Potential and Limitations of Cross-Domain Sentiment Classification

Dirk von Grünigen
Zurich University of Applied Sciences
vongrdir@zhaw.ch

Martin Weilenmann
Zurich University of Applied Sciences
weilemar@zhaw.ch

Jan Deriu
Zurich University of Applied Sciences
deri@zhaw.ch

Mark Cieliebak
Zurich University of Applied Sciences
ciel@zhaw.ch

Abstract

In this paper we investigate the cross-domain performance of sentiment analysis systems. For this purpose we train a convolutional neural network (CNN) on data from different domains and evaluate its performance on other domains. Furthermore, we evaluate the usefulness of combining a large amount of different smaller annotated corpora to a large corpus. Our results show that more sophisticated approaches are required to train a system that works equally well on various domains.

1 Introduction

Most work regarding sentiment analysis focuses on training and testing a sentiment classifier on data of the same domain. For example a new classifier is trained on tweets and tested on tweets. However, in real-world scenarios the data might originate from different sources and domains. Often it is the case that sentiment analysis is performed on a domain for which there is no training data available. Instead of investing large amounts of money to create such a corpus it would make more sense to use an existing classifier. However, it is not always clear how well the existing classifier generalizes on the target domain. Although, it is obvious that the performance will be affected negatively, the magnitude is not known. This missing information is often useful for assessing the need of generating a new classifier for a given domain which is very costly.

Thus, our work is driven by the question of how useful sentiment classifiers are if we evaluate them with datasets from unseen domains, and if a combination of data from different domains might help to overcome the recurring problem of having too little data.

Furthermore, we assess the usefulness of large weakly supervised corpora where the labels are inferred from properties of the text, e.g. the smileys in the text or the rating of a review. We answer the question of how much gain one can expect from leveraging such corpora.

Usually, cross-domain sentiment analysis has a low performance due to the vocabulary mismatch (Pan et al., 2010). Thus, we asses the impact of word embeddings trained on large amounts of data, thus guaranteeing a large coverage of the vocabulary. We then asses how word embeddings trained on different types of data (e.g. News, Twitter) impact the performance of the system. For this, we train a convolutional neural network (CNN) based on (Deriu et al., 2016) on data from different combinations of domains and evaluate its performance on foreign domains.

Related Work Some research has been done already in the field of cross-domain sentiment classification. Most of the work in this area focuses on the mismatch in the vocabularies of the different domains.

(Pan et al., 2010) overcome the challenge of vocabulary-mismatch by employing a spectral feature alignment algorithm to map domain-specific words to a unified representation which can then be used in conjunction with the domain-independent words to lower the mismatch between the domains. (Blitzer et al., 2007) use structural correspondence learning to adapt the vocabulary of the various domains. (Li et al., 2008) experiment with ensembles of classifiers where each classifier was trained on a specific domain and then used in combination to boost the cross-domain performance. (Bollegala et al., 2011) use a semi-supervised algorithm, which leverages supervised and unsupervised data, to create a sentiment-sensitive thesaurus which is used to

Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media , pages 17–24,
Valencia, Spain, April 3-7, 2017.

compute the relatedness of words from different domains. (Bollegala et al., 2016) uses the aforementioned sentiment-sensitive thesaurus to generate sentiment-sensitive word embeddings. (Glorot et al., 2011) apply unsupervised cross-domain sentiment classification, where they use spectral embeddings to project words and documents into a low dimensional embedding space. (Yu et al., 2016) borrow ideas from SCL and combine it with auxiliary binary predicition tasks to learn dense sentence embeddings which incorporate sentiment and can be used in a cross-domain context.

Contribution Our work presents an in-depth analysis on the generalization power of the current state-of-the-art in a cross-domain setting. This work can be used to estimate and predict the expected drop in performance for a given sentiment classifier.

2 Experimental Setup

2.1 Training

Model We use a state-of-the-art model based on the CNN used by (Deriu et al., 2016). The architecture is composed by two consecutive convolutional- and pooling-layers followed by a fully-connected and a softmax layer. Table 1 gives an overview on the hyper-parameters used for the CNN.

