If you've got it, flaunt it:
Making the most of fine-grained sentiment annotations

Jeremy Barnes, Lilja Øvrelid, Erik Velldal
University of Oslo
Department of Informatics
{jeremycb,liljao,erikve}@ifi.uio.no

Abstract

Fine-grained sentiment analysis attempts to extract sentiment holders, targets and polar expressions and resolve the relationship between them, but progress has been hampered by the difficulty of annotation. *Targeted* sentiment analysis, on the other hand, is a more narrow task, focusing on extracting sentiment targets and classifying their polarity.

In this paper, we explore whether incorporating holder and expression information can improve target extraction and classification and perform experiments on eight English datasets. We conclude that jointly predicting target and polarity BIO labels improves target extraction, and that augmenting the input text with gold expressions generally improves targeted polarity classification. This highlights the potential importance of annotating expressions for fine-grained sentiment datasets. At the same time, our results show that performance of current models for predicting polar expressions is poor, hampering the benefit of this information in practice.

1 Introduction

Sentiment analysis comes in many flavors, arguably the most complete of which is what is often called fine-grained sentiment analysis (Wiebe et al., 2005; Liu, 2015). This approach models the sentiment task as minimally extracting all opinion holders, targets, and expressions in a text and resolving the relationships between them. This complex task is further complicated by interactions between these elements, strong domain effects, and the subjective nature of sentiment. Take the annotated sentence in Figure 1 as an example. Knowing that the target "UMUC" is modified by the expression "5 stars" and not "don't believe" is important to correctly classifying the polarity. Additionally, the fact that this is a belief held by "some others"

as apposed to the author of the sentence can help us determine the overall polarity expressed in the sentence.

Compared to document- or sentence-level sentiment analysis, where distant labelling schemes can be used to obtain annotated data, fine-grained annotation of sentiment does not occur naturally, which means that current machine learning models are often hampered by the small size of datasets. Furthermore, fine-grained annotation is demanding, leads to relatively small datasets, and has low inter-annotator agreement (Wiebe et al., 2005; Wang et al., 2017). This begs the question: **is it worth it to annotate full fine-grained sentiment**?

Targeted sentiment (Mitchell et al., 2013; Zhang et al., 2015) is a reduction of the fine-grained sentiment task which concentrates on extracting sentiment targets and classifying their polarity, effectively ignoring sentiment holders and expressions. The benefit of this setup is that it is faster to annotate and simpler to model. But **would targeted sentiment models benefit from knowing the sentiment holders and expressions**?

In this work, we attempt to determine whether holder and expression information is useful for extracting and then classifying sentiment targets. Specifically, we ask the following research questions:

> **RQ1:** Given the time and difficulty required to annotate opinion holders, expressions, and polarity, is this information useful to **extract sentiment targets**?
>
> (a) Does augmenting the input text with holders and expressions improve target extraction?
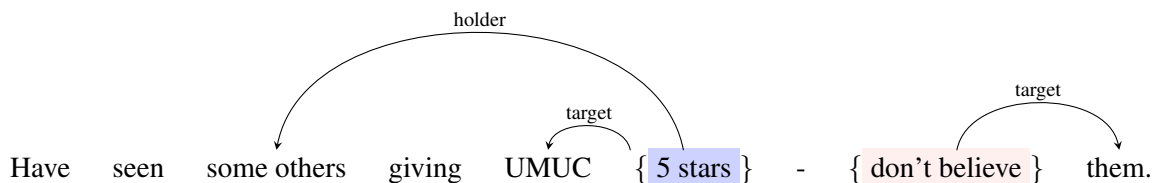> (b) Do target extraction models benefit from predicting holders and expressions?

Figure 1: An opinion annotation from the Darmstadt Review Corpus.

(c) Do target extraction models benefit from predicting the polarity of targets and/or expressions?

RQ2: Can holder and expression information improve **polarity classification** on extracted targets?

