Learning from Domain Complexity

Robert Remus[†], **Dominique Ziegelmayer**[‡]

†Natural Language Processing Group, University of Leipzig, Germany
‡Institut für Informatik, University of Cologne, Germany
rremus@informatik.uni-leipzig.de, ziegelmayer@zpr.uni-koeln.de

Abstract

Sentiment analysis is genre and domain dependent, i. e. the same method performs differently when applied to text that originates from different genres and domains. Intuitively, this is due to different language use in different genres and domains. We measure such differences in a sentiment analysis gold standard dataset that contains texts from 1 genre and 10 domains. Differences in language use are quantified using certain language statistics, viz. domain complexity measures. We investigate 4 domain complexity measures: percentage of rare words, word richness, relative entropy and corpus homogeneity. We relate domain complexity measurements to performance of a standard machine learning-based classifier and find strong correlations. We show that we can accurately estimate its performance based on domain complexity using linear regression models fitted using robust loss functions. Moreover, we illustrate how domain complexity may guide us in model selection, viz. in deciding what word n-gram order to employ in a discriminative model and whether to employ aggressive or conservative word n-gram feature selection.

Keywords: Domain complexity; model selection; sentiment analysis

1. Introduction

Sentiment Analysis (SA) is—just like natural language processing in general (Sekine, 1997; Escudero et al., 2000)—*genre and domain dependent* (Wang and Liu, 2011), i.e. the same SA method performs differently when applied to different genres and domains. Intuitively, this is due to *different language use* in different *genres*[1] and *domains*[2].

In this paper, we measure such differences in an SA gold standard dataset that contains texts from 1 genre—product reviews—and 10 domains, e. g. apparel and music. Differences in language use are quantified using certain language statistics, viz. *domain complexity measures*.

We relate domain complexity to performance of a standard Machine Learning (ML)-based classifier and find strong correlations. We show that we can accurately *estimate its performance* based on domain complexity. Moreover, we illustrate how domain complexity may *guide us in model selection*.

1.1. Related Work

Related to our work are Ponomareva and Thelwall (2012), who estimate—using domain complexity and domain similarity—the accuracy loss when transferring an SA method from a source to a target domain. Van Asch and Daelemans (2010) estimate—using domain similarity—the accuracy loss when transferring a part of speech-tagger from one domain to another. Blitzer et al. (2007) compute an $\mathcal{A}$-distance proxy and show that it correlates with accuracy loss when transferring their SA method from a source to a target domain.

[1] A *genre* is an identifiable text category (Crystal, 2008, p. 210) based on external, non-linguistic criteria such as intended audience, purpose, and activity type (Lee, 2001) as well as textual structure, form of argumentation, and level of formality (Crystal, 2008, p. 210).

[2] A *domain* is a genre attribute that describes the subject area that an instantiation of a certain genre deals with (Steen, 1999; Lee, 2001).

1.2. Outline

This paper is structured as follows: In Section 2. we describe domain complexity measures that we use to quantify differences in language use. In Section 3. we relate domain complexity to performance and show what we can *learn* from their relationship. In Section 4. we conclude and point out directions for future work.

2. Domain Complexity

Domain complexity is a measure that "reflects the difficulty of [a] classification task for a given data set" (Ponomareva and Thelwall, 2012). We use 4 approximations of domain complexity: 3 approximations proposed by Ponomareva and Thelwall (2012) and 1 proposed by Remus (2012).

2.1. Ponomareva and Thelwall (2012)

Ponomareva and Thelwall (2012) proposed 3 measures as approximations of domain complexity:

Percentage of rare words. The percentage of rare words is defined as in Equation (1)

$$PRW = \frac{|\{w \in W \mid c(w) < 3\}|}{|W|} \tag{1}$$

where W is the vocabulary, vocabulary size $|W|$ equals the number of *types*, i. e. the number of different words in a text sample, and $c(w)$ is the number of occurrences of w in a text sample.

Word richness. The word richness is defined just as the *type/token ratio* TTR in Equation (2)

$$TTR = \frac{|W|}{\sum_{w \in W} c(w)} \tag{2}$$

where $\sum_{w \in W} c(w)$ equals the number of *tokens*, i. e. the total number of words in a text sample (Crystal, 2008, p. 498). From hereon, we refer to word richness by type/token ratio.

