Recommendation Chart of Domains for Cross-Domain Sentiment Analysis: Findings of A 20 Domain Study

Akash Sheoran$^\diamond$, Diptesh Kanojia$^{\dagger,\clubsuit,\star}$, Aditya Joshi$^\heartsuit$, and Pushpak Bhattacharyya†

$^\diamond$Samsung Research Institute Bangalore, India
†Indian Institute of Technology Bombay, India
$^\clubsuit$IITB-Monash Research Academy, India
*Monash University, Australia
$^\heartsuit$CSIRO Data61, Australia
$^\diamond$a.sheoran@samsung.com, †\{diptesh, pb\}@cse.iitb.ac.in, $^\heartsuit$aditya.joshi@data61.csiro.au

Abstract

Cross-domain sentiment analysis (CDSA) helps to address the problem of data scarcity in scenarios where labelled data for a domain (known as the target domain) is unavailable or insufficient. However, the decision to choose a domain (known as the source domain) to leverage from is, at best, intuitive. In this paper, we investigate text similarity metrics to facilitate source domain selection for CDSA. We report results on 20 domains (all possible pairs) using 11 similarity metrics. Specifically, we compare CDSA performance with these metrics for different domain-pairs to enable the selection of a suitable source domain, given a target domain. These metrics include two novel metrics for evaluating domain adaptability to help source domain selection of labelled data and utilize word and sentence-based embeddings as metrics for unlabelled data. The goal of our experiments is a recommendation chart that gives the K best source domains for CDSA for a given target domain. We show that the best K source domains returned by our similarity metrics have a precision of over 50%, for varying values of K.

Keywords: sentiment analysis evaluation, cross-domain sentiment analysis, similarity metrics

1. Introduction

Sentiment analysis (SA) deals with automatic detection of opinion orientation in text (Liu and others, 2010). Domain-specificity of sentiment words, and, as a result, sentiment analysis is also a well-known challenge. A popular example being '*unpredictable*' that is positive for a book review (as in '*The plot of the book is unpredictable*') but negative for an automobile review (as in '*The steering of the car is unpredictable*'). Therefore, a classifier that has been trained on book reviews may not perform as well for automobile reviews (Pang et al., 2008).

However, sufficient datasets may not be available for a domain in which an SA system is to be trained. This has resulted in research in cross-domain sentiment analysis (CDSA). CDSA refers to approaches where the training data is from a different domain (referred to as the '*source domain*') as compared to that of the test data (referred to as the '*target domain*'). Ben-David et al. (2007) show that similarity between the source and target domains can be used as indicators for domain adaptation, in general.

In this paper, we validate the idea for CDSA. We use similarity metrics as a basis for source domain selection. We implement an LSTM-based sentiment classifier and evaluate its performance for CDSA for a dataset of reviews from twenty domains. We then compare it with similarity metrics to understand which metrics are useful. The resultant deliverable is a recommendation chart of source domains for cross-domain sentiment analysis.

The key contributions of this work are:

1. We compare eleven similarity metrics (four that use labelled data for the target domain, seven that do not use labelled data for the target domain) with the CDSA performance of 20 domains. Out of these eleven metrics, we introduce two new metrics.

2. Based on CDSA results, we create a recommendation chart that prescribes domains that are the best as the source or target domain, for each of the domains.

3. In general, we show which similarity metrics are crucial indicators of the benefit to a target domain, in terms of source domain selection for CDSA.

With rising business applications of sentiment analysis, the convenience of cross-domain adaptation of sentiment classifiers is an attractive proposition. We hope that our recommendation chart will be a useful resource for the rapid development of sentiment classifiers for a domain of which a dataset may not be available. Our approach is based on the hypothesis that if source and target domains are similar, their CDSA accuracy should also be higher given all other conditions (such as data size) are the same. The rest of the paper is organized as follows. We describe related work in Section 2. We then introduce our sentiment classifier in Section 3. and the similarity metrics in Section 4. The results are presented in Section 5. followed by a discussion in Section 6. Finally, we conclude the paper in Section 7.

2. Related Work

Cross-domain adaptation has been reported for several NLP tasks such as part-of-speech tagging (Blitzer et al., 2006), dependency parsing (Zhang and Wang, 2009), and named entity recognition (Daume III, 2007). Early work in CDSA is by Denecke (2009). They show that lexicons such as SentiWordnet do not perform consistently for sentiment classification of multiple domains. Typical statistical approaches for CDSA use active learning (Li et al., 2013),

co-training (Chen et al., 2011) or spectral feature alignment (Pan et al., 2010). In terms of the use of topic models for CDSA, He et al. (2011) adapt the joint sentiment tying model by introducing domain-specific sentiment-word priors. Similarly, cross-domain sentiment and topic lexicons have been extracted using automatic methods (Li et al., 2012). Glorot et al. (2011) present a method for domain adaptation of sentiment classification that uses deep architectures. Our work differs from theirs in terms of computational intensity (deep architecture) and scale (4 domains only).

In this paper, we compare similarity metrics with cross-domain adaptation for the task of sentiment analysis. This has been performed for several other tasks. Recent work by Dai et al. (2019) uses similarity metrics to select the domain from which pre-trained embeddings should be obtained for named entity recognition. Similarly, Schultz et al. (2018) present a method for source domain selection as a weighted sum of similarity metrics. They use statistical classifiers such as logistic regression and support vector machines. However, the similarity measures used are computationally intensive. To the best of our knowledge, this is the first work at this scale that compares different *cost-effective* similarity metrics with the performance of CDSA.

3. Sentiment Classifier

The core of this work is a sentiment classifier for different domains. We use the DRANZIERA benchmark dataset (Dragoni et al., 2016), which consists of Amazon reviews from 20 domains such as automatives, baby products, beauty products, *etc.* The detailed list can be seen in Table 1. To ensure that the datasets are balanced across all domains, we randomly select 5000 positive and 5000 negative reviews from each domain. The length of the reviews ranges from 5 words to 1654 words across all domains, with an average length ranging from 71 words to 125 words per domain. We point the reader to the original paper for detailed dataset statistics.