(a) Does augmenting the input text with holders and expressions improve polarity classification?

(b) Do potential benefits of augmenting the input depend on how we model the target, *i.e.,* using the [CLS] embeddings, mean pooling the target embeddings, etc.?

(c) Can sentiment lexicons provide enough information on expressions to give improvements?

We conduct a series of experiments on eight English sentiment datasets (three with full fine-grained sentiment and five targeted) with state-of-the-art models based on fine-tuned BERT models. We show that (1) it is possible to improve target extraction by also trying to predict the polarity, and that (2) classification models benefit from having access to information about sentiment expressions. We also (3) release the code[1] to reproduce the experiments, as well as the scripts to download, preprocess, and collect the datasets into a compatible JSON format, with the hope that this allows future research on the same data.

2 Related work

Fine-grained approaches to sentiment analysis attempt to discover opinions from text, where each opinion is a tuple of *(opinion holder, opinion target, opinion expression, polarity, intensity)*. Annotation of datasets for this granularity requires creating in-depth annotation guidelines, training

---

[1] https://github.com/ltgoslo/finegrained_modelling

annotators, and generally leads to lower inter-annotator scores than other sentiment tasks, *e.g.,* document- or sentence-level classification, as deciding on the spans for multiple elements and their relationships is undeniably harder than choosing a single label for a full text. **Targeted sentiment**, on the other hand, generally concentrates only on target extraction and polarity classification. This has the benefit of allowing non-experts and crowd-sourcing to perform annotation, making it easier to collect larger datasets for machine learning. This simplified annotation can be crowd-sourced, leading to larger datasets for machine learning.

2.1 Datasets

The Multi-purpose Question Answering dataset (*MPQA*) (Wiebe et al., 2005) is the first dataset that annotated opinion holders, targets, expressions and their relationships. The news wire data leads to complex opinions and a generally difficult task for sentiment models. Normally, the full opinion extraction task is modelled as extraction of the individual elements (holders, targets, and expressions) and the subsequent resolution of the relationship between them.

The *Darmstadt Review Corpora* (Toprak et al., 2010) contain annotated opinions for consumer reviews of universities and services. The authors annotate holders, targets, expressions, polarity, modifiers, and intensity. They achieve between 0.5 and 0.8 agreement using the agr method (Wiebe et al., 2005), with higher disagreement on what they call "polar targets" – targets that have a polarity but no annotated sentiment expression – holders, and expressions.

The *Open* Domain Targeted dataset (Mitchell et al., 2013) makes use of crowd sourcing to annotate NEs from scraped tweets in English and Spanish (Etter et al., 2013) with their polarities. The authors use majority voting to assign the final labels for the NEs, discarding tweets without sentiment consensus on all NEs.

The 2014 *SemEval* shared task (Pontiki et al., 2014) on aspect-based sentiment analysis include labeled data from restaurant and laptop reviews for two subtasks: 1) target extraction, which they call "aspect term extraction" and 2) classification of polarity with respect to targets ("aspect term polarity").

As most targeted datasets only contain a single target, or multiple targets with the same polarity, sentence-level classifiers are strong baselines. In order to mitigate this, Jiang et al. (2019) create a *Challenge* dataset which has both multiple targets and multiple polarities in each sentence. Similarly, Wang et al. (2017) also point out that most targeted sentiment methods perform poorly with multiple targets and propose *TDParse*, a corpus of UK election tweets with multiple targets per tweet.

2.2 Modelling

Katiyar and Cardie (2016) explore jointly extracting holders, targets, and expressions with LSTMs. They find that adding sentence-level and relation-level dependencies (IS-FROM or IS-ABOUT) improve extraction, but find that the LSTM models lag behind CRFs with rich features.