Relative entropy. The relative entropy is defined as in Equation (3)

$$H_{\text{rel}} = \frac{H}{H_{\text{max}}} \tag{3}$$

where H as in Equation (4)

$$H = -\sum_{w \in W} p(w) \log_2 p(w) \tag{4}$$

is the *entropy* of W's distribution and H_{max} as in Equation (5)

$$\begin{aligned} H_{\text{max}} &= -\sum_{w \in W} \frac{1}{|W|} \log_2 \frac{1}{|W|} \\ &= \log_2 |W| \end{aligned} \tag{5}$$

is the *maximum entropy* of W's distribution, i. e. its entropy if W was distributed uniformly.

2.2. Remus (2012)

Remus (2012) proposed *corpus homogeneity* (Kilgarriff, 2001) as another approximation of domain complexity. Corpus homogeneity or *corpus self-similarity* uses repeated random subsampling validation and is estimated as shown in Pseudocode 1.

Pseudocode 1: Corpus homogeneity.

```
for i = 1, ..., k {
    shuffle corpus c
    split c into 2 equally-sized subcorpora c1, c2
    selfsimilarity si:= sim(c1, c2)
}
homogeneity Hom := average(s1,...,sk)
```

We use *Jensen-Shannon (JS) divergence* as similarity function $\texttt{sim}(c_1, c_2)$. The JS divergence (Lin, 1991) is based on the Kullback-Leibler (KL) divergence D_{KL} (Kullback and Leibler, 1951) as given in Equation (6)

$$D_{\text{KL}}(Q||R) = \sum_{w \in W} Q(w) \log \frac{Q(w)}{R(w)} \tag{6}$$

where Q and R are probability distributions over a finite set W, e.g. words. The JS divergence D_{JS} is then defined as shown in Equation 7

$$D_{\text{JS}}(Q||R) = \frac{1}{2}\left[D_{\text{KL}}(Q||M) + D_{\text{KL}}(R||M)\right] \tag{7}$$

where $M = \frac{1}{2}(Q + R)$ is the average distribution of Q and R and $0 \leq D_{\text{JS}}(Q||R) \leq 1$. The larger D_{JS}, the more the probability distributions diverge. For $k \to \infty$, the homogeneity estimate approaches the "actual" corpus homogeneity. We set k to 10 (Remus, 2012).

2.3. Sample Size Normalization

The domain complexity measures described in the previous sections are *sample size dependent* (Remus and Bank, 2012). Therefore, we compute percentage of rare words, type/token ratio, and relative entropy on *fixed length subsamples* rather than on the full sample as shown in Pseudocode 2.

Pseudocode 2: Sample size-normalized domain complexity

```
for i = 1, ..., k {
    subsample si := extract random word window of size
        1000 from full sample
    measurement mi := domain complexity(si)
}
normalized domain complexity := average(m1,...,mk)
```

Using a sufficient number of iterations k—10,000 in our case—we obtain a stable approximation of the expected domain complexity value, which is *normalized* with respect to sample size.

To compute sample size-normalized homogeneity, we proceed as shown in Pseudocode 1. But instead of shuffling the corpus and splitting it into 2 equally-sized subcorpora, we randomly extract 2 fixed length subsamples s_i^1, s_i^2 analogously to Pseudocode 2 with the constraint that s_i^1, s_i^2 must not overlap. We then measure $\texttt{sim}(s_i^1, s_i^2)$. In deviation from corpus homogeneity as described in Section 2.2. we here set k to 10,000 instead of 10.

3. Learning from Domain Complexity

In this section we *learn* from an SA gold standard dataset's domain complexity: In Section 3.1. we relate differences in domain complexity in an SA gold standard dataset to the differences in performance of a standard ML-based classifier evaluated on the same SA gold standard dataset. In Section 3.2. and Section 3.3. we let domain complexity guide us in model selection, viz. in deciding what word n-gram order to employ in a discriminative model. In Section 3.3. we let domain complexity guide us in deciding whether to employ aggressive or conservative word n-gram feature selection.

We analyze a common SA subtask: *document-level polarity classification*. Our experimental setup for this subtask is as follows: We use Blitzer et al. (2007)'s Multi-Domain Sentiment Dataset v2.0 (MDSD v2.0)[3] as gold standard dataset. MDSD v2.0 contains star-rated *product reviews* of *various domains*. We chose 10 domains: apparel, books, dvd, electronics, health & personal care, kitchen & housewares, music, sports & outdoors, toys & games, and videos. Those are exactly the domains for which a balanced amount of 1,000 positive and 1,000 negative reviews is available. Blitzer et al. (2007) considered reviews with more than 3 stars positive, and less than 3 stars negative; so do we.