We normalize the dataset by removing numerical values, punctuations, stop words, and changing all words to the lower case. To train the sentiment classifier, we use an LSTM-based sentiment classifier. It consists of an embedding layer initialized with pre-trained GloVe word embeddings of 100 dimensions. We specify a hidden layer with 128 units and maintain the batch size at 300[1]. We train this model for 20 epochs with a dropout factor of 0.2 and use *sigmoid* as the activation function. For *In-domain sentiment analysis*, we report a 5-fold classification accuracy with a train-test split of 8000 and 2000 reviews. In *cross-domain set up*, we report an average accuracy over 5 splits of 2000 reviews in the target domain in Table 1.

4. Similarity Metrics

In table 2, we present the n-gram percent match among the domain data used in our experiments. We observe that the

n-gram match from among this corpora is relatively low and simple corpus similarity measures which use orthographic techniques cannot be used to obtain domain similarity. Hence, we propose the use of the metrics detailed below to perform our experiments.

We use a total of 11 metrics over two scenarios: the first that uses labelled data, while the second that uses unlabelled data.

1. **Labelled Data:** Here, each review in the target domain data is labelled either positive or negative, and a number of such labelled reviews are insufficient in size for training an efficient model.

2. **Unlabelled Data:** Here, positive and negative labels are absent from the target domain data, and the number of such reviews may or may not be sufficient in number.

We explain all our metrics in detail later in this section. These 11 metrics can also be classified into two categories:

- **Symmetric Metrics** - The metrics which consider domain-pairs (D_1, D_2) and (D_2, D_1) as the same and provide similar results for them *viz.* Significant Words Overlap, Chameleon Words Similarity, Symmetric KL Divergence, Word2Vec embeddings, GloVe embeddings, FastText word embeddings, ELMo based embeddings and Universal Sentence Encoder based embeddings.

- **Asymmetric Metrics** - The metrics which are 2-way in nature *i.e.,* (D_1, D_2) and (D_2, D_1) have different similarity values *viz.* Entropy Change, Doc2Vec embeddings, and FastText sentence embeddings. These metrics offer additional advantage as they can help decide which domain to train from and which domain to test on amongst D_1 and D_2.

4.1. Metrics: Labelled Data

Training models for prediction of sentiment can cost one both valuable time and resources. The availability of pre-trained models is *cost-effective* in terms of both time and resources. One can always train new models and test for each source domain since labels are present for the source domain data. However, it is feasible only when trained classification models are available for all source domains. If pre-trained models are unavailable, training for each source domain can be highly intensive both in terms of time and resources. This makes it important to devise *easy-to-compute* metrics that use labelled data in the source and target domains.

When target domain data is labelled, we use the following four metrics for comparing and ranking source domains for a particular target domain:

LM1: Significant Words Overlap

All words in a domain are not significant for sentiment expression. For example, *comfortable* is significant in the 'Clothing' domain but not as significant in the 'Movie' domain. In this metric, we build upon existing work by Sharma et al. (2018) and extract significant words from

---

[1]A smaller batch size results in better accuracy, but at the cost of time. Since the purpose is only to compare the domains and we deal with 400 domain-pairs, we use a larger batch size to save time.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
D1	84.84	70.15	72.58	73.94	**76.92**	74.61	70.34	71.06	77.78	**82.83**	**76.94**	72.93	70.64	69.74	73.56	73.03	74.41	70.19	77.85	76.81
D2	76.34	83.24	77.71	77.83	71.28	**81.53**	80.15	76.61	81.63	75.62	68.03	80.85	79.63	76.45	82.69	77.64	81.49	79.86	80.58	74.62
D3	74.47	75.00	85.78	**80.16**	68.12	77.38	76.06	74.47	80.83	73.03	67.58	78.21	74.35	74.24	75.76	75.59	78.44	74.79	78.97	72.93
D4	75.66	74.32	**80.31**	84.49	68.01	79.01	75.66	**78.60**	82.08	75.87	73.14	77.64	77.71	77.08	82.18	75.02	80.73	77.38	80.54	75.05
D5	81.14	69.81	70.86	73.09	81.56	73.93	65.89	69.26	73.42	79.91	74.97	72.43	71.93	71.67	78.53	74.99	73.38	79.93	77.22	75.20
D6	74.19	73.87	71.99	76.37	67.03	93.67	73.14	74.80	78.95	75.39	67.55	75.15	75.58	69.90	**83.76**	70.99	**82.53**	73.88	77.89	70.87
D7	75.57	77.93	75.67	75.08	67.83	80.60	85.16	73.51	83.47	74.93	72.62	80.65	77.70	71.34	79.64	77.07	79.55	78.31	79.47	72.81
D8	73.83	73.29	78.39	79.82	68.64	78.40	72.10	83.74	81.03	72.80	68.99	77.73	77.83	**78.52**	83.08	74.57	77.54	77.54	80.74	74.55
D9	75.39	72.90	78.70	76.93	68.48	78.65	79.40	76.42	87.01	74.86	73.07	79.22	78.29	73.17	81.13	76.75	79.94	77.09	81.80	75.46
D10	**82.75**	72.69	73.83	73.59	75.46	76.39	71.69	69.29	76.86	84.50	76.59	71.96	71.46	70.18	77.63	73.14	74.45	72.74	77.43	75.78
D11	77.11	65.46	66.81	72.53	72.59	71.69	64.73	69.97	76.28	77.77	84.98	69.19	68.21	65.39	79.05	68.69	71.89	66.95	78.45	76.76
D12	78.31	79.11	78.49	78.69	69.70	80.15	80.10	74.63	**83.58**	76.43	72.59	84.79	79.92	76.94	81.45	**78.52**	81.04	80.62	80.73	77.5
D13	76.00	79.46	77.00	78.42	70.22	81.12	79.35	75.07	82.86	75.98	70.71	81.23	83.74	76.04	78.92	78.11	81.33	**81.89**	81.19	75.28
D14	74.28	77.31	80.29	78.66	68.89	78.32	75.50	77.77	80.99	72.3	69.51	79.19	79.26	84.81	78.85	75.77	78.28	77.91	79.68	73.09
D15	71.91	72.34	71.26	75.29	65.59	77.28	73.09	69.79	77.22	72.34	68.52	73.94	70.79	67.61	95.48	69.78	77.51	71.08	78.00	70.31
D16	72.15	75.18	76.59	75.44	71.09	75.96	77.94	72.14	79.58	72.47	68.65	79.89	77.45	72.78	75.79	84.49	76.62	76.09	77.79	77.07
D17	77.14	77.22	77.27	77.06	69.69	84.02	76.59	75.45	81.09	75.66	72.78	78.66	78.68	71.81	81.93	76.02	88.18	77.87	81.19	73.93
D18	77.15	**80.04**	76.21	79.09	72.11	80.09	**80.92**	75.52	82.29	75.59	68.54	**81.92**	**82.95**	76.06	81.38	78.66	81.71	82.18	81.14	75.61
D19	78.83	71.26	75.33	76.18	71.94	77.90	73.14	71.98	82.17	77.20	74.09	75.36	75.99	73.53	82.69	74.35	76.76	75.29	86.77	**78.85**
D20	79.08	70.15	71.98	73.72	72.27	77.21	66.81	70.14	79.33	78.92	74.29	69.83	74.32	70.09	80.37	69.79	74.83	70.38	**81.19**	83.19

