features will have a higher value in their specific domains. For example, going back to our part-of-speech tagging task for the WSJ dataset vs product reviews, consider two words: 'the' and 'monitor'. The feature 'the as a determiner' is common across domains, which means it's domain-specific weight is higher. The feature 'monitor as a verb' will have a large weight in the WSJ domain, and 'monitor as a noun' for the target domain. A common classifier is trained on samples from the source and target domain to obtain these weights. One evident disadvantage is that a substantial amount of labelled target data is needed to achieve a reliable estimate of these weights, though a semi-supervised improvement was suggested in a later paper titled EasyAdapt++ (Daumé III et al., 2010).

2. **Structural Correspondence Learning (SCL)**
   Proposed by (Blitzer et al., 2006), SCL introduced the idea of using *pivot features*, a concept that has been adopted by several other domain adaptation methods proposed since. Pivot features are those that behave similarly in both source and target domains, and are also indicative of the behaviour of non-pivot features. Thus, an important feature of the algorithm is that it can model hidden *correlations* between features, rather than just dealing with lexical similarities.

   Let us consider the following set of sentences (Figure 1) from two corpora - the WSJ and the BIO corpus, which contains abstracts of publications in the medical domain.

Figure 1

(b) MEDLINE occurrences of signal, together with pivot features	**(c)** Corresponding WSJ words, together with pivot features
the **signal** *required* to	of **investment** *required*
stimulatory **signal** *from*	of **buyouts** *from* buyers
essential **signal** *for*	to **jail** *for* violating

   The words in italics are pivot features - they have the same part-of-speech tags in both domains. Further, they are indicative of the PoS tags of the non-pivot features, i.e, the words in bold. Thus, we can say that if "required" appears to the right of a word, then that word is likely to be a noun.
   The SCL algorithm proceeds as follows:

   (a) Define m pivot features.

   (b) Build m linear classifiers for each of the pivot features to model the correspondence between them and the non-pivot features. Let the joint weight matrix of these classifiers be W.

   (c) Obtain a projection matrix θ by performing Singular Value Decomposition (SVD) on W.

   (d) Project the original features χ into this new space, $\theta\chi$.

   (e) Train a classifier on the source domain using both the original and the transformed feaures.

   SCL can be used both with and without labelled data in the target domain. The key idea is that if two non-pivot features from different domains are highly correlated with the same

pivot features, then they will be projected to the same space in the latent space. Thus, our classifier trained with the transformed features on the source domain should also be effective in the target domain.

3. **Distributed representations**
   Distributed representations are currently the most pervasive and popular approach to task-invariant learning. Before jumping into neural models, we'll briefly look at some of the early work in building latent representations of words.

   Distributed representations refer to encodings of words as continuous vectors in some n-dimensional latent space. In contrast to symbolic, or one-hot, encodings, they allow us to capture notions of similarity among words - and also make them easier to use with statistical optimization techniques like gradient descent.

   One of the earliest influential works in distributional semantics was Latent Semantic Analysis/Indexing (LSA/LSI) (Deerwester et al., 1990), a count-based method that used SVD to obtain latent representations. This was followed by other topic-model based methods such as Probabilistic LSA (PLSA) (Hofmann, 1999) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003). In the context of domain adaptation, these techniques are used to identify common latent topics among different corpora, and a model is trained using only the common features (Guo et al., 2009). Though the need for labelled target data is eliminated, topic models usually involve hyperparameters that are hard to set without some ground truth knowledge (such as the number of topics).

The surveys in (Pan et al., 2010), (Li, 2012) and (Jiang, 2008) provide an excellent and more detailed guide to the above mentioned methods. In the following section, we'll shift our focus to domain adaptation and transfer learning approaches involving neural architectures.

6 Domain Adaptation with Neural Networks

As mentioned before, transfer learning techniques with neural networks have seen great success in the computer vision domain. Pre-trained representations obtained from deep Convolutional Neural Networks (CNNs) trained for object recogntition on the ImageNet dataset (Deng et al., 2009) have worked extremely well across a wide array of tasks (Huh et al., 2016). The various layers of the CNN model hierarchically capture different facets of an object - the lower layers capture features such as edges and contours, higher level layers capture more complex features like body parts and faces. For a new task, one can simply use these off-the-shelf representations along with a shallow network for classifiation, with fine-tuning if needed.