For sentence segmentation and tokenization of MDSD v2.0 we use OpenNLP[4]. As classifiers we employ Support Vector Machines (SVMs) (Vapnik, 1995) as implemented by LibSVM[5] using a linear kernel with their optimal cost factor C chosen from {2.0E-3, 2.0E-2, 2.0E-1, 2.0, 2.0E1, 2.0E2, 2.0E3} via 10-fold cross validation (CV) on the training data. As features we use word uni-, bi-, and/or trigrams extracted from the training data by a purely data-driven feature induction (Remus and Rill, 2013). We simply encode presence or absence of those word n-grams. We perform no feature selection; neither stop words nor punc-

[3] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[4] http://opennlp.apache.org

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 2: Pearson correlation r between domain complexity measurements and accuracies of SVM models based on word unigrams as well as r's significance level p.

Domain complexity measure	r	p
Percentage of rare unigrams	-0.673	0.023
Unigram type/token ratio	-0.723	0.012
Unigram relative entropy	-0.425	0.192
Unigram homogeneity	-0.708	0.015

tuation characters are removed. From hereon we refer to this classifier as *our SA method*.

All classification experiments are *binary* and construed as 10-fold CVs. In each fold $^9/_{10}$th of the available data are used for training, the remaining $^1/_{10}$th is used for testing. Training and testing data never overlap. As performance measure we report *accuracy* A.

3.1. Performance Estimation

In this section we relate differences in domain complexity of an SA gold standard dataset to differences in performance of our SA method evaluated on the same SA gold standard dataset. Table 1 depicts the differences in performance when we evaluate our SA method—viz. SVM models based on word unigrams, uni- and bigrams, and uni-, bi-, and trigrams—on different domains from MDSD v2.0. Table 1 also depicts the differences in domain complexity of the same domains.

We correlate the accuracies achieved by our SVM models based on word unigrams and the corresponding domain complexity measurements. Table 2 shows the results. All correlations—except of unigram relative entropy—are strong ($|r| > 0.67$) and statistically significant ($p < 0.05$). From Table 2 we learn that

- the smaller the percentage of rare unigrams, i. e. the less hapax legomena and dis legomena,
- the smaller the unigram type/token ratio, i. e. the more tokens per type,
- the smaller the unigram relative entropy, i. e. the farther the distribution from a uniform distribution and
- the smaller the unigram homogeneity value, i. e. the more homogeneous the corpus,

the higher the accuracy of our SA method.

Given such strong correlations, we perform an ordinary Linear Regression (LR) using squared error loss with single domain complexity measurements as *predictors*[6] and single accuracies as *responses*. We measure the mean residual standard error (MRSE) of the LR models in leave-one-domain-out CVs[7].

[6]We do not use more than one predictor in our LR models in accordance with Harrell (2001, p. 61), who suggests to obey the rule of thumb $p < {}^n/_{10}$ where p is the number of predictors and n is the total sample size. In our leave-one-domain-out CV experiments $n = 9$ (and hence $1 > {}^9/_{10}$).

[7]Leave-one-(domain)-out CV (Hastie et al., 2009, p. 242) is a special case of a K-fold CV, where $K = n$ and n is the number of instances—e. g. domains—in the data: we fit a model to $n - 1$ parts of the data and validate it on the held out n-th part. The n results are then averaged.

Table 3: MRSEs of ordinary LR models fitted using squared error loss in leave-one-domain-out CVs with domain complexity measurements as predictors and accuracies of SVM models based on word unigrams as responses.