Table 1: Accuracy percentage for all train-test pairs. Domains on rows are source domains and columns are target domains. Domain labels are D1: Amazon Instant Video, D2: Automotive, D3: Baby, D4: Beauty, D5: Books, D6: Clothing Accessories, D7: Electronics, D8: Health, D9: Home, D10: Kitchen, D11: Movies TV, D12: Music, D13: Office Products, D14: Patio, D15: Pet Supplies, D15: Shoes, D16: Software, D17: Sports Outdoors, D18: Tools Home Improvement, D19: Toys Games, D20: Video Games.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
D1	-	0.27	0.27	0.40	0.23	0.50	0.37	0.20	0.57	0.20	0.33	0.20	0.30	0.30	0.20	0.07	0.23	0.20	0.20	0.87
D2	-	-	0.53	0.37	0.27	0.23	0.43	0.30	0.30	0.30	0.40	0.33	0.43	0.40	0.40	0.20	0.33	0.50	0.43	0.27
D3	-	-	-	0.30	0.30	0.27	0.63	0.33	0.30	0.33	0.67	0.23	0.50	0.50	0.63	0.30	0.47	0.53	0.53	0.27
D4	-	-	-	-	0.27	0.37	0.30	0.17	0.37	0.20	0.30	0.17	0.30	0.33	0.23	0.07	0.27	0.27	0.27	0.40
D5	-	-	-	-	-	0.20	0.33	0.23	0.23	0.23	0.37	0.17	0.37	0.43	0.37	0.13	0.43	0.30	0.40	0.23
D6	-	-	-	-	-	-	0.33	0.17	0.37	0.20	0.30	0.17	0.27	0.27	0.20	0.07	0.23	0.20	0.20	0.47
D7	-	-	-	-	-	-	-	0.33	0.37	0.37	0.73	0.33	0.73	0.57	0.63	0.30	0.53	0.50	0.60	0.33
D8	-	-	-	-	-	-	-	-	0.27	0.30	0.37	0.30	0.33	0.47	0.33	0.20	0.27	0.30	0.33	0.20
D9	-	-	-	-	-	-	-	-	-	0.20	0.37	0.17	0.33	0.30	0.23	0.13	0.23	0.27	0.27	0.60
D10	-	-	-	-	-	-	-	-	-	-	0.37	0.43	0.40	0.33	0.33	0.17	0.37	0.37	0.47	0.20
D11	-	-	-	-	-	-	-	-	-	-	-	0.30	0.73	0.57	0.60	0.30	0.50	0.50	0.53	0.33
D12	-	-	-	-	-	-	-	-	-	-	-	-	0.37	0.27	0.27	0.10	0.30	0.30	0.40	0.20
D13	-	-	-	-	-	-	-	-	-	-	-	-	-	0.57	0.50	0.30	0.47	0.53	0.53	0.30
D14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.57	0.23	0.43	0.43	0.53	0.30
D15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.30	0.47	0.43	0.60	0.23
D16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.27	0.23	0.27	0.07
D17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.40	0.53	0.27
D18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.50	0.23
D19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.23

Table 2: N-grams co-occurrence matrix depicting the percent point match among the top-10 bigrams, trigrams and quadgrams for the data used in each domain. Domain labels are D1: Amazon Instant Video, D2: Automotive, D3: Baby, D4: Beauty, D5: Books, D6: Clothing Accessories, D7: Electronics, D8: Health, D9: Home, D10: Kitchen, D11: Movies TV, D12: Music, D13: Office Products, D14: Patio, D15: Pet Supplies, D15: Shoes, D16: Software, D17: Sports Outdoors, D18: Tools Home Improvement, D19: Toys Games, D20: Video Games.

each domain using the χ^2 test. This method relies on computing the statistical significance of a word based on the polarity of that word in the domain. For our experiments, we consider only the words which appear at least 10 times in the corpus and have a χ^2 value greater than or equal to 1. The χ^2 value is calculated as follows:

$$\chi^2(w) = \frac{(c_p{}^w - \mu^w)^2 + (c_n{}^w - \mu^w)^2}{\mu^w} \qquad (1)$$

Where $c_p{}^w$ and $c_n{}^w$ are the observed counts of word w in positive and negative reviews, respectively. μ^w is the expected count, which is kept as half of the total number of occurrences of w in the corpus. We hypothesize that, if a domain-pair (D_1, D_2) shares a larger number of significant words than the pair (D_1, D_3), then D_1 is closer to D_2 as

compared to D_3, since they use relatively higher number of similar words for sentiment expression. For every target domain, we compute the intersection of significant words with all other domains and rank them on the basis of intersection count. The utility of this metric is that it can also be used in a scenario where target domain data is unlabelled, but source domain data is labelled. It is due to the fact that once we obtain significant words in the source domain, we just need to search for them in the target domain to find out common significant words.