Regarding modelling the interaction between elements, there are several previous attempts to jointly learn to extract and classify targets, using factor graphs (Klinger and Cimiano, 2013), multi-task learning (He et al., 2019) or sequence tagging with collapsed tagsets representing both tasks (Li et al., 2019). In general, the benefits are small and have suggested that there is only a weak relationship between target extraction and polarity classification (Hu et al., 2019).

3 Data

One of the difficulties of working with fine-grained sentiment analysis is that there are only a few datasets (even in English) and they come in incompatible, competing data formats, *e.g.,* BRAT or various flavors of XML. With the goal of creating a simple unified format to work on fine-grained sentiment tasks, we take the eight datasets mentioned in Section 2 – *MPQA* (Wiebe et al., 2005), *Darmstadt Services and Universities* (Toprak et al., 2010), *TDParse* (Wang et al., 2017), *SemEval Restaurant and Laptop* (Pontiki et al., 2014), *Open* Domain Targeted Sentiment (Mitchell et al., 2013), and the *Challenge* dataset from Jiang et al. (2019) – and convert them to

a standard *JSON* format. The datasets are sentence and word tokenized using NLTK (Loper and Bird, 2002), except for MPQA, DS. Service and DS. Uni, which already contain sentence and token spans. All polarity annotations are mapped to positive, negative, neutral, and conflict[2]. As such, each sentence contains a sentence id, the tokenized text, and a possibly empty set of opinions which contain a holder, target, expression, polarity, and intensity. We allow for empty holders and expressions in order generalize to the targeted corpora. Finally, we use 10 percent of the training data as development and another 10 percent for test for the corpora that do not contain a suggested train/dev/test split. For training and testing models, however, we convert the datasets to CoNLL format.

Table 1 presents an overview of the different datasets and highlights important differences between them. The fully fine-grained sentiment datasets (MPQA, DS. Services, and DS. Uni) tend to be larger but have fewer targets annotated, due to a larger number of sentences with no targets. However, the MPQA dataset contains much longer targets than the other datasets – an average of 6, but a maximum of 56 tokens. It also contains more opinion holders and expressions and these also tend to be longer, all of which marks MPQA as an outlier among the datasets. The distribution of polarity is also highly dependent on the dataset, with DS. Services being the most skewed and SemEval Laptop the least skewed. Finally, the challenge dataset is by far the largest with over 11,000 training targets. Additionally, Table 6 in Appendix A shows the percentage of unique targets per dataset, as well as the percentage of targets shared between the training set and the dev and test sets. Again MPQA has the largest number of unique targets and the least overlap.[3]

4 Experimental Setup

We split the task of targeted sentiment analysis into the **extraction of sentiment targets** and subsequent **polarity classification** of extracted targets, given their context. Figure 2 shows the two tasks and the eight models used in the experiments. As a base model, we take the tar-

---

[2]We discard conflict during evaluation because there are not enough examples to properly learn this class in most datasets

[3]We do not, however, consider partial overlap which may exaggerate the true uniqueness of targets.