Predictor	**MRSE**	p
Percentage of rare unigrams	1.238	0.033
Unigram type/token ratio	1.116	0.018
Unigram relative entropy	1.837	0.221
Unigram homogeneity	*1.058*	0.007

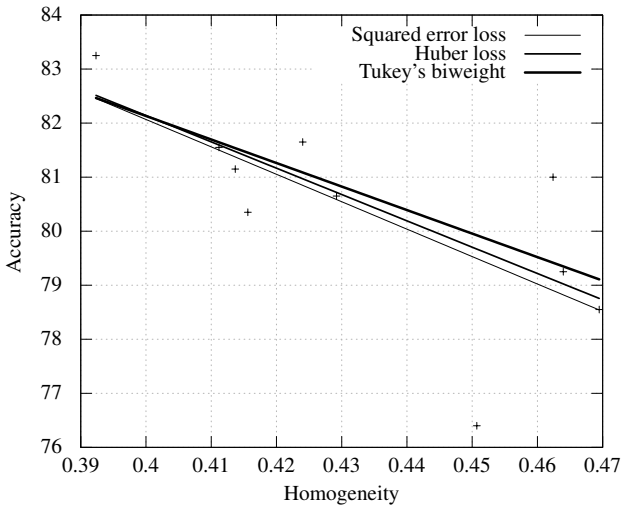

Figure 1: Accuracy of an SVM model based on word unigrams vs. unigram homogeneity plus LR models fitted using squared error loss, Huber loss, and Tukey's biweight.

Table 3 shows the resulting MRSEs as well as the significance level p of the predictor's influence on the response. All predictors' influences on the response—except of unigram relative entropy—are statistically significant ($p < 0.05$).

From Table 3 we learn that—analogously to the correlations we found—3 out of 4 domain complexity measures allow us to accurately estimate our SA method's performance based solely on domain complexity measurement. Unigram homogeneity appears to be the most informative domain complexity measure: it yields the smallest MRSE (1.058).

As we can see in Figure 1 our data contains (at least) one outlier: the domain MUSIC with an accuracy of 76.4 and a unigram homogeneity of 0.451. Outliers such as MUSIC affect the slope of the LR fit. We counteract outliers by employing loss functions that are more *robust* than ordinary squared error loss: Huber loss (Huber, 1964) and Tukey's biweight (Holland and Welsch, 1977). Using these robust loss functions in LR leads to small improvements in estimating accuracy—i. e. reduces the MRSEs—as shown in Table 4.

Figure 1 depicts LR models fitted to our data using squared

Table 1: Accuracies on MDSD v2.0 of SVM models based on word unigrams, uni- and bigrams, or uni-, bi-, and trigrams as well as domain complexity measurements.

Domain	**Our SA method's accuracy**			**Domain complexity measure**			
	uni	uni, bi	uni, bi, tri	PRW	TTR	H_{rel}	Hom
APPAREL	83.25	85.55	85.05	0.6801	0.4294	0.8891	0.3923
BOOKS	79.25	79.65	79.5	0.7261	0.4595	0.8871	0.4640
DVD	78.55	79.8	79.25	0.7244	0.4631	0.8897	0.4695
ELECTRONICS	80.65	82.05	81.6	0.6916	0.4409	0.8888	0.4292
HEALTH	80.35	83.55	83.45	0.6813	0.4306	0.888	0.4156
KITCHEN	81.15	82.1	81.85	0.6857	0.4369	0.8879	0.4137
MUSIC	76.4	78.45	78.9	0.7175	0.4600	0.8926	0.4507
SPORTS	81.65	83	82.95	0.687	0.4361	0.88789	0.4240
TOYS	81.55	83.15	82.75	0.6765	0.4308	0.8916	0.4112
VIDEO	81	81.65	81.65	0.7248	0.4618	0.8882	0.4624

Table 4: MRSEs of robust LR models fitted using Huber loss and Tukey's biweight in leave-one-domain-out CVs with domain complexity measurements as predictors and accuracies of SVM models based on word unigrams as responses.

Predictor	**Huber**	**Tukey's**
Percentage of rare unigrams	1.205	1.208
Unigram type/token ratio	1.054	1.073
Unigram relative entropy	1.959	2.050
Unigram homogeneity	*1.027*	1.082

error loss, Huber loss, and Tukey's biweight. Both robust LR models are less influenced by outliers. Thus, they result in a more accurate fit of the data, especially when applied to subsamples of the data as in our leave-one-domain-out CVs.

Performance estimation does not only work for SVM models based on word unigrams, but also for SVM models based on higher order word n-grams, i. e. SVM models based on word uni- and bigrams and SVM models based on word uni-, bi-, and trigrams: we just use higher order word n-gram domain complexity measurements as *additional predictors* in our LR models. E. g., to estimate the accuracy of an SVM model based on word uni- and bigrams, we measure both word unigram relative entropy and word bigram relative entropy, or both unigram type/token ratio and bigram type/token ratio etc. These additional predictors are either *kept separately* or *averaged*. Averaging predictors, e. g. averaging word uni-, bi-, and trigram relative entropy, results in a single predictor in our LR models. Keeping predictors separately results in multiple predictors in our LR models.