LM2: Symmetric KL-Divergence (SKLD)
KL Divergence can be used to compare the probabilistic distribution of polar words in two domains (Kullback and Leibler, 1951). A lower KL Divergence score indicates that the probabilistic distribution of polar words in two domains

is identical. This implies that the domains are close to each other, in terms of sentiment similarity. Therefore, to rank source domains for a target domain using this metric, we inherit the concept of symmetric KL Divergence proposed by Murthy et al. (2018) and use it to compute average Symmetric KL-Divergence of common polar words shared by a domain-pair. We label a word as 'polar' for a domain if,

$$|P - N| >= 0.5 \tag{2}$$

where P is the probability of a word appearing in a review which is labelled positive and N is the probability of a word appearing in a review which is labelled negative.

SKLD of a polar word for domain-pair (D_1, D_2) is calculated as:

$$A = N_1 * log\left(\frac{N_1}{N_2}\right) + P_1 * log\left(\frac{P_1}{P_2}\right) \tag{3}$$

$$B = N_2 * log\left(\frac{N_2}{N_1}\right) + P_2 * log\left(\frac{P_2}{P_1}\right) \tag{4}$$

$$SKLD = \frac{A + B}{2} \tag{5}$$

where P_i and N_i are probabilities of a word appearing under positively labelled and negatively labelled reviews, respectively, in domain i. We then take an average of all common polar words.

We observe that, on its own, this metric performs rather poorly. Upon careful analysis of results, we concluded that the imbalance in the number of polar words being shared across domain-pairs is a reason for poor performance. To mitigate this, we compute a confidence term for a domain-pair (D_1, D_2) using the *Jaccard Similarity Coefficient* which is calculated as follows:

$$J = \frac{C}{W_1 + W_2 - C} \tag{6}$$

where C is the number of common polar words and W_1 and W_2 are number of polar words in D_1 and D_2 respectively. The intuition behind this being that the domain-pairs having higher percentage of polar words overlap should be ranked higher compared to those having relatively higher number of polar words. For example, we prefer $(C : 40, W_1 : 50, W_2 : 50)$ over $(C : 200, W_1 : 500, W_2 : 500)$ even though 200 is greater than 40. To compute the final similarity value, we add the reciprocal of J to the SKLD value since a larger value of J will add a smaller fraction to SLKD value. For a smaller SKLD value, the domains would be relatively more similar. This is computed as follows:

$$(SKLD)_{avg} + \frac{1}{J} \tag{7}$$

Domain pairs are ranked in increasing order of this similarity value. After the introduction of the confidence term, a significant improvement in the results is observed.

LM3: Chameleon Words Similarity
This metric is our novel contribution for domain adaptability evaluation. It helps in detection of '*Chameleon Word*(s)'

which change their polarity across domains (Sharma and Bhattacharyya, 2013). The motivation comes from the fact that chameleon words directly affect the CDSA accuracy. For example, *poignant* is positive in movie domain whereas negative in many other domains *viz.* Beauty, Clothing *etc.* For every common polar word between two domains, L_1 *Distance* between two vectors $[P_1, N_1]$ and $[P_2, N_2]$ is calculated as;

$$|P_1 - P_2| + |N_1 - N_2| \tag{8}$$

The overall distance is an average overall common polar words. Similar to SKLD, the confidence term based on *Jaccard Similarity Coefficient* is used to counter the imbalance of common polar word count between domain-pairs.

$$(L_1 \ Distance)_{avg} + \frac{1}{J} \tag{9}$$

Domain pairs are ranked in increasing order of final value.

LM4: Entropy Change
Entropy is the degree of randomness. A relatively lower change in entropy, when two domains are concatenated, indicates that the two domains contain similar topics and are therefore closer to each other. This metric is also our novel contribution. Using this metric, we calculate the percentage change in the entropy when the target domain is concatenated with the source domain. We calculate the entropy as the combination of entropy for unigrams, bigrams, trigrams, and quadrigrams. We consider only polar words for unigrams. For bi, tri and quadrigrams, we give priority to polar words by using a *weighted entropy function* and this weighted entropy E is calculated as:

$$P = - \sum_{i \ \epsilon \ \{X\}} p(X_i) log p(X_i) * w \tag{10}$$

$$Q = - \sum_{j \ \epsilon \ \{Y\}} p(Y_j) log p(Y_j) * \frac{1}{w} \tag{11}$$

$$E = P + Q \tag{12}$$

Here, X is the set of n-grams that contain at least one polar word, Y is the set of n-grams which do not contain any polar word, and w is the weight. For our experiments, we keep the value of w as 1 for unigrams and 5^2 for bi, tri, and quadrigrams.

We then say that a source domain D_2 is more suitable for target domain D_1 as compared to source domain D_3 if;

$$\Delta E(D_2, D_2 + D_1) < \Delta E(D_3, D_3 + D_1) \tag{13}$$

where $D_2 + D_1$ indicates combined data obtained by mixing D_1 in D_2 and ΔE indicates percentage change in entropy before and after mixing of source and target domains.

Note that this metric offers the advantage of asymmetricity, unlike the other three metrics for labelled data.

---

[2]We observe that any value of w does not change the relative ranking of domains.