The success of the above method implies that object recognition, as a task, involves extracting knowledge that is relevant for several other vision tasks as well, such as image captioning. Unfortunately, we are yet to find such a generic proxy objective for language understanding. The closest we have come is perhaps language modelling - contextual representations obtained from networks trained on next word prediction tasks are just beginning to gain traction. Pretrained representations with fine-tuning fall under the supervised domain adaptation category, and the main aim of most of the following methods is to obtain representations that work well across a range of tasks and domains.

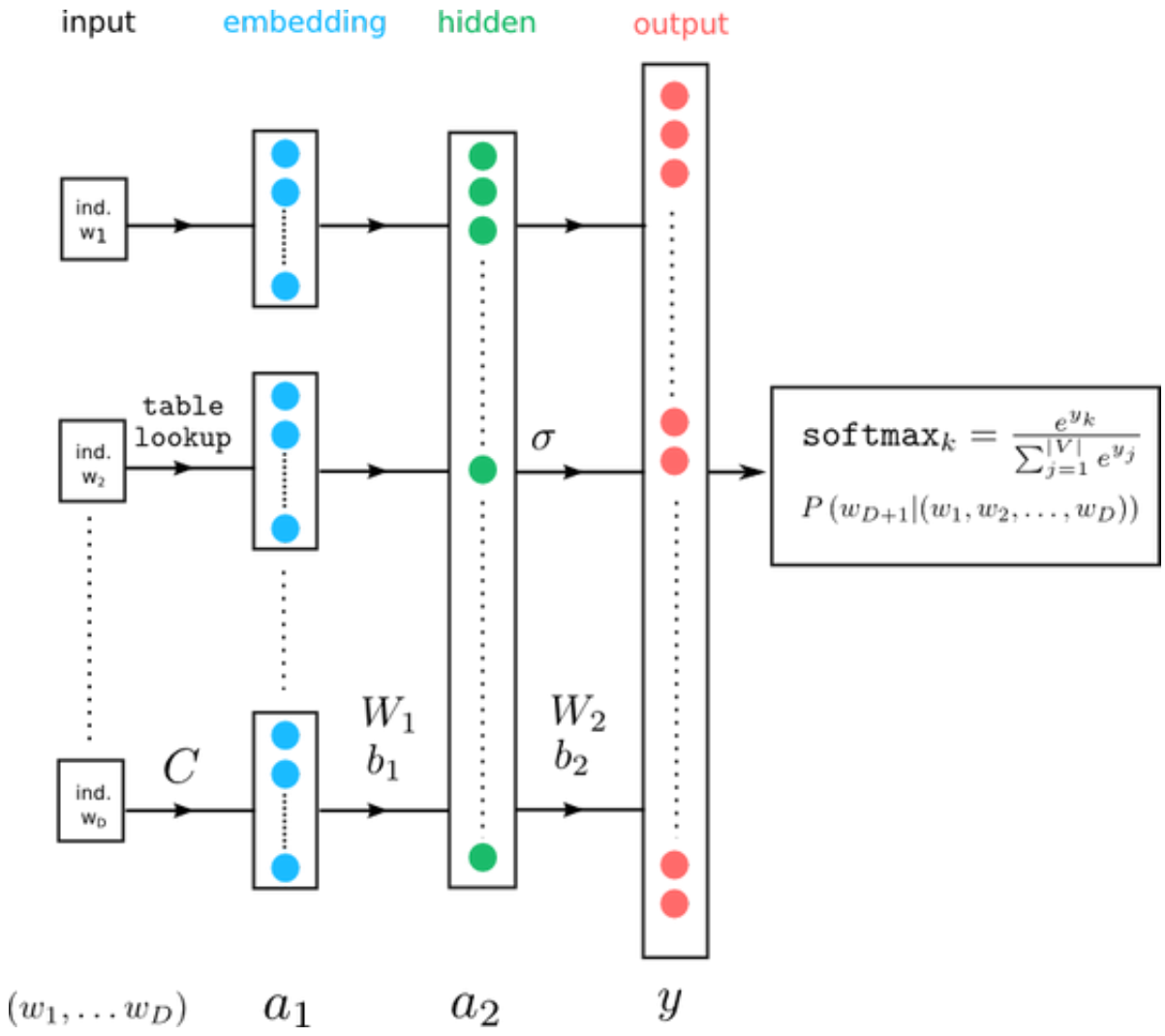
6.1 Pre-trained Representations

Neural representations were the successors to the distributed methods we looked at in section 5. In (Bengio et al., 2003), the authors first coined the term *word embeddings* for continuous representations obtained via language modelling with multi-layered perceptron networks (MLP). They were further popularised by Collobert and Weston in 2008 (Collobert and Weston, 2008). Their work combined word embedding training with several downstream tasks such as part-of-speech tagging and semantic role-labelling using convolutional networks. The general architecture of these models is shown in Figure 2, taken from (Himmetoglu, 2016). The hidden layer in the figure can be a single fully connected layer, an MLP, a deep CNN, a Recurrent Neural Network (RNN), and so on. Word representations are obtained from the blue embedding layer, which is usually randomly initialised and then trained along with the model parameters. The objective function can be any task. (Bengio et al., 2003) used language modelling, while (Collobert and Weston, 2008) trained their network jointly on six tasks and showed that multi-task learning leads to better representations. Their work shared only the embedding layer among different task architectures, but this need not always be the case.

6.2 word2vec and GloVe

The deep neural representations described above proved too computationally intensive to be widely adopted in their time. (Mikolov et al., 2013) and (Pennington et al., 2014) introduced two new methods that considerably reduced the complexity by utilising shallow architectures with no hidden layers. (Mikolov et al., 2013)'s word2vec model is trained using only the words within an n-sized window of a target word, and employs tricks such as negative sampling and sub-sampling to reduce computation time. Pre-trained word2vec embeddings remain the most popular word representations till date, usually employed as the initial embedding layer of a task-specific network.

Figure 2: A Neural Language Model



(Pennington et al., 2014) use global co-occurance information rather than just the local context used by word2vec, and train a regression model that seeks to minimize the difference between the vector dot product of two words and the log of their co-occurence ratio.