We then proceed as described earlier. Results of the accuracy estimation for SVM models based on word uni- and bigrams are shown in Table 5. Results of the accuracy estimation for SVM models based on word uni-, bi-, and trigrams are shown in Table 6.

For accuracy estimation of SVM models based on word uni- and bigrams using percentage of rare words as separate (i. e. not averaged) predictors and an LR model fitted using Tukey's biweight yields the smallest MRSE (0.472). For accuracy estimation of SVM models based on word uni-, bi-, and trigrams using type/token ratio as separate (i. e. not averaged) predictors and an LR model fitted using Tukey's biweight yields the smallest MRSE (0.634).

Table 5: MRSEs of LR models fitted using squared error loss, Huber loss, and Tukey's biweight in leave-one-domain-out CVs with domain complexity measurements as predictors and accuracies of SVM models based on word uni- and bigrams as responses. "sep" denotes separately kept predictors, "avg" denotes averaged predictors.

Predictor(s)		**Squared**	**Huber**	**Tukey's**
PRW	sep	0.94	0.506	*0.472*
	avg	0.963	0.905	0.907
TTR	sep	0.942	0.591	0.579
	avg	0.921	0.777	0.765
H_{rel}	sep	0.902	0.882	0.87
	avg	1.604	1.514	1.464
Hom	sep	1.02	1.063	1.067
	avg	0.927	0.925	0.958

Table 6: MRSEs of LR models fitted using squared error loss, Huber loss, and Tukey's biweight in leave-one-domain-out CVs with domain complexity measurements as predictors and accuracies of SVM models based on word uni-, bi-, and trigrams as responses. "sep" denotes separately kept predictors, "avg" denotes averaged predictors.

Predictor(s)		**Squared**	**Huber**	**Tukey's**
PRW	sep	1.143	0.867	0.738
	avg	0.913	0.943	0.928
TTR	sep	1.048	0.747	*0.634*
	avg	0.781	0.713	0.75
H_{rel}	sep	1.002	1.022	1.049
	avg	1.43	1.429	1.620
Hom	sep	1.037	0.996	0.904
	avg	0.877	0.854	0.894

Discussion

Our performance estimates are not 100% accurate: on average we over- or underestimate our SA approach's performance by about 1 accuracy point. Partly, this is because a discriminative model's power ultimately also depends on

Table 7: Accuracies of our model selector for word n-gram model order. "1–2" denotes first vs. second order, "2–3" denotes second vs. third order. "sep" denotes separately kept predictors, "avg" denotes averaged predictors.

Predictor(s)		**Squared**			**Huber**			**Tukey's**		
		1–2	2–3		1–2	2–3		1–2	2–3	
PRW	sep	100	80	90	90	70	80	80	50	65
	avg	100	80	90	100	70	85	100	40	70
TTR	sep	90	90	90	90	80	85	90	80	85
	avg	100	80	90	100	60	80	90	40	65
H_{rel}	sep	60	80	70	60	70	65	60	70	65
	avg	90	80	85	90	70	80	80	60	70
Hom	sep	100	80	90	100	80	90	90	70	80
	avg	100	80	90	100	90	95	90	90	90

e. g. whether the gold standard dataset contains erroneous labels, its size, and its class boundary complexity (Ho and Basu, 2002).

3.2. Model Selection: Word n-gram Model Order

In this section, we let domain complexity guide us in model selection, viz. in deciding what word n-gram model order to employ in our SA method for a given domain from MDSD v2.0. For example, we decide whether to employ a first order SVM model based on word unigrams, or a second order SVM model based on word uni- and bigrams for MDSD v2.0's domain HEALTH.

An algorithm for model selection, viz. a *model selector* estimates the accuracies of n-th ($n \geq 1$) order SVM models for a given domain as described in Section 3.1. The model selector then chooses the SVM model that yields the highest estimated accuracy as shown in Pseudocode 3.