4.2. Metrics: Unlabelled Data

For unlabelled target domain data, we utilize word and sentence embeddings-based similarity as a metric and use various embedding models. To train word embedding based models, we use Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017), and ELMo (Peters et al., 2018). We also exploit sentence vectors from models trained using Doc2Vec (Le and Mikolov, 2014), FastText, and Universal Sentence Encoder (Cer et al., 2018). In addition to using plain sentence vectors, we account for sentiment in sentences using SentiWordnet (Baccianella et al., 2010), where each review is given a sentiment score by taking harmonic mean over scores (obtained from SentiWordnet) of words in a review[3].

ULM1: Word2Vec
We train SKIPGRAM models on all the domains to obtain word embeddings. We build models with 50 dimensions[4] where the context window is chosen to be 5. For each domain pair, we then compare embeddings of common adjectives in both the domains by calculating *Angular Similarity* (Cer et al., 2018). It was observed that cosine similarity values were very close to each other, making it difficult to clearly separate domains. Since Angular Similarity distinguishes nearly parallel vectors much better, we use it instead of *Cosine Similarity*. We obtain a similarity value by averaging over all common adjectives. For the final similarity value of this metric, we use *Jaccard Similarity Coefficient* here as well:

$$(Angular\ Similarity)_{avg} + J \qquad (14)$$

For a target domain, source domains are ranked in decreasing order of final similarity value.
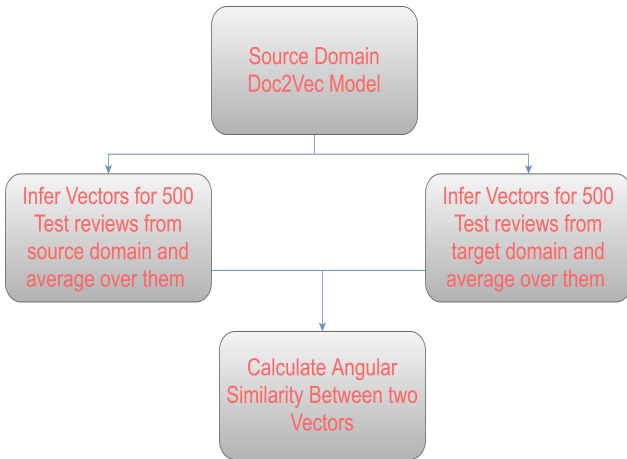

Figure 1: Experimental Setup for Doc2Vec

ULM2: Doc2Vec
Doc2Vec represents each sentence by a dense vector which is trained to predict words in the sentence, given the model.

It tries to overcome the weaknesses of the bag-of-words model. Similar to Word2Vec, we train Doc2Vec models on each domain to extract sentence vectors. We train the models over 100 epochs for 100 dimensions, where the learning rate is 0.025. Since we can no longer leverage adjectives for sentiment, we use SentiWordnet for assigning sentiment scores (ranging from -1 to +1 where -1 denotes a negative sentiment, and +1 denotes a positive sentiment) to reviews (as detailed above) and select reviews which have a score above a certain threshold. We have empirically arrived at ± 0.01 as the threshold value. Any review with a score outside this window is selected. We also restrict the length of reviews to a maximum of 100 words to reduce sparsity.

After filtering out reviews with sentiment score less than the threshold value, we are left with a minimum of 8000 reviews per domain. We train on 7500 reviews form each domain and test on 500 reviews. To compare a domain-pair (D_1, D_2) where D_1 is the source domain and D_2 is the target domain, we compute Angular Similarity between two vectors V_1 and V_2. V_1 is obtained by taking an average over 500 test vectors (from D_1) inferred from the model trained on D_1. V_2 is obtained in a similar manner, except that the test data is from D_2. Figure 1 shows the experimental setup for this metric.

ULM3: GloVe
Both Word2Vec and GloVe learn vector representations of words from their co-occurrence information. However, GloVe is different in the sense that it is a *count-based* model. In this metric, we use GloVe embeddings for adjectives shared by domain-pairs. We train GloVe models for each domain over 50 epochs, for 50 dimensions with a learning rate of 0.05. For computing similarity of a domain-pair, we follow the same procedure as described under the Word2Vec metric. The final similarity value is obtained using equation (14).

ULM4 and ULM5: FastText
We train monolingual word embeddings-based models for each domain using the FastText library[5]. We train these models with 100 dimensions and 0.1 as the learning rate. The size of the context window is limited to 5 since FastText also uses sub-word information. Our model takes into account character n-grams from 3 to 6 characters, and we train our model over 5 epochs. We use the default loss function (*softmax*) for training.
We devise **two different metrics** out of FastText models to calculate the similarity between domain-pairs. In the first metric (**ULM4**), we compute the Angular Similarity between the word vectors for all the common adjectives, and for each domain pair just like Word2Vec and GloVe. Overall, similarity for a domain pair is calculated using equation (14). As an additional metric (**ULM5**), we extract sentence vectors for reviews and follow a procedure similar to Doc2Vec. SentiWordnet is used to filter out train and test data using the same threshold window of ± 0.01.

---

[3]Github: Sentiment Classifier

[4]We train the models with different dimensions and compute scores for dimensions 50, 100, 200, and 300. We choose the dimension which gives us the best results and report it in the following sections, for each metric below.

[5]Github: FastText

	Domains	In Domain Accuracy (%)	Average CDSA Accuracy Degradation (%)	Best Source Domain	Best Target Domain
D1	Amazon Instant Video	84.84	10.82	D10	D10
D2	Automotive	83.24	5.32	D18	D15
D3	Baby	85.78	10.50	D4	D9
D4	Beauty	84.49	7.33	D3	D9
D5	Books	81.56	7.48	D1	D1
D6	Clothing Accessories	93.67	19.26	D2	D15
D7	Electronics	85.16	8.65	D18	D9
D8	Health	83.74	7.58	D4	D15
D9	Home Kitchen	87.01	10.29	D12	D19
D10	Movies TV	84.50	10.08	D1	D1
D11	Music	84.98	13.43	D1	D15
D12	Office Products	84.79	6.45	D18	D9
D13	Patio	83.74	5.84	D18	D18
D14	Pet Supplies	84.81	8.19	D8	D9
D15	Shoes	95.48	23.18	D6	D17
D16	Software	84.49	9.19	D12	D12
D17	Sports Outdoors	88.18	11.12	D6	D6
D18	Tools Home Improvement	82.18	4.01	D13	D13
D19	Toys Games	86.77	10.83	D20	D15
D20	Video Games	83.19	9.26	D19	D19

Table 3: Our reccomendation chart based on CDSA results: In Domain Accuracy(when source and target domains are same), Average CDSA Accuracy Degradation(average cross-domain testing accuracy loss over all target domains), Best Source Domain, Best Target Domain for each domain.