Hyper-Parameter	Value
Number of convolutional Filters	200
Filter width (both layers)	6
Pooling Length (first layer)	4
Pooling Stride (first layer)	2
Activation	*relu*

Table 1: *Overview of the hyper-parameters chosen for the CNN. Note that we define a layer as one convolutional layer followed by one pooling-layer. For the second pooling layer the length is chosen over the whole feature.*

3-Phase Learning We apply the 3-Phase learning procedure (see Figure 1) proposed by (Severyn et al., 2015) where we first create word embeddings based on the skip-gram model (Mikolov et al., 2013). For our purposes we create embeddings with 52 dimensions as in (Deriu et al., 2016) . In a second step we apply a distant-phase where we pre-train the CNN on a large corpus of weakly supervised data, where the sentiment labels are inferred by properties of the texts. In this phase the word embeddings are updated to incorporate sentiment-specific information. The third and final phase is the supervised phase, where we train the CNN on a corpus of manually annotated texts.

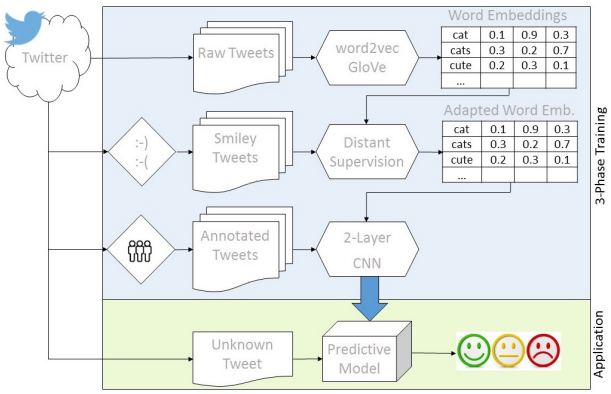

Figure 1: *Overview of the 3-Phase training procedure.*

Training For the distant-supervised and the supervised phase we employ the *AdaDelta* optimizer to train the CNN. The hyper-parameters are set to the default values of $\epsilon = 1e^{-6}$, $\rho = 0.95$, and the learning rate is set to $lr = 1.0$. Many of the datasets are unbalanced (see Table 2) and, to mitigate this problem, we use class-weights during the learning procedure. The following formula was used to compute the class-weights for each dataset D and each class $i \in S$:

$$c_i = \frac{|D|}{|S| * d_i} \tag{1}$$

where d_i denotes the number of elements in D that belong to class i. Thus, over-represented classes will get a lower weight than under-represented classes. The loss function is scaled with the class-weight for the respective class when training the model.

2.2 Data

For each of the aforementioned phases we experiment with different corpora. We use 3 different corpora for word embeddings, 2 corpora for the distant-supervised phase where the sentiment is inferred by the smiley in case of the tweets and the user ratings in case of the product reviews, and 8 corpora for the supervised phase. A detailed overview of the data is provided in Table 2.

Phase	Dataset	Total	Neutral	Neg.	Pos.	Source
Word Embeddings	Twitter	590M	-	-	-	Public Twitter-API[1]
	News	90M	-	-	-	STATMT website[2]
	Wikipedia	4.5M	-	-	-	Wikimedia[3]
Distant Phase	Reviews	82M	7M	11M	64M	(McAuley et al., 2015)
	Twitter	100M	-	20M	80M	Public Twitter-API[2]
Supervised Phase (Train)	DAI (Tweets)	3274	2191	447	636	(Narr et al., 2012)
	SEval (Tweets)	8226	3958	1210	3058	(Nakov et al., 2016)
	DIL (Reviews)	3420	1739	615	1066	(Ding et al., 2008)
	HUL (Reviews)	3156	1822	438	896	(Hu et al., 2004)
	TAC (Reviews)	2152	381	991	780	(Täckström et al., 2011)
	MPQ (News)	8888	4934	2637	1317	(Wiebe et al., 2005)
	JCR (Quotations)	1032	736	141	155	(Balahur et al., 2013)
	SEM (Headlines)	1000	610	246	144	(Strapparava et al., 2007)
Supervised Phase (Test)	DAI (Tweets)	819	556	101	162	(Narr et al., 2012)
	SEval (Tweets)	3813	1640	601	1572	(Nakov et al., 2016)
	DIL (Reviews)	855	441	144	270	(Ding et al., 2008)
	HUL (Reviews)	789	421	197	171	(Hu et al., 2004)
	TAC (Reviews)	537	65	329	143	(Täckström et al., 2011)
	MPQ (News)	2223	1225	708	290	(Wiebe et al., 2005)
	JCR (Quotations)	258	127	93	38	(Balahur et al., 2013)
	SEM (Headlines)	250	154	66	30	(Strapparava et al., 2007)

