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

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.

Introduction

Sentiment analysis (SA) deals with automatic detection of opinion orientation in text BIBREF0. 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 BIBREF1.

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'). ben2007analysis 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:

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.

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.

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 SECREF2 We then introduce our sentiment classifier in Section SECREF3 and the similarity metrics in Section SECREF4 The results are presented in Section SECREF5 followed by a discussion in Section SECREF6 Finally, we conclude the paper in Section SECREF7

Related Work

Cross-domain adaptation has been reported for several NLP tasks such as part-of-speech tagging BIBREF2, dependency parsing BIBREF3, and named entity recognition BIBREF4. Early work in CDSA is by denecke2009sentiwordnet. 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 BIBREF5, co-training BIBREF6 or spectral feature alignment BIBREF7. In terms of the use of topic models for CDSA, he2011automatically 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 BIBREF8. glorot2011domain 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 dai2019using uses similarity metrics to select the domain from which pre-trained embeddings should be obtained for named entity recognition. Similarly, schultz2018distance 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.

Sentiment Classifier

The core of this work is a sentiment classifier for different domains. We use the DRANZIERA benchmark dataset BIBREF9, 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. 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 TABREF5.

Similarity Metrics

In table TABREF6, 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.

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.

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.

Similarity Metrics ::: 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:

Similarity Metrics ::: Metrics: Labelled Data ::: 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 sharma2018identifying and extract significant words from each domain using the $\chi ^2$ 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 $\chi ^2$ value greater than or equal to 1. The $\chi ^2$ value is calculated as follows:

Where ${c_p}^w$ and ${c_n}^w$ are the observed counts of word w in positive and negative reviews, respectively. $\mu ^w$ 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.

Similarity Metrics ::: Metrics: Labelled Data ::: LM2: Symmetric KL-Divergence (SKLD)

KL Divergence can be used to compare the probabilistic distribution of polar words in two domains BIBREF10. 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 murthy2018judicious 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,

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:

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:

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:

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.

Similarity Metrics ::: Metrics: Labelled Data ::: 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 BIBREF11. 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;

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.

Domain pairs are ranked in increasing order of final value.

Similarity Metrics ::: Metrics: Labelled Data ::: 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:

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 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;

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.

Similarity Metrics ::: 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 BIBREF12, GloVe BIBREF13, FastText BIBREF14, and ELMo BIBREF15. We also exploit sentence vectors from models trained using Doc2Vec BIBREF16, FastText, and Universal Sentence Encoder BIBREF17. In addition to using plain sentence vectors, we account for sentiment in sentences using SentiWordnet BIBREF18, where each review is given a sentiment score by taking harmonic mean over scores (obtained from SentiWordnet) of words in a review.

Similarity Metrics ::: Metrics: Unlabelled Data ::: ULM1: Word2Vec

We train SKIPGRAM models on all the domains to obtain word embeddings. We build models with 50 dimensions 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 BIBREF17. 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:

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

Similarity Metrics ::: Metrics: Unlabelled Data ::: 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 FIGREF30 shows the experimental setup for this metric.

Similarity Metrics ::: Metrics: Unlabelled Data ::: 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 (DISPLAY_FORM29).

Similarity Metrics ::: Metrics: Unlabelled Data ::: ULM4 and ULM5: FastText

We train monolingual word embeddings-based models for each domain using the FastText library. 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 (DISPLAY_FORM29). 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.

Similarity Metrics ::: Metrics: Unlabelled Data ::: ULM6: ELMo

We use the pre-trained deep contextualized word representation model provided by the ELMo library. 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 (DISPLAY_FORM29).

Similarity Metrics ::: Metrics: Unlabelled Data ::: 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 BIBREF19. 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. 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 for a particular target domain in decreasing order of similarity.

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 TABREF31 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 FIGREF36 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 FIGREF37 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 TABREF33 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.

Discussion

Table TABREF31 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.

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 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.