	Top 3		Top 5		Top 7		Top 10	
	Precision (%)	NRA	Precision (%)	NRA	Precision (%)	NRA	Precision (%)	NRA
				Labelled				
LM1	45.00	0.200	54.00	**0.190**	64.29	**0.150**	72.00	0.060
LM2	**46.67**	0.183	**62.00**	0.150	62.86	0.114	77.50	0.055
LM3	**46.67**	0.183	**62.00**	0.150	63.57	0.114	**78.00**	0.055
LM4	**46.67**	**0.233**	59.00	0.180	**69.29**	0.129	76.00	**0.110**
				Unlabelled				
ULM1	51.67	0.267	62.00	**0.210**	66.43	**0.186**	77.50	0.080
ULM2	40.00	0.100	49.00	0.110	61.43	0.122	76.00	0.100
ULM3	41.67	0.150	50.00	0.140	59.29	0.136	73.50	0.045
ULM4	51.67	0.217	55.00	0.170	62.14	0.157	75.50	0.065
ULM5	45.00	0.267	54.00	0.180	60.71	0.157	68.00	**0.125**
ULM6	56.67	0.233	63.00	0.200	70.00	0.179	**81.00**	0.070
ULM7	**58.33**	**0.300**	**64.00**	**0.210**	**70.72**	0.179	80.00	0.090

Table 4: Precision and Normalised Ranking Accuracy (NRA) for top-K source domain matching over all domains.

ULM6: ELMo

We use the pre-trained deep contextualized word representation model provided by the ELMo library[6]. Unlike Word2Vec, GloVe, and FastText, ELMo gives multiple embeddings for a word based on different contexts it appears in the corpus.

In ELMo, higher-level LSTM states capture the context-dependent aspects of word meaning. Therefore, we use only the topmost layer for word embeddings with 1024 dimensions. Multiple contextual embeddings of a word are averaged to obtain a single vector. We again use average Angular Similarity of word embeddings for common adjectives to compare domain-pairs along with *Jaccard Similarity Coefficient*. The final similarity value is obtained using equation (14).

ULM7: Universal Sentence Encoder

One of the most recent contributions to the area of sentence embeddings is the Universal Sentence Encoder. Its transformer-based sentence encoding model constructs sentence embeddings using the encoding sub-graph of the transformer architecture (Vaswani et al., 2017). We leverage these embeddings and devise a metric for our work.

We extract sentence vectors of reviews in each domain using tensorflow-hub model toolkit[7]. The dimensions of each vector are 512. To find out the similarity between a domain-pair, we extract top 500 reviews from both domains based on the sentiment score acquired using SentiWordnet (as detailed above) and average over them to get two vectors with 512 dimensions each. After that, we find out the Angular Similarity between these vectors to rank all source domains

---

[6]GitHub: ELMo

[7]TensorFlow Hub

for a particular target domain in decreasing order of similarity.

5. Results

We show the results of the classifier's CDSA performance followed by metrics evaluation on the top 10 domains. Finally, we present an overall comparison of metrics for all the domains.

Table 3 shows the average CDSA accuracy degradation in each domain when it is selected as the source domain, and the rest of the domains are selected as the target domain. We also show in-domain sentiment analysis accuracy, the best source domain (on which CDSA classifier is trained), and the best target domain (on which CDSA classifier is tested) in the table. D15 suffers from the maximum average accuracy degradation, and D18 performs the best with least average accuracy degradation, which is also supported by its number of appearances *i.e.,* 4, as the best source domain in the table. As for the best target domain, D9 appears the maximum number of times.

To compare metrics, we use two parameters: *Precision* and *Ranking Accuracy*.

- **Precision:** It is the intersection between the top-K source domains predicted by the metric and top-K source domains as per CDSA accuracy, for a particular target domain. In other words, it is the number of true positives.

- **Ranking Accuracy (RA):** It is the number of predicted source domains that are **ranked correctly** by the metric.

Figure 2 shows the number of true positives (precision) when K = 5 for each metric over the top 10 domains. The X-axis denotes the domains, whereas the Y-axis in the bar graph indicates the precision achieved by all metrics in each domain. We observe that the highest precision attained is 5, by 4 different metrics. We also observe that all the metrics reach a precision of at least 1. A similar observation is made for the remaining domains as well. Figure 3 displays the RA values of K = 5 in each metric for the top 10 domains. Here, the highest number of correct source domain rankings attained is 4 by ULM6 (ELMo) for domain D5.

Table 4 shows results for different values of K in terms of precision percentage and normalized RA (NRA) over all domains. Normalized RA is RA scaled between 0 to 1. For example, entries 45.00 and 0.200 indicate that there is 45% precision with NRA of 0.200 for the top 3 source domains. These are the values when the metric LM1 (Significant Words Overlap) is used to predict the top 3 source domains for all target domains. Best figures for precision and NRA have been shown in bold for all values of K in both labelled as well as unlabelled data metrics. ULM7 (Universal Sentence Encoder) outperforms all other metrics in terms of both precision and NRA for K = 3, 5, and 7. When K = 10, however, ULM6 (ELMo) outperforms ULM7 marginally at the cost of a 0.02 degradation in terms of NRA. For K = 3 and 5, ULM2 (Doc2Vec) has the least precision percentage

and NRA, but UML3 (GloVe) and ULM5 (FastText Sentence) take the lowest pedestal for K = 7 and K = 10 respectively, in terms of precision percentage.