Evaluated on both semantic and syntactic tasks, these vectors were shown to perform better than distributional semantic models like LSA.

6.3 ELMo, BERT and Fine-tuning

After word2vec, several improvements to building better word embeddings were proposed by way of character-level, subword-level and word sense embeddings (Santos and Zadrozny, 2014) (Bojanowski et al., 2017) (Trask et al., 2015). There was also work on building phrase-

level, sentence-level and document-level vector representations (Li et al., 2015), however, the transferability of these to different domains and tasks is unclear.

In what has been hailed as "NLPs ImageNet moment" (Ruder, 2018), the last two years have seen a resurgance of pre-trained language model embeddings. With better frameworks and architectures such as Transformers and attention mechanisms, they are on the verge of replacing word2vec as the go-to choice for word representations. Works such as ULMFiT, Embeddings from Language Models (ELMo) (Peters et al., 2018) and most recently, Bideirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) show that language model pre-training on deep neural networks followed by fine-tuning of the top few layers gives state-of-the-art results on a wide range of NLP tasks like text classification, sequence tagging and natural language inference with only a few hundred samples - a huge improvement over few thousand per-class samples usually needed.

6.4 Domain-specific embeddings

All of the above work focussed on supervised domain adaptation with general-purpose word representations. (Bollegala et al., 2015) introduce the cross-domain word representation task, where the goal is to learn a domain-specific representation for each common word w. Inspired by the concept of pivot and non-pivot features introduced in (Blitzer et al., 2006), they constrain the representations of pivot features to be similar across domains. Their objective function closely follows that of word2vec, but is limited to predicting non-pivot features from the surrounding pivot fetures. Evaluated on cross-domain sentiment classification, they showed that their method outperformed SCL and other baselines.

(Yang et al., 2017) alternatively add a regularisation term to the word2vec objective, that measures the relevance of a word to each domain via a frequency-based similarity measure. It loosely follows the same inuition as (Bollegala et al., 2015), wherein they try to keep the vectors of words that are frequent in both domains as close to each other as possible.

Both of the above methods train a logistic regression classifier on the source domain dataset with the learned embeddings, and evaluate on the target domain. Thus, their only target domain requirements are a relatively large unlabelled dataset.