		domain		sentences		holders			targets			expressions			polarity		
				#	avg.	#	avg.	max	#	avg.	max	#	avg.	max	+	neu	−
Fine-grained Sentiment	MPQA	newswire	train	4500	25	1306	2.6	27	1382	6.1	56	1656	2.4	14	675	271	658
			dev	1622	23	377	2.6	16	449	5.3	41	552	2.1	8	241	105	202
			test	1681	24	371	2.8	32	405	6.4	42	479	2.0	8	166	89	199
	DS. Services	service reviews	train	5913	16	18	1.2	2	2504	1.2	7	1273	1.2	10	1623	46	838
			dev	744	18	1	1.7	3	288	1.2	4	144	1.4	5	103	1	104
			test	748	17	2	1	1	328	1.2	5	168	1.4	6	241	7	80
	DS. Uni	university reviews	train	2253	20	65	1.2	2	1252	1.2	5	837	1.9	9	495	149	610
			dev	232	9	17	1.1	3	151	1.2	3	106	1.7	6	40	19	92
			test	318	20	12	1.3	4	198	1.2	6	139	2.0	5	77	18	103
Targeted Sentiment	TDParse	political tweets	train	2889	6.9	-	-	-	9088	1.2	7	-	-	-	1238	3931	3919
			dev	321	6.6	-	-	-	1040	1.2	5	-	-	-	128	454	458
			test	867	6.9	-	-	-	2746	1.2	6	-	-	-	378	1162	1206
	SemEval R.	restaurant reviews	train	2740	13	-	-	-	3293	1.4	19	-	-	-	1902	574	734
			dev	304	11.3	-	-	-	350	1.4	5	-	-	-	226	54	63
			test	800	9.6	-	-	-	1128	1.4	8	-	-	-	724	195	195
	SemEval L.	laptop reviews	train	2744	22.5	-	-	-	2049	1.5	6	-	-	-	870	402	747
			dev	304	21.1	-	-	-	244	1.6	5	-	-	-	99	44	96
			test	800	18.6	-	-	-	633	1.6	7	-	-	-	327	162	128
	Open	tweets	train	1903	12.8	-	-	-	2594	1.6	8	-	-	-	578	1801	215
			dev	211	12.3	-	-	-	291	1.6	6	-	-	-	46	220	25
			test	234	11.6	-	-	-	337	1.6	7	-	-	-	74	232	31
	Challenge	restaurant reviews	train	4297	8.8	-	-	-	11186	1.3	9	-	-	-	3380	5042	2764
			dev	500	8.9	-	-	-	1332	1.3	8	-	-	-	403	604	325
			test	500	8.9	-	-	-	1336	1.3	8	-	-	-	400	607	329

Table 1: Statistics of the datasets, including number of sentences, as well as average, and max lengths (in tokens) for holder, target, and expression annotations. Additionally, we include the distribution of polarity – restricted to positive, neutral, and negative – in each dataset.

get extraction and classification models from Xu et al. (2019), which achieve state-of-the-art performance on the SemEval task. The approach first fine-tunes BERT (Devlin et al., 2019) on domain-specific unlabeled data as a domain-adaptation step. We use the datasets themselves to perform this step, except for the SemEval datasets. For these, we follow Rietzler et al. (2020) and instead use larger amounts of unlabeled data – 1,710,553 and 2,000,000 sentences for SemEval Laptop and Restaurant respectively – taken from Amazon Laptop reviews (He and McAuley, 2016) and the Yelp Dataset Challenge.[4] We further deviate from Xu et al. (2019) by not pretraining the models on the SQUAD question answering dataset and in-domain sentiment questions which they create, as this data is not publicly available. Finally, a linear prediction is added after the BERT model and the full model is updated on the sentiment task.

For **target extraction**, we use the contextualized BERT embeddings as input to a softmax layer

and predict the sequence of tags. We compare three prediction strategies:

1. **TARG.:** The model predicts the labels $y \in \{B, I, O\}$ for the targets only.

2. **PRED.:** We additionally predict the labels for holders and expressions and predict $y \in \{B\text{-}holder, I\text{-}holder, B\text{-}target, I\text{-}target, B\text{-}expression, I\text{-}expression, O\}$.

3. **+POL.:** Finally, we add the polarity (*positive*, *negative*, *neutral*) to the annotation specific BIO-tag, which leads to an inventory of 19 labels for the full fine-grained setup and 7 for the targeted setup.