6. Discussion

Table 3 shows that, if a suitable source domain is not selected, CDSA accuracy takes a hit. The degradation suffered is as high as 23.18%. This highlights the motivation of these experiments: the choice of a source domain is critical. We also observe that the automative domain (D2) is the best source domain for clothing (D6), both being unrelated domains in terms of the products they discuss. This holds for many other domain pairs, implying that mere intuition is not enough for source domain selection.

From the results, we observe that LM4, which is one of our novel metrics, predicts the best source domain correctly for D_2 and D_4, which all other metrics fail to do. This is a good point to highlight the fact that this metric captures features missed by other metrics. Also, it gives the best RA for K=3 and 10. Additionally, it offers the advantage of asymmetricity unlike other metrics for labelled data.

For labelled data, we observe that LM2 (Symmetric KL-Divergence) and LM3 (Chameleon Words Similarity) perform better than other metrics. Interestingly, they also perform identically for K = 3 and K = 5 in terms of both precision percentage and NRA. We accredit this observation to the fact that both determine the distance between probabilistic distributions of polar words in domain-pairs.

Amongst the metrics which utilize word embeddings, ULM1 (Word2Vec) outperforms all other metrics for all values of K. We also observe that word embeddings-based metrics perform better than sentence embeddings-based metrics. Although ULM6 and ULM7 outperform every other metric, we would like to make a note that these are computationally intensive models. Therefore, there is a trade-off between the performance and time when a metric is to be chosen for source domain selection. The reported NRA is low for all the values of K across all metrics. We believe that the reason for this is the unavailability of enough data for the metrics to provide a clear distinction among the source domains. If a considerably larger amount of data would be used, the NRA should improve.

We suspect that the use of ELMo and Universal Sentence Encoder to train models for contextualized embeddings on review data in individual domains should improve the precision for ULM6 (ELMo) and ULM7 (Universal Sentence Encoder). However, we cannot say the same for RA as the amount of corpora used for pre-trained models is considerably large. Unfortunately, training models using both these recur a high cost, both computationally and with respect to time, which defeats the very purpose of our work *i.e.,* to pre-determine best source domain for CDSA using non-intensive text similarity-based metrics.

7. Conclusion and Future Work

In this paper, we investigate how text similarity-based metrics facilitate the selection of a suitable source domain for CDSA. Based on a dataset of reviews in 20 domains, our recommendation chart that shows the best source and target
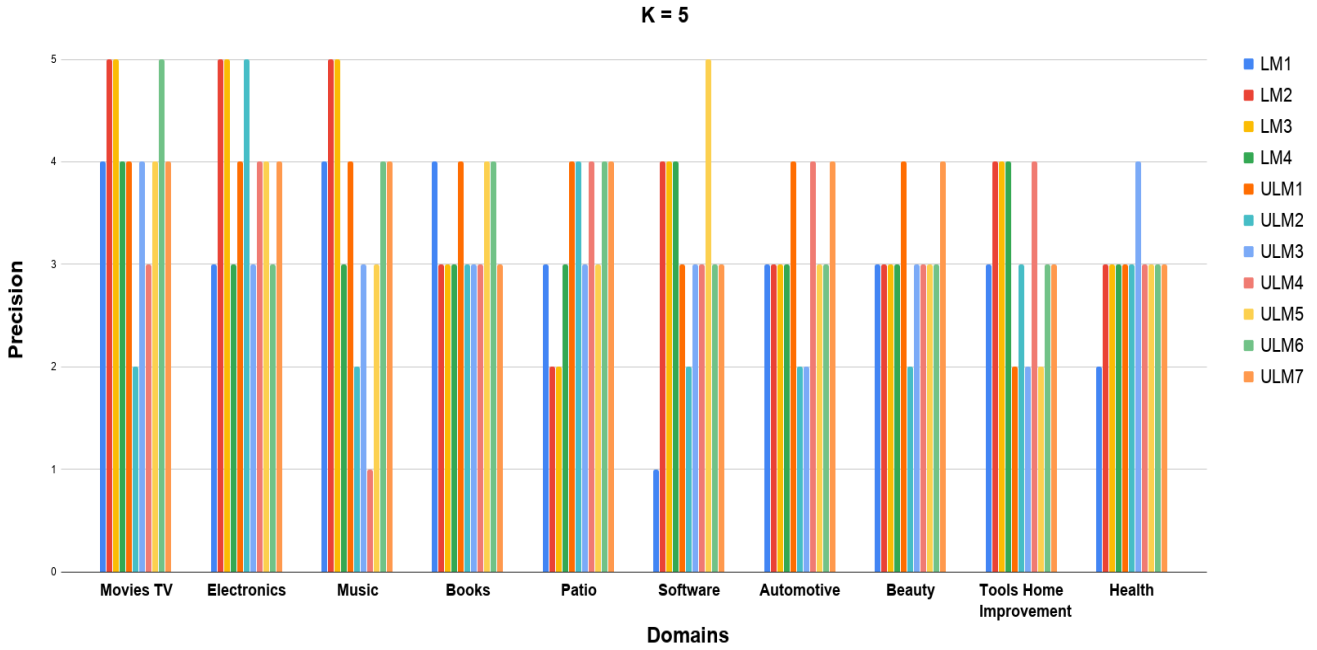
Figure 2: Precision for K=5 over top 10 domains. Precision is the intersection between the top-K source domains predicted by the metric and top-K source domains as per CDSA accuracy.
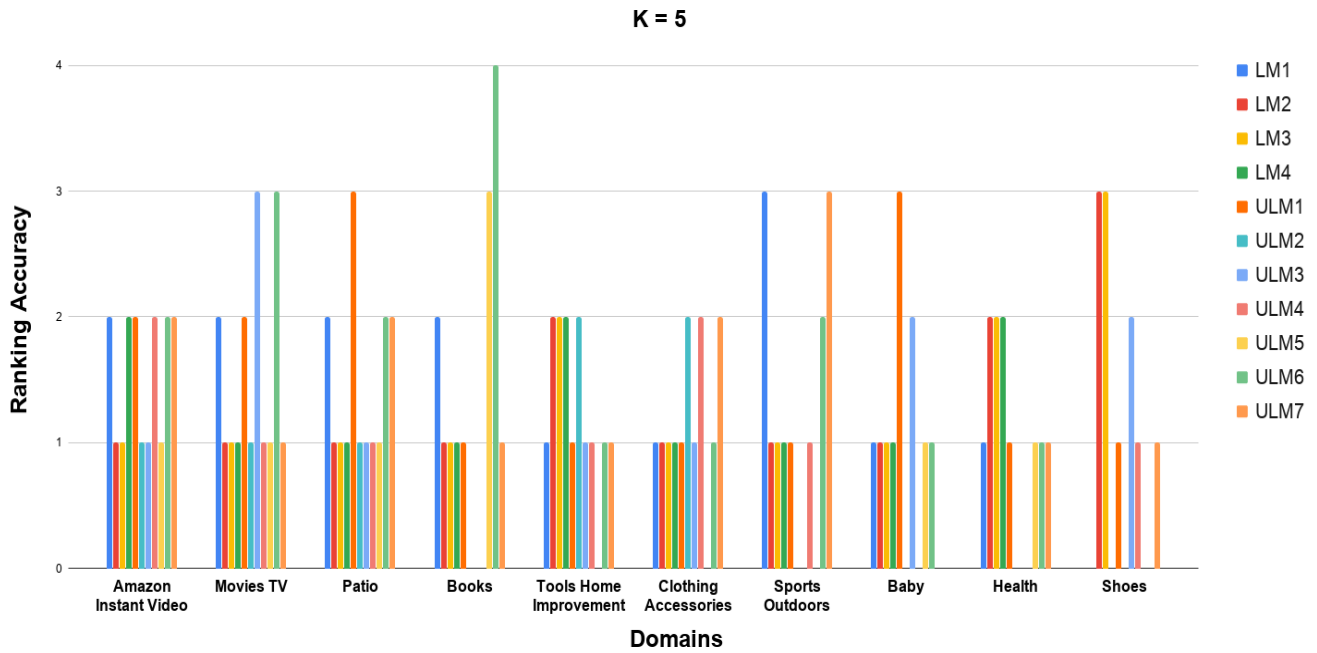