6.5 Autoencoder based domain adaptation

Autoencoders are another key representation learning approach, apart from the methods mentioned in the above sections. Autoencoders are networks that are trained to reconstruct the input representation. Mathematically, they comprise two functions: and enocder function $h(.)$ and

a decoder function $g(.)$. The reconstruction is given by $r(x) = g(h(x))$, and the reconstruction error by some loss function $l(x, r(x))$. (Vincent et al., 2008) proposed the stacked denoising autoencoder (SDA), where several autoencoders were sequentially stacked on top of one another, with the encoded output of the first serving as the input for the second and so on. The parameters of each level serve as hierarchical representations of the input, much like other deep neural networks. A denoising autoencoder, as opposed to the regular autoencoder, is fed a slightly corrupted input representation $\tilde{x}$ instead of x.

1. (Glorot et al., 2011) propose an SDA-based domain adaptation method for sentiment analysis. The SDA is trained on the unlabelled data from all domains to build effective feature representations, and an SVM classifier is then learnt on the source domain data.

2. (Chen et al., 2012) improved the computational complexity of the above approach by proposing the marginalised SDA (mSDA) model, which allows only linear transformations in the denoisers. Their method achieved performance comparable to that of (Glorot et al., 2011), while massively improving on the training time.

6.6 Domain-adversarial training

Another representation learning approach that achieved state-of-the-art sentiment classification results is domain-adversarial training (Ganin et al., 2016). Like most of the above methods, it is based on the idea of learning *domain-invariant representations*, rather than domain-specific ones. The authors introduce an additional loss term based on the model's prediction of which domain the input sample comes from. The model now tries to maximise this loss term, which effectively means that the representations of samples from different domains become indistinguishable.

7 Related Work

In this section, we'll look at some concepts that are closely related to those of transfer learning, but either have enough work done in them to merit a separate field name, or do not fall under any specific methodological category.

7.1 Looking at data

Looking back at Section 4, we see that instance weighting is a supervised domain adaptation algorithm. In (Plank et al., 2014), the authors apply the technique to unsupervised cross-domain part-of-speech tagging and present a *negative* result. After experimenting with different

weighting schemes across six corpora, they conclude that most of the performance drop across domains is due to out-of-vocabulary words - an issue that importance weighting doesn't help much with.

In (Plank, 2016), Prof. Barbara Plank takes a step back to ask what actually constitutes a domain difference, and discusses how models need to be able to detect this change and adapt to it without any prior knowledge of the target domain. For example, even within a single dataset such as social media tweets, one observes great variety in the language style relating to different social groups, topics, user demographics, etc. She introduces the term *fortuitous data* for data that can be harvested from side-sources, such as Wiktionary, to help with unsupervised domain adaptation. Further work by her and others focusses on optimally selecting data for transfer learning purposes. (Ruder and Plank, 2017) uses Bayesian Optimisation to learn a measure of similarity between domains, and then find the most promising examples from the source domain for transfer. Looking at data and understanding when and why different transfer learning methods work (or do not work) is an important research direction, and one that can help us avoid *negative transfer*. Negative transfer is when a transfer method actually decreases performance when compared to traditional supervised leaerning (Torrey and Shavlik, 2010).

7.2 Multi-task learning

Multi-task learning (Caruna, 1993) seeks to optimise the objectives for several tasks simultaneously, as opposed to transfer learning that is concerned only with the target domain/task. Though the assumption is implicit that one has labelled data for every task, it's been observed that multi-task learning can push up the performance on a task with very little labelled data by leveraging information learnt from the other tasks. With neural networks, parameter sharing has emerged as a widely used multi-task learning technique where all network layers except the last classification one are shared among all tasks. We looked at one instance of this in NLP already, with the work in (Collobert and Weston, 2008) that used it to learn word representations. Recent work has leveraged it for cross-domain classification (Yu and Jiang, 2016) (Liu et al., 2015), representation learning (Hashimoto et al., 2016) and sequence tagging (Søgaard and Goldberg, 2016) (Rei, 2017).

7.3 Few-shot learning

Few-shot learning, and other techniques like one-shot and zero-shot learning, refer to learning classifiers with very few (or zero, or one) example per class. It is motivated by the ability of humans to generalise and quickly learn new concepts with very little supervision; for example,

learning to identify different animals. (Vinyals et al., 2016) explored this idea by focussing on two goals - rapid acquisition of new examples while providing excellent generalisation from common examples. Using neural networks that are augmented with memory space, they achieve an accuracy of around 40% on a language modelling related task (predicting the missing word in a sentence). While not impressive compared to state-of-the-art methods, it is considerably above the random baseline of 20%. Few-shot learning has been mostly succesful in cross-lingual learning in NLP (Artetxe and Schwenk, 2018) (Upadhyay et al., 2018).