For **polarity classification**, we take as a baseline the classification architecture from Xu et al. (2019), which makes use of the two-sentence training procedure for BERT, by prepending the target before the sentence separation token, and then adding the full sentence after. We compare five strategies for producing the input to the softmax layer for predicting the sentiment of the target:

1. **[CLS]:** this model uses the [CLS] embedding from the final BERT layer.
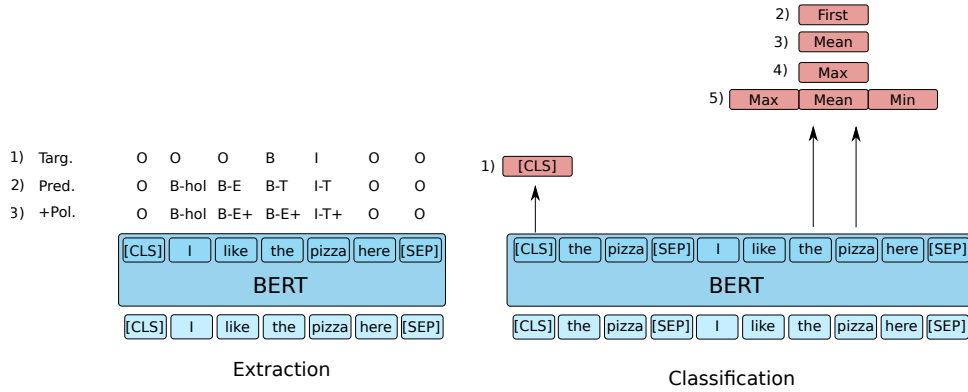
Figure 2: Our BERT-based *target extraction* and *classification models*, with the three strategies for extraction ((1) predict only targets, (2) predict holders, targets and expressions, and (3) predict the polarity of the targets and expressions as well) and five strategies for sentiment classification (passing to the softmax layer the contextualized embedding from (1) the [CLS] embedding, (2) the first token in the target (3) averaging all embeddings in the target phrase, (4) taking the max of the target embeddings, (5) concatenating the max, mean, and min).

2. **FIRST:** uses the contextualized BERT embedding from the first token of the target in context.

3. **MEAN:** instead takes the average of the BERT embeddings for the tokens in the target.

4. **MAX:** uses the max of the contextualized BERT embeddings for the tokens in the target.

5. **MAXMM:** takes the max, min, and mean pooled representations and passes the concatenation to the softmax layer, which has shown to perform well for sentiment tasks (Tang et al., 2014). However, this triples the size of the input representation to the softmax layer.

The TARG. and [CLS] models correspond to the models used in Xu et al. (2019) and serve as baselines. The extraction and classification models are fine-tuned for 50 epochs using Adam with an initial learning rate of $3e-5$, with a linear warmup of 0.1 and all other hyperparameters are left at default BERT settings (further details in Appendix B). The best model on the development set is used for testing. Combined with the four input manipulations (Table 2), this leads to eleven extraction experiments – TARG. and PRED. on the original data which only has annotated targets are the same and for simplicity we only show the results from TARG.– and twenty classification experiments per dataset. In order to control for the effect of random initialization, we run each experiment 5 times on different random seeds and report the mean and standard deviation.

4.1 Training with gold annotations

Given that we are interested in knowing whether it is beneficial to include information about additional annotations (holder, expressions, polarity), we perform experiments where we systematically include these. We do so by adding special tags, *e.g.,*, $\left[<\text{E}\right]$, into the input text surrounding the annotated spans, as shown in Table 2. The models then have access to this information both during training and at test time, albeit in an indirect way. For the first set of experiments, we perform controlled experiments under ideal conditions, *i.e.,* having gold annotations during testing. This allows us to isolate the effects of incorporating the additional annotations, without worrying about noisy predictions

4.2 Training with predicted expressions

It is equally important to know whether the models are able to use noisy predicted annotations. In order to test this, we train *expression prediction models* on the three full fine-grained sentiment corpora. We use the same BERT-based model and hyperparameters from the target extraction models above and train five models with different random seeds. Preliminary results suggested that these models had high precision, but low recall. Therefore, we take a simple ensemble of the five trained models, where for each token, we keep labels predicted by at least one of the expression models in order to increase recall.