Figure 3: Ranking Accuracy for K=5 over top 10 domains. Ranking accuracy is the number of predicted source domains which are **ranked correctly** by the metric.

domain pairs for CDSA would be useful for deployments of sentiment classifiers for these domains.

In order to compare the benefit of a domain with similarity metrics between the source and target domains, we describe a set of symmetric and asymmetric similarity metrics. These also include two novel metrics to evaluate domain adaptability: namely as LM3 (Chameleon Words Similarity) and LM4 (Entropy Change). These metrics perform at par with the metrics that use previously proposed methods. We observe that, amongst word embedding-based metrics, ULM6 (ELMo) performs the best, and amongst

sentence embedding-based metrics, ULM7 (Universal Sentence Encoder) is the clear winner. We discuss various metrics, their results and provide a set of recommendations to the problem of source domain selection for CDSA.

A possible future work is to use a weighted combination of multiple metrics for source domain selection. These similarity metrics may be used to extract suitable data or features for efficient CDSA. Similarity metrics may also be used as features to predict the CDSA performance in terms of accuracy degradation.

8. Bibliographical References

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144.

Blitzer, J., McDonald, R., and Pereira, F. (2006). 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.

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

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Chen, M., Weinberger, K. Q., and Blitzer, J. (2011). Co-training for domain adaptation. In *Advances in neural information processing systems*, pages 2456–2464.

Dai, X., Karimi, S., Hachey, B., and Paris, C. (2019). Using similarity measures to select pretraining data for ner. *arXiv preprint arXiv:1904.00585*.

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

Denecke, K. (2009). Are sentiwordnet scores suited for multi-domain sentiment classification? In *2009 Fourth International Conference on Digital Information Management*, pages 1–6. IEEE.

Dragoni, M., Tettamanzi, A. G., and Pereira, C. D. C. (2016). Dranziera: an evaluation protocol for multi-domain opinion mining. In *Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 267–272. European Language Resources Association (ELRA).

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

He, Y., Lin, C., and Alani, H. (2011). Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 123–131. Association for Computational Linguistics.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, pages 1188–1196.

Li, F., Pan, S. J., Jin, O., Yang, Q., and Zhu, X. (2012). Cross-domain co-extraction of sentiment and topic lexicons. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 410–419. Association for Computational Linguistics.

Li, S., Xue, Y., Wang, Z., and Zhou, G. (2013). Active learning for cross-domain sentiment classification. In *Twenty-Third IJCAI*.

Liu, B. et al. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2(2010):627–666.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Murthy, R., Kunchukuttan, A., and Bhattacharyya, P. (2018). Judicious selection of training data in assisting language for multilingual neural ner. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 401–406.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on WWW*, pages 751–760. ACM.

Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

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

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

Schultz, L. R., Loog, M., and Esfahani, P. M. (2018). Distance based source domain selection for sentiment classification. *arXiv preprint arXiv:1808.09271*.

Sharma, R. and Bhattacharyya, P. (2013). Detecting domain dedicated polar words. In *Proceedings of the Sixth IJCNLP*, pages 661–666.

Sharma, R., Bhattacharyya, P., Dandapat, S., and Bhatt, H. S. (2018). Identifying transferable information across domains for cross-domain sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 968–978.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhang, Y. and Wang, R. (2009). Cross-domain dependency parsing using a deep linguistic grammar. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 378–386.