Table 2: *Data used for training the CNN model.*

[1] https://dev.twitter.com/rest/public
[2] http://www.statmt.org/wmt14/training-monolingual-news-crawl
[3] https://dumps.wikimedia.org/enwiki/latest/

Evaluation For the evaluation we use the macro-averaged F1-score of positive and negative classes F1 = ($F1_{pos}$ + $F1_{neg}$) / 2, since it is also used in SemEval (Nakov et al., 2016) as standard measure of quality.

3 Experiments & Results

In the following we refer to the system trained on a single target domain (TD) data as *specialized TD system*, a system trained on one foreign domain (FD) dataset and evaluated on the TD test set is called a *specialized FD system*, a system trained on a combinations of FD corpora is called a *generalized FD system*, and a system trained on all data is called a *generalized system*.

3.1 Word Embeddings and Distant-Phase

We train the CNN with all possible combinations of word-embeddings and distant-phases to assess which combination works best for each domain. Additionally we include experiments where we use randomly initialized word embeddings denoted as *Random*, as well as experiments where the distant-phase is omitted, denoted as *None*. Tables 4, 5, and 6 give an overview of the results. In the following we present the main findings.

The complexity among the domains varies. The differences of the averaged scores over each domain are very high. The average score of the *DAI*-tweets is 66 points in F1 score, whereas the average score of the *JCR*-quotations is only at 39.3 points. These differences could be caused by the different sized of the corpora, variations in the quality of the annotations or by the difficulty of the domains itself.

Random word embeddings are not necessarily bad. Generally it is assumed that using pre-trained word embeddings would increase the performance compared to using randomly initialized values. Indeed, the average performance of the random word embeddings (see Table 5.B) lies 3 point below the averages achieved by the *News*-embeddings. Random word embeddings yield the best score only for one domain out of eight. However a closer look at the averaged scores over the combinations of word embeddings and distant-phases (see Table 6) reveals that the combination of random word embeddings with a distant-phase on reviews achieves an average score of 59.4, which is the second-highest average score. Thus, a distant-phase can compensate the lack of pre-trained word embeddings.

Pretrained word embeddings are not necessarily good. The same analysis as above reveals a similar picture for the *Wikipedia*-embeddings. The average score achieved using the *Wikipedia*-embeddings lies 2 points below the average score achieved by the *News*-embeddings. The average scores achieved by using the *Wikipedia*-embeddings on each domain (see Table 5.B) is up to 6 points worse than the best score for the particular domain. Thus, pre-trained word embeddings do not imply an increase in score.

Vocabulary coverage is important. Table 3 shows for each domain the percentage of missing words in the corresponding word embedding. Both the News and Twitter embeddings cover most of the vocabulary. They are missing only up to 3.87% of the vocabulary most of the dataset are missing less than 1% of the vocabulary. On the other hand the Wikipedia embeddings have a much lower coverage, for all of the datasets between 15% and 30% of the vocabulary is not covered. As we have previously seen the *Wikipedia*-embeddings perform worse than the embeddings based on news and tweets. Thus, having an adequate coverage of the vocabulary is important.