Pseudocode 3: Model selector for word n-gram model order.

```
input: dataset
for n = 1, 2, ..., k {
    estimate accuracy of an SVM model based on word {1,
        ..., n}-grams on dataset
}
output: n that yields the highest estimated accuracy
```

3.2.1. Evaluation

We evaluate our model selector in a leave-one-domain-out CV on MDSD v2.0's 10 domains, in which for each run we train our model selector on 9 domains and decide what word n-gram model order to employ in an SVM for the remaining 1 domain.

Data We decide between first, second and third order word n-gram models, i. e. between SVM models based on word unigrams, word uni- and bigrams, or word uni-, bi-, and trigrams. To produce data for our leave-one-domain-out CV, we evaluate 3 SVM models per domain in 10-fold CVs: one SVM model based on word unigrams, one SVM model based on word uni- and bigrams and one SVM model based word uni-, bi-, and trigrams. The evaluation results are shown in Table 1. SVM models based word uni- and bigrams always outperform SVM models based solely on word unigrams. SVM models based on word uni-, bi-, and trigrams outperform SVM models based on word uni- and bigrams only for 1 domain: MUSIC.

Experiments We vary 3 parameters of our model selector's accuracy estimation. (i) We compare 4 predictors: percentage of rare words, type/token ratio, relative entropy, and homogeneity. (ii) We compare separately kept and averaged predictors. (iii) We compare 3 LR loss functions: squared error loss, Huber loss, and Tukey's biweight. Evaluation results of our leave-one-domain-out CV are shown in Table 7.

Results Our model selector yields an average accuracy between 60–100 when deciding between first or second order. It yields an average accuracy between 40–90 when deciding between second or third order. It yields an overall accuracy between 65–95.

The most reliable model selector uses averaged homogeneity as predictor and fits the LR model using Huber loss: it yields an average accuracy of 100 when deciding between first or second order. It yields an average accuracy of 90 when deciding between second or third order. Thus, it yields an overall average accuracy of 95.

Note that for our data a naïve *baseline* also yields an overall average accuracy of 95: a naïve model selector that *always* decides for second order yields an average accuracy of 100 when deciding between first or second order. It yields an average accuracy of 90 when deciding between second or third order. Thus, its overall average accuracy is also 95.

3.3. Model Selection: Aggressive vs. Conservative Word n-gram Feature Selection

In this section we let domain complexity guide us in another model selection, viz. in deciding whether to employ aggressive or conservative word n-gram feature selection in our SA method for a given domain from MDSD v2.0. We face 2 questions when we perform word n-gram feature selection:

1. Which feature selection method should we use?
2. How many features should we select?

We answer question 1 up front: as feature selection method we use Information Gain (IG) (Yang and Pedersen, 1997), because it has been shown that IG is superior to other feature selection methods for word n-gram based text classification (Yang and Pedersen, 1997; Forman, 2003).

We answer question 2 analogously to Section 3.2. A model selector estimates—based on domain complexity—how many features to select for our SA method.

Table 8: Accuracies of SVM models based on word unigrams with (w/) and without (w/o) feature selection via IG based on the ideal CO in percent (and word unigram types).

Domain	w/o	w/	Δ	CO
APPAREL	83.25	83.6	0.35	78% (7,927)
BOOKS	79.25	80.45	1.2	11% (3,136)
DVD	78.55	80.55	2	60% (18,169)
ELECTRONICS	80.65	81.75	1.1	85% (13,139)
HEALTH	80.35	80.85	0.5	90% (11,778)
KITCHEN	81.15	82.8	1.65	17% (2,214)
MUSIC	76.4	78.45	2.05	2% (506)
SPORTS	81.7	82.55	0.85	19% (2,715)
TOYS	81.5	82.45	0.95	4% (564)
VIDEO	81.05	81.6	0.55	80% (20,301)
average	80.39	81.51	1.12	45% (8,044)

Table 9: Pearson correlation r between ideal CO and domain complexity measurements as well as r's significance level p.

Domain complexity measure	r	p
Percentage of rare word unigrams	-0.08	0.814
Word unigram type/token ratio	-0.119	0.727
Word unigram relative entropy	-0.35	0.291
Word unigram homogeneity	-0.095	0.78

3.3.1. Evaluation

As in Section 3.2.1. we evaluate our model selector—which we develop in our experiments—in a leave-one-domain-out CV.

Data Feature selection methods such as IG produce an implicit ranking with the most predictive features ranked highest and the least predictive features ranked lowest. To employ feature selection via IG we have to determine a *cut off (CO)*: features ranked above the CO are kept, while features ranked below the CO are discarded.