7.4 Cross-lingual learning

Learning cross-lingual representations with minimal data has seen impressive improvements in recent times. Moving from bilingual to multilingual analysis, recent work has aligned word embeddings of more than 80 languages in a single latent space (Conneau et al., 2017). This allows us to translate between any two language pairs, even those with no parallel data between them, i.e., zero-shot translation. Cross-lingual embeddings also allow us to exploit similarities between languages that share typological properties, such as morphology and sentence structure (Johnson et al., 2017).

7.5 Semi-supervised learning

If one ignores the domain difference between source and target datasets, the domain adaptation problem is very similar to that of semi-supervised learning. Semi-supervised learning methods typically assume limited availability of labelled data, and aim to make use of unlabelled data during training. Classic semi-supervised learning technqiues include self-training (Yarowsky, 1995), co-training (Blum and Mitchell, 1998) and boosting (Mallapragada et al., 2009).

(Ruder and Plank, 2018) considers how semi-supervised algorithms can be adapted to learn under domain shift. Their evaluation on sentiment classification and PoS tagging showed that these techniques achieved performance comparable to many neural and state-of-the-art transfer learning methods.

8 Case Study: Part-of-Speech tagging

In this section, we look at one specific NLP application, part-of-speech tagging, and how the different domain adaptation methods we've talked above in previous sections fare on the task.

8.1 Why domain adaptation?

We briefly spoke about why domain matters in PoS tagging in the introductory section of this paper, with the word 'monitor' being used as a verb and a noun depending on the domain. Out-of-vocabulary (OOV) words provide another challenge, especially with new proper nouns being introduced in each domain. One can also imagine that domains like Twitter have very different sentence structures when compared to news articles.

8.2 Datasets

The standard dataset in English for tagging and parsing experiments is the Wall Street Journal (WSJ) portion of the Penn Treebank corpus (Marcus et al., 1993). The Syntactic Analysis of Non-Canonical Language (SANCL) 2012 shared task (Petrov and McDonald, 2012) provided five more domains - newsgroups, weblogs, reviews, answers and emails.

8.3 Methods

We present numbers, where available, for the above datasets and the following methods:

1. SCL - Structural Correspondence Learning from (Blitzer et al., 2006)
2. Stanford Tagger - A Maximum Entropy Markov Model used by the Stanford PoS tagger, (Toutanova and Manning, 2000).
3. mSDA - The marginalised Stacked Denoisig Autoencoder from (Chen et al., 2012).
4. FLORS - A representation learning based model proposed specifically for PoS tagging that uses handcrafted features like word suffixes, (Schnabel and Schütze, 2014).
5. FEMA - Feature EMbeddings for domain Adaptation, a representation learning model proposed specifically for PoS tagging that used feature embeddings along with domain attribute embeddings, (Yang and Eisenstein, 2015).
6. Bi-LSTM - A bidirectional Long Short Term Memory network (LSTM) based tagger from (Plank et al., 2016).

8.4 Results

Table 1 below lists the F1 scores obtained by each of the above methods. The numbers on the source domain test set, i.e, WSJ, average around 97% for all the methods, and is the accepted

SoTA. However, performance on the other domains is at least 5-6% percentage points lower. This difference is also dependent on the target domain itself - the EMAILS domain consistently proves to be the hardest to adapt to. This further drives home the importance of looking at data and being able to adapt dynamically to the domain.

Table 1: F_1 scores of DA algorithms for PoS Tagging. Best results in each domain are highlighted in bold.