	News	Twitter	Wikipedia
DAI	3.87%	3.70%	29.5%
DIL	0.98%	0.85%	21.3%
HUL	1.41%	0.85%	21.9%
JCR	0.31%	1.37%	14.5%
MPQ	0.56%	1.67%	16.9%
SEM	0.53%	1.48%	14.3%
SEval	2.38%	3.01%	26.5%
TAC	1.01%	1.26%	21.2%

Table 3: *Overview of the percentage of missing vocabulary in the word embeddings.*

Distant-Phase as score-booster. Performing a distant-phase yields the best scores for eight out of nine domains, the exception being the *MPQ*-reviews. The average scores achieved performing a distant-phase show the same picture (see Table 5.C *Avg.*-column), where using the *Review*-corpus performs 7 points above omitting the distant-phase. Using tweets for the distant-phase improves the score by 4 points on average. Thus, a distant-phase boosts the performance of the system. This is consistent with the results shown in (Deriu et al., 2016). However we cannot give any recommendation as to which corpus to use, even if using reviews mostly performed better in our case.

	None	Reviews	Twitter
Random	0.502	0.594	0.550
News	0.560	0.604	0.568
Twitter	0.539	0.594	0.585
Wikipedia	0.513	0.586	0.557

Table 6: *Shows the average F1 score for each combination of word embeddings, distant-phase corpus.*

3.2 Cross-Domain Experiments

We train the system on the data of one domain called *target domain* (TD) and test it on the TD as well as the *foreign domains* (FD). The system is optimized for the TD by using the test set of the TD to perform early-stopping. Furthermore we trained the system on the union of all domains and tested it on all the domains separately. For optimization we used the TD test set for early-stopping. For each domain we use the best combination of word embeddings and distant-phase from Section 3.1 as base model (see Table 7). In Table 8 an overview of the results is given.

	Word Emb.	Dist. Phase
DAI	Twitter	Twitter
DIL	Twitter	Reviews
HUL	Wikipedia	Reviews
JCR	Wikipedia	Reviews
MPQ	News	None
SEM	Twitter	Twitter
SEval	News	Twitter
TAC	Random	Reviews

Table 7: *Shows for each domain the best combination of word embeddings and distant phase.*

The generalization power of a specialized systems is poor. As expected the best score is achieved by training and testing on the same domain. However there is a large deterioration in score when the system is tested on another domain than it is trained on. The average score achieved by a specialized FD system on the TD is far below the scores achieved for a specialized TD system. The differences range from 15 (*JCR*) up to 30 (*DAI* and *DIL*) points in F1 score.

Embedding Type	DAI *(T/ 3.2k)*	MPQ *(N/ 8.8k)*	DIL *(R/ 3.4k)*	TAC *(R/ 2.1k)*	SEM *(H/ 3.1k)*	JCR *(Q/ 1k)*	SEval *(T/ 8.2k)*	HUL *(R/ 3.1k)*	Union
Random	0.599	0.469	**0.509**	0.577	0.436	0.263	0.598	**0.513**	0.550
News	**0.631**	**0.581**	0.485	**0.644**	0.527	**0.405**	**0.673**	0.480	**0.615**
Twitter	0.629	0.504	0.507	0.585	**0.541**	0.360	0.629	0.511	0.584
Wikipedia	0.553	0.529	0.455	0.570	*0.506*	0.375	0.613	0.451	0.567
Average	0.603	0.520	0.489	0.594	0.502	0.351	0.628	0.489	0.579

(a) No Distant Phase

Embedding Type	DAI *(T/ 3.2k)*	MPQ *(N/ 8.8k)*	DIL *(R/ 3.4k)*	TAC *(R/ 2.1k)*	SEM *(H/ 3.1k)*	JCR *(Q/ 1k)*	SEval *(T/ 8.2k)*	HUL *(R/ 3.1k)*	Union
Random	0.698	0.539	0.595	**0.714**	0.477	0.401	0.659	0.659	0.603
News	0.692	**0.563**	0.590	0.682	**0.519**	0.433	**0.685**	0.649	**0.624**
Twitter	**0.701**	0.540	**0.603**	0.694	0.472	0.383	0.683	0.657	0.611
Wikipedia	0.661	0.542	0.569	0.684	0.500	**0.457**	0.622	**0.666**	0.577
Average	0.688	0.546	0.589	0.694	0.492	0.418	0.662	0.658	0.604