To produce data for our leave-one-domain-out CV for each domain we determine the CO for which our SVM model's accuracy peaks. First we rank a domain's word unigrams via IG. We then set the CO to 1, 2, ..., 100% of the domain's original word unigram vocabulary size. If it is set to 1% we keep its 1% highest ranked word unigrams, if it is set to 2% we keep its 2% highest ranked word unigrams etc. For each of the resulting 100 word unigram vocabularies we evaluate an SVM model based on this word unigram vocabulary in a 10-fold CV. We call the CO for which our SVM model's accuracy peaks *ideal CO*. Table 8 shows evaluation results of SVM models based on word unigrams with and without feature selection via IG. Feature selection is based on the ideal CO.

With feature selection using the ideal CO average accuracy is 1.12 higher than without feature selection. Ideal COs are scattered, with 85% (13,139 word unigram types) being the most conservative feature selection and 2% (506 word unigram types) being the most aggressive feature selection. The average ideal CO is 45% (8,045 word unigram types).

Experiments Table 9 correlates domain complexity measurements of MDSD v2.0's 10 domains with their ideal CO.

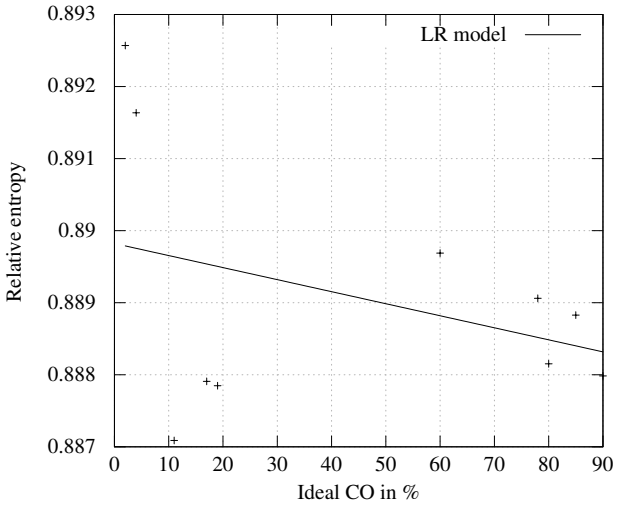

Figure 2: Ideal CO vs. relative entropy.

Table 10: Ideal COs and COs estimated by our model selector fitted using Huber loss as well as accuracies of SVM models using feature selection based on ideal and estimated CO.

Domain	Ideal		Estimated	
	CO	A	CO	A
APPAREL	78%	83.6	41%	83.50
BOOKS	11%	80.45	74%	78.25
DVD	60%	80.55	38%	80.10
ELECTRONICS	85%	81.75	42%	80.45
HEALTH	90%	80.85	46%	80.50
KITCHEN	17%	82.8	59%	82.30
MUSIC	2%	78.45	39%	77.65
SPORTS	19%	82.55	60%	81.75
TOYS	4%	82.45	37%	81.00
VIDEO	80%	81.6	47%	80.00
average	45%	81.51	48%	80.55

Relative entropy correlates strongest with ideal CO (-0.35): the smaller the domain's relative entropy, the larger its ideal CO. Hence, the less uniform a domain's word unigram distribution, the more of its word unigrams are kept as features in our SVM model. Figure 2 plots relative entropy vs. ideal CO. Additionally, it shows an LR model fitted to the data using squared error loss. It achieves no perfect fit, but it still roughly estimates ideal CO.

Given its correlation with the ideal CO, we use as our model selector a robust LR model with relative entropy as single predictor and ideal CO as response. We compare LR models fitted using Huber loss and Tukey's biweight. Table 10 shows the evaluation results of our leave-one-domain-out CV for Huber loss, Table 11 shows the evaluation results for Tukey's biweight.

Results and Discussion Our model selector over- or underestimates a domain's ideal CO on average by 40%. But SVM models with feature selection using the estimated CO outperform SVM models without feature selection in 6 out of 10 domains. SVM models with feature selection using

Table 11: Ideal COs and COs estimated by our model selector fitted using Tukey's biweight as well as accuracies of SVM models using feature selection based on ideal and estimated CO.