	WSJ	ANSWERS	NEWSGROUPS	WEBLOGS	EMAILS	REVIEWS
SCL	-	90.04	91.51	92.32	88.04	90.29
Stanford Tagger	97.43	89.74	91.25	92.32	87.77	90.30
Bi-LSTM	**97.50**	90.43	91.83	92.44	87.95	90.04
mSDA	-	90.61	91.83	92.39	88.11	90.95
FLORS	97.11	91.17	92.41	93.14	88.67	**92.25**
FEMA	-	**91.35**	**92.60**	**93.43**	**89.02**	92.15

8.5 Discussion

One problem while evaluating performance of an algorithm across domains is inconsistencies in tag annotation. In (Schnabel and Schütze, 2014), the authors draw attention to some examples of this: file names, such as "xyz.doc" are annotated as NN, whereas their behaviour is closer to that of NNPs. For the BIO dataset introduced in (Blitzer et al., 2006), many bio-specific names are again annotated as NNs rather than NNPs. They show that converting all NNP tags to NN improved the F_1 scores of algorithms by almost 4 percentage points.

Further anaysis by them and in (Ruder and Plank, 2018) examines how algorithms fare on OOV words, unknown tags and unknown word-tag combinations. FLORS, which uses contextual information, performs much better on these categories than other methods that use lexical information, as is to be expected. An important point mentioned is that there are some things DA simply cannot do, such as predicting tags that don't occur anywhere in the training data.

Finally, the top two systems from Table 1 are both designed specifically for the task at hand. General purpose embeddings like word2vec are far from reaching SoTA on unsupervised domain adaptation. This is also the case for other NLP applications like named entity recognition and sentiment analysis, and indicates that we are far from finding optimal representations that are pertinent to multiple tasks simultaneously.

9 Conclusion

In this survey, we have attempted to provide a brief overview of domain adaptation methods for natural language processing. We have covered the broad categories under which DA algorithms fall and their mathematical underpinning, however, it is by no means comprehensive with regards to papers that have been published under the topic.

Many domain adaptation methods have been formulated specifically for certain applications. Sentiment analysis in particular has seen a lot of work in the area. Recently, there has been an increased focus on analysing social media data, especially that of Twitter, with algorithms desgined to deal with its format of short text, hashtags and emoticons. For sequence labelling tasks like Named Entity Recognition (NER) and PoS tagging, heterogenous label sets and OOV words present a significant challenge.

With recent efforts concentrating on building domain and task invariant word representations, the line between domain adaptation and transfer learning has blurred. Deep neural models like ELMo can dynamically generate embeddings for a word based on the context it appears in, replacing word2vec as the de-facto choice for pre-trained word vectors. Large scale language-model pre-training followed by fine tuning has drastically reduced the labelled data requirements for classification tasks, though the number is still in the few thousands. Transfer learning is currently the field that is seeing the most interest and development in NLP.

However, the performance numbers in Section 8.4 demonstrate that there is still a lot of progress to be made. Apart from building general purpose representations, one needs to explicitly identify and use target-domain specific data where necessary. Semi-supervised learning methods are seeing a resurgence in this regard, e.g., (Clark et al., 2018). Cross-lingual learning is another relevant field that is receiving attention, and in particular, improving translation performance on low-resource languages.

Shared tasks such as CLEF (`http://clef2018.clef-initiative.eu/index.php?`) and SemEval (`http://alt.qcri.org/semeval2019/index.php?id=tasks`) have been crucial in providing datasets and challenges encouraging development in these areas, and are also a good starting point to survey previous work. Other resources that I have found useful include NLP blogs (`http://ruder.io/`), libraries (`http://nlp.fast.ai/`) and last but not least, #NLProc Twitter threads (`https://twitter.com/arxiv_cscl`).

Bibliography

M. Artetxe and H. Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *arXiv preprint arXiv:1812.10464*, 2018.

A. Axelrod, X. He, and J. Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, pages 355–362. Association for Computational Linguistics, 2011.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, 2000.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(Sep):2137–2155, 2009.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics, 2006.

J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

D. Bollegala, T. Maehara, and K.-i. Kawarabayashi. Unsupervised cross-domain word representation learning. *arXiv preprint arXiv:1505.07184*, 2015.

R. Caruna. Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 41–48, 1993.

Y. S. Chan and H. T. Ng. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 89–96. Association for Computational Linguistics, 2006.

S. Changpinyo, H. Hu, and F. Sha. Multi-task learning for sequence tagging: An empirical study. *arXiv preprint arXiv:1808.04151*, 2018.

C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4):382–399, 2006.

M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.

C. Chu, R. Dabre, and S. Kurohashi. An empirical comparison of simple domain adaptation methods for neural machine translation. *arXiv preprint arXiv:1701.03214*, 2017.

K. Clark, M.-T. Luong, C. D. Manning, and Q. V. Le. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*, 2018.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

H. Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, 2007.