(b) Review Distant Phase

Embedding Type	DAI *(T/ 3.2k)*	MPQ *(N/ 8.8k)*	DIL *(R/ 3.4k)*	TAC *(R/ 2.1k)*	SEM *(H/ 3.1k)*	JCR *(Q/ 1k)*	SEval *(T/ 8.2k)*	HUL *(R/ 3.1k)*	Union
Random	0.684	0.490	0.523	0.612	0.468	0.399	0.659	0.517	0.595
News	0.678	**0.567**	**0.546**	**0.676**	0.539	0.412	**0.691**	**0.554**	0.444
Twitter	**0.734**	0.518	0.543	0.652	**0.554**	**0.412**	0.685	0.553	**0.610**
Wikipedia	0.663	0.544	0.520	0.619	0.505	0.411	0.642	0.531	0.580
Average	0.690	0.530	0.533	0.640	0.517	0.409	0.669	0.539	0.558

(c) Twitter Distant Phase

Table 4: *Shows the score for each combination of word embeddings, distant-phase corpus, and domain. The last row shows the average score achieved on a particular dataset. The scores in bold denote the best score achieved on the dataset. For each domain we denote the text-type as follows: T: Tweets, N: News, R: Reviews, H: Headlines and Q: Quotations. Alongside with the text-type we also note the size of the corpus.*

	DAI	MPQ	DIL	TAC	SEM	JCR	SEval	HUL	Union
Full Average	0.660	0.532	0.537	0.643	0.504	0.393	0.653	0.562	0.580

(a) Shows the average scores for each dataset over each combination of word embedding type and distant phase.

Embedding Type	DAI	MPQ	DIL	TAC	SEM	JCR	SEval	HUL	Union	Avg.
Random	0.660	0.499	0.542	0.635	0.460	0.354	0.639	0.563	0.583	0.548
News	0.667	***0.570***	0.540	***0.668***	***0.528***	***0.417***	***0.683***	0.561	0.561	0.577
Twitter	***0.688***	0.521	***0.551***	0.643	0.522	0.385	0.666	***0.573***	***0.602***	0.572
Wikipedia	0.626	0.538	0.515	0.625	0.504	0.414	0.626	0.549	0.575	0.552

(b) Shows the average scores achieved for each word embedding type on each dataset. The last column shows the average score of the word embedding types.

Distant Phase Type	DAI	MPQ	DIL	TAC	SEM	JCR	SEval	HUL	Union	Avg.
None	0.603	0.520	0.489	0.594	0.502	0.351	0.628	0.489	0.579	0.528
Reviews	0.688	***0.546***	***0.589***	***0.694***	0.492	***0.418***	0.662	***0.658***	***0.604***	0.595
Twitter	***0.690***	0.530	0.533	0.640	***0.517***	0.409	***0.669***	0.539	0.558	0.565

(c) Shows the average scores achieved by each distant-phase on each data-set. The last column shows the average score achieved by the distant phase.

Table 5: *Gives an overview of the averaged scores. In Panel A the average score for each dataset is shown. Panel B shows the average scores achieve by each embedding type. Panel C shows the average scores for the distant supervised phases.*

Train \ Test	DAI *T*	MPQ *N*	DIL *R*	TAC *R*	SEM *H*	JCR *Q*	SEval *T*	HUL *R*	Union
DAI T	**0.734**	0.161	0.401	0.369	0.283	0.269	0.554	0.397	0.447
MPQ N	0.495	**0.581**	0.307	0.402	0.313	0.411	0.471	0.318	0.489
DIL R	0.381	0.210	**0.603**	0.478	0.135	0.227	0.365	0.602	0.350
TAC R	0.395	0.376	0.501	**0.714**	0.360	0.409	0.480	0.517	0.442
SEM H	0.360	0.148	0.188	0.247	**0.554**	0.054	0.250	0.181	0.227
JCR Q	0.450	0.319	0.402	0.461	0.254	**0.457**	0.402	0.452	0.384
SEval T	0.525	0.441	0.489	0.577	0.445	0.421	**0.691**	0.479	0.578
HUL R	0.404	0.252	0.567	0.535	0.176	0.312	0.392	**0.666**	0.373
Union	**0.725**	0.55	0.554	0.614	0.422	0.465	0.69	0.528	0.624
FD Avg.	0.43	0.272	0.408	0.438	0.281	0.301	0.416	0.421	0.411
Diff.	0.304	0.308	0.195	0.276	0.273	0.156	0.275	0.245	0.213