Domain	Ideal		Estimated	
	CO	A	CO	A
APPAREL	78%	83.6	41%	83.5
BOOKS	11%	80.45	78%	79.2
DVD	60%	80.55	37%	80.1
ELECTRONICS	85%	81.75	42%	80.45
HEALTH	90%	80.85	46%	80.5
KITCHEN	17%	82.8	62%	82.45
MUSIC	2%	78.45	39%	77.65
SPORTS	19%	82.55	62%	81.85
TOYS	4%	82.45	37%	81
VIDEO	80%	81.6	47%	80
average	45%	81.51	49%	80.67

the estimated CO yield an average accuracy of 80.55 (Huber loss) and 80.67 (Tukey's biweight). Without feature selection average accuracy is 80.39. Thus, feature selection using the estimated CO yields an average accuracy gain of 0.16 and 0.28, respectively.

Compared to SVM models with feature selection using the ideal COs (81.51), using the estimated CO performs 0.95 and 0.83 lower, respectively. SVM models with feature selection using the average ideal CO (45%) yield an average accuracy of 80.56, which is on par with SVM models with feature selection using the estimated COs.

4. Conclusions & Future Work

We investigated domain dependencies in SA. We showed that our ML-based SA method—an SVM model based on word n-grams—performs differently when applied to different domains. We also showed that domains differ in their textual characteristics, viz. their domain complexity. We then showed that there is a clear relationship between performance of our SA method in certain domains and their domain complexity. Finally, we used their relationship to (i) estimate our SA method's accuracy in certain domains based solely on its domains complexity and (ii) guide us in model selection for different domains.

In future work domain complexity may guide us in even more model selection or feature engineering tasks: whether to use super- or sub-word character n-gram representations (Raaijmakers and Kraaij, 2008) instead of word n-gram representations; whether to use non-binary word n-gram weighting, e. g. weighting using tf-idf (Manning and Schütze, 1999, p. 543); whether to employ non-lexical features, e. g. part of speech tags or dependency parses.

5. References

John Blitzer, Mark Dredze, and Fernando C.N. Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–447.

David Crystal. 2008. *A Dictionary of Linguistics and Phonetics*. Blackwell, 6th edition.

Gerard Escudero, Lluis Màrquez, and German Rigau. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Very Large Corpora (VLC)*, pages 172–180.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research (JMLR)*, 3:1289–1305.

Frank E. Harrell. 2001. *Regression Modeling Strategies: with Applications to Linear Models, Logistic Regression, and Survival Analysis*. Springer.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The elements of statistical learning*. Springer Series in Statistics. Springer, 2nd edition.

Tin Kam Ho and Mitra Basu. 2002. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300.

Paul W. Holland and Roy E. Welsch. 1977. Robust regression using iteratively reweighted least-squares. *Communications in Statistics*, 6(9):813–827.

Peter J. Huber. 1964. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101.

Adam Kilgarriff. 2001. Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.

Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

David Lee. 2001. Genres, registers, text types, domains, and styles: Clarifying the concepts and navigating a path through the bnc jungle. *Language Learning & Technology*, 5(3):37–72.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Natalia Ponomareva and Mike Thelwall. 2012. Biographies or blenders: Which resource is best for cross-domain sentiment analysis? In *Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 488–499.

Stephan Raaijmakers and Wessel Kraaij. 2008. A shallow approach to subjectivity classification. In *Proceedings of the 2nd International Conference on Weblogs and Social Media (ICWSM)*, pages 216–217.

Robert Remus and Mathias Bank. 2012. Textual characteristics of different-sized corpora. In *Proceedings of the 5th Workshop on Building and Using Comparable Corpora (BUCC)*, pages 156–160.

Robert Remus and Sven Rill. 2013. Data-driven vs. dictionary-based approaches for machine learning-based

sentiment analysis. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*, number 8105 in LNCS, pages 176–183. Springer.

Robert Remus. 2012. Domain adaptation using domain similarity- and domain complexity-based instance selection for cross-domain sentiment analysis. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012) Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction (SENTIRE)*, pages 717–723.

Satoshi Sekine. 1997. The domain dependence of parsing. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP)*, pages 96–102.

Gerard Steen. 1999. Genres of discourse and the definition of literature. *Discourse Processes*, 28(2):109–120.

Vincent Van Asch and Walter Daelemans. 2010. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing (DANLP)*, pages 31–36.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning*. Springer New York, NY.

Dong Wang and Yang Liu. 2011. A cross-corpus study of unsupervised subjectivity identification based on calibrated EM. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 161–167.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pages 412–420.