H. Daumé III, A. Kumar, and A. Saha. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59. Association for Computational Linguistics, 2010.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6): 391–407, 1990.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

J. R. Finkel and C. D. Manning. Hierarchical bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610. Association for Computational Linguistics, 2009.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

Y. Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017.

H. Guo, H. Zhu, Z. Guo, X. Zhang, X. Wu, and Z. Su. Domain adaptation with latent semantic association for named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 281–289. Association for Computational Linguistics, 2009.

K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

B. Himmetoglu. Deciphering the neural language model. `https://burakhimmetoglu.com/2016/12/16/deciphering-the-neural-language-model/`, 2016.

T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339, 2018.

M. Huh, P. Agrawal, and A. A. Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

J. Jiang. A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 3:1–12, 2008.

J. Jiang and C. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 264–271, 2007.

M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5: 339–351, 2017.

J. Y. Lee, F. Dernoncourt, and P. Szolovits. Transfer learning for named-entity recognition with neural networks. *arXiv preprint arXiv:1705.06273*, 2017.

J. Li, M.-T. Luong, and D. Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

Q. Li. Literature survey: domain adaptation algorithms for natural language processing. *Department of Computer Science The Graduate Center, The City University of New York*, pages 8–10, 2012.

X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. 2015.

P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 31(11):2000–2014, 2009.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111*, 2016.

S. J. Pan, Q. Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

S. Petrov and R. McDonald. Overview of the 2012 shared task on parsing the web. In *Notes of the first workshop on syntactic analysis of non-canonical language (sancl)*, volume 59. Citeseer, 2012.

D. Pinto, J. Civera, A. Barrón-Cedeno, A. Juan, and P. Rosso. A statistical approach to crosslingual natural language tasks. *Journal of Algorithms*, 64(1):51–60, 2009.

B. Plank. What to do about non-standard (or non-canonical) language in nlp. *arXiv preprint arXiv:1608.07836*, 2016.

B. Plank, A. Johannsen, and A. Søgaard. Importance weighting and unsupervised domain adaptation of pos taggers: a negative result. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 968–973, 2014.

B. Plank, A. Søgaard, and Y. Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *arXiv preprint arXiv:1604.05529*, 2016.

M. Rei. Semi-supervised multitask learning for sequence labeling. *arXiv preprint arXiv:1704.07156*, 2017.

S. Ruder. Nlp's imagenet moment has arrived. `https://thegradient.pub/nlp-imagenet/`, 2018.

S. Ruder and B. Plank. Learning to select data for transfer learning with bayesian optimization. *arXiv preprint arXiv:1707.05246*, 2017.

S. Ruder and B. Plank. Strong baselines for neural semi-supervised learning under domain shift. *arXiv preprint arXiv:1804.09530*, 2018.

C. D. Santos and B. Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826, 2014.

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

A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

A. Søgaard and Y. Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235, 2016.

S. Srivastava, I. Labutov, and T. Mitchell. Zero-shot learning of classifiers from natural language quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 306–316, 2018.

L. Torrey and J. Shavlik. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 242–264. IGI Global, 2010.

K. Toutanova and C. D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000.

A. Trask, P. Michalak, and J. Liu. sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*, 2015.

S. Upadhyay, M. Faruqui, G. Tür, H.-T. Dilek, and L. Heck. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE, 2018.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

R. Xia, X. Hu, J. Lu, J. Yang, C. Zong, et al. Instance selection and instance weighting for cross-domain sentiment classification via pu learning. In *IJCAI*, pages 2176–2182, 2013.

R. Xu, J. Xu, and X. Wang. Instance level transfer learning for cross lingual opinion analysis. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 182–188. Association for Computational Linguistics, 2011.

J. Yang, R. Yan, and A. G. Hauptmann. Adapting svm classifiers to data with shifted distributions. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 69–76. IEEE, 2007.

W. Yang, W. Lu, and V. Zheng. A simple regularization-based algorithm for learning cross-domain word embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2898–2904, 2017.

Y. Yang and J. Eisenstein. Unsupervised multi-domain adaptation with feature embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 672–682, 2015.

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

J. Yu and J. Jiang. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 236–246, 2016.

Y. Ziser and R. Reichart. 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)*, volume 1, pages 1241–1251, 2018.