Table 8: *Results obtained by training on a target domain (TD) and evaluation on all domains. The line* FD Avg. *shows the average scores for each TD when trained on a foreign domain (FD). The line* Diff. *shows the difference between the best score of TD and FD Avg.*

A general system does not increase the systems prediction power. The results achieved by training on the union of all data and optimizing for a specific TD shows no increase in score on the TD. Only on the *JCR*-quotations the score increased, on the twitter datasets (*DAI* and *SEval*) the score is similar to the score of the target specific system. In all the other cases the systems trained on the union of all data perform worse. In the case of the *HUL*-reviews the drop is even by 14 points.

3.3 Ablation Experiments

To further assess the generalization performance we ran ablation experiments as follows: We combine all the training sets except for the target domain set, train the system on this combination of data, and then evaluate the system on the target domain.

The generalized FD system performs better than a specialized FD system. Table 9 shows the performance of the system trained on the combination of FD data excluding the TD. The results show that in most cases training on a mixture of FD data achieves better scores on the TD data than training using a single FD for training (see Table 8). As expected the general FD system is usually not able to achieve the score on the TD data achieved by the specialized TD system. Table 9 shows the difference between the specialized TD system and the generalized FD system. The differences range from 3 points in the case the *DIL*-reviews up to 17 points for the *MPQ*-news. Only for the *JCR*-quotations the generalized FD system performs better. Thus, it is best to have TD data, although in some cases an acceptable score might be achieved using a generalized FD system.

	Ablation Sys. without TD	Specific TD System	Diff.
DAI	0.658	0.734	0.076
MPQ	0.404	0.581	0.177
DIL	0.573	0.603	0.030
TAC	0.558	0.714	0.156
SEM	0.426	0.554	0.128
JCR	0.485	0.457	-0.029
SEval	0.658	0.691	0.033
HUL	0.566	0.666	0.099

Table 9: *Results of the ablation experiments. The last column shows the difference between the specific TD system and the Ablation System trained on a mix of FD data excluding data from the TD.*

3.4 Augmentation Experiments

To further investigate the difference between a specialized system and a general system we performed experiments where we start with a specialized TD, specialized FD, or a general FD system (referred to as *base system*) and gradually transform it to a generalized system by adding data. Let n be the number of texts used to train the base system. Then we augment the training set by adding $n/2$, n and $2n$ datapoints. The evaluation is always performed on the TD.

Adding FD to a specialized TD system decreases the performance on the target domain. For each of the 8 TDs we start with a specialized TD system and gradually add a combination of FD data (*mixed FD augmentation*) or data from a single FD (*single FD augmentation*) and evaluate the performance on TD. Figure 2 shows the scores averaged over all experiments for each TD. The trend shows that adding more data from one or more FDs for training decreases the performance of the system.

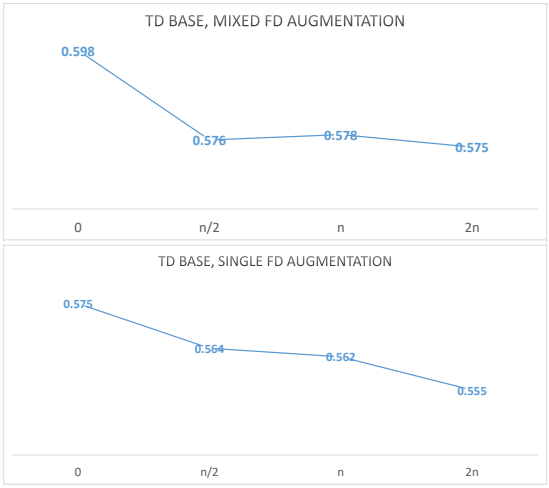

Figure 2: *Averaged F1 scores for the increasing amount of FD (mixed set or single domain set) for a TD data basis.*

Adding TD to a FD system increases the score. For each TD we start with a specialized FD system (*single FD base*) or a generalized FD system (*mix FD base*) and gradually add more data from the TD. In both cases adding more data from the TD increases the performance of the system when it is evaluated on the TD (see Figure 3).

4 Conclusion

In this work we gave an overview of the deterioration of the quality when using a sentiment classifier on a domain it was not trained on. Our in-depth analysis showed that having a large corpus of weakly labelled data boosts the score by 7 points on average. We also showed that using pre-trained word embeddings helps to increase the score by 3-4 points on average. This work can be used as a basis when evaluating sentiment classifiers that were trained on a domain different from the target domain. Future work in this area would include more indepth analysis of the inter-

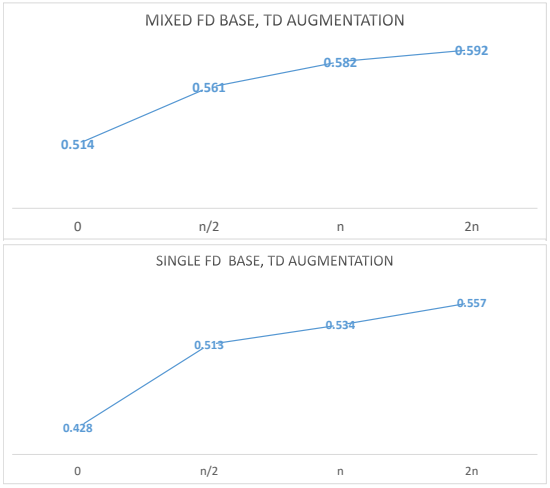

Figure 3: *Averaged F1 for increasing the amount of TD data starting with either a base of mixed FD data or single FD data.*

play among different domains: for instance our results show that a system trained on tweets performs better on reviews than a system trained on news. Here, a better understanding of these mechanisms is necessary to better assess the potential of cross domain classification. Furthermore, one can analyse the effect of the distant-phases and word embeddings in the cross-domain setting. How does the usage of different types of word embeddings and weakly labelled data impact the performance in a cross-domain setting? Does the usage of weakly-labelled data increase the performance of a sentiment classifier on a foreign domain? We are convinced that answering these questions will help to develop sentiment analysis systems that perform better on new, unknown domains.

References

Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. 2013. Sentiment analysis in the news. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 2216–2220. European Language Resources Association.

John Blitzer, Mark Dredze, and Fernanod Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 187–205. Association for Computational Linguistics.

Danushka Bollegala, David Weir, and John Carroll.

2011. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 132–141. Association for Computational Linguistics.

Danushka Bollegala, Tingting Mu, and John Yanis Goulermas. 2016. Cross-domain sentiment classification using sentiment sensitive embeddings. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):398–410.

Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. 2016. Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1124–1128. Association for Computational Linguistics.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240. Association for Computing Machinery.

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

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. Association for Computing Machinery.

Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 257–260. Association for Computational Linguistics.

Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119. Curran Associates, Inc.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016)*, pages 1–18. Association for Computational Linguistics.

Sascha Narr, Michael Hulfenhaus, and Sahin Albayrak. 2012. Language-independent twitter sentiment analysis. *Knowledge Discovery and Machine Learning (KDML), LWA*, pages 12–14.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. Association for Computing Machinery.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter Sentiment Analysis with Deep Convolutional Neural Networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. Association for Computing Machinery.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.

Oscar Täckström and Ryan McDonald. 2011. Discovering fine-grained sentiment with latent variable structured prediction models. In *In Proceedings of the 33rd European Conference on Advances in Information Retrieval*, pages 368–374. Springer.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 236–246, Austin, Texas, November. Association for Computational Linguistics.