We perform an additional set of experiments where we use sentiment lexicons and assume any word in these lexicons is a sentiment expres-

53

original	Money Magazine rated E-Trade highly .
+ holders	[<H] Money Magazine [H>] rated E-Trade highly
+ expressions	Money Magazine [<E] rated [E>] E-Trade [<E] highly [E>]
+ full	[<H] Money Magazine [H>] [<E] rated [E>] E-Trade [<E] highly [E>]

Table 2: We inform our models regarding annotations other than targets by inserting special tags into the input text before and after annotated holders and expressions .

sion. We use the **Hu and Liu** lexicon (Hu and Liu, 2004), the **SoCal** and **SoCal-Google** lexicons (Taboada et al., 2006) and the **NRC** emotion lexicon (Mohammad and Turney, 2013), which also contains sentiment annotations. The lexicons contain 6,789, 5,824, 2,142, and 5,474 entries, respectively. The MPQA and Darmstadt experiments show the effect of predicted vs. gold expressions, as well as domain transfer. The experiments on the targeted datasets, on the other hand, will show us whether it is possible to improve the targeted models with predicted expressions.

5 Results

In this section we describe the main results from the extraction and two classification experiments described in Section 4.

5.1 Target extraction

Table 3 shows the results for the extraction experiment, where token-level F_1 is measured only on targets. The models perform poorer than the state-of-the-art, as we did not finetune on the SQUAD question answering dataset and in-domain sentiment questions or perform extensive hyperparameter tuning. The average F_1 score depends highly on the dataset – MPQA is the most difficult dataset with 13.1 F_1 on the original data, while the Darmstadt Universities corpus is the easiest for target extraction with 84.6. Augmenting the input text with further annotations, but predicting only sentiment targets (TARG. in Table 3) hurts the model performance in all cases. Specifically, adding holder tags leads to an average drop of 1.3 percentage points (pp), expressions 1.2 and full 1.5. Attempting to additionally predict these annotations (PRED. in Table 3) leads to mixed results – the model leads to improvements on MPQA + exp. and Darmstadt Services + holders, no notable difference on MPQA + full and Darmstadt Universities + exp., and a loss on the rest.

Adding the polarity to the target BIO tags (original +POL. in Table 3) leads to the most consistent improvements across experiments – an average of 0.5 pp – with the largest improvement of 1.5 pp on the TDParse dataset. This suggests a weak-to-moderate relationship between polarity and extraction, which contradicts previous conclusions (Hu et al., 2019). Finally, further adding the holder and expression tags (+POL. in Table 3) tends to decrease performance.

5.2 Polarity classification with gold annotations

Table 4 shows the macro F_1 scores for the polarity classification task on the gold targets. The model performs better than the best reported results on Challenge (Jiang et al., 2019), and similar to previous results on the SemEval corpora. Regarding the choice of target representation, FIRST is the strongest overall, with an average of 64.7 F_1 across the original eight datasets, followed by MAX (64.6), MEAN (64.4), MAXMM (64.2), and finally [CLS] (64.1). It is, however, unclear exactly which representation is the best, as it differs for each dataset. But we can conclude that [CLS] is in general the weakest model, while either FIRST or MAX provide good starting points.

Adding holder annotations to the input text delivers only small improvements on four of the fifteen experiments, and has losses on seven. The +exp. model, however, leads to significant improvements on 10 experiments. The outlier seems to be Darmstadt Services, which contains a large number of "polar targets" in the data, which do not have polar expressions. This may explain why including this information has less effect on this dataset. Finally, +full performs between the original input and +exp.

	MPQA	DS. Services	DS. Unis	Challenge	SemEval R.	SemEval L.	Open	TDParse
Xu et al. (2019)	n/a	n/a	n/a	n/a	78.0	84.3	n/a	n/a
BiLSTM-CRF	12.2 (1)	85.0 (1)	84.4 (1)	73.4 (1)	72.5 (1)	74.0 (1)	62.2 (1)	82.6
original	14.1 (2)	85.9 (1)	**84.6 (0)**	75.8 (1)	51.9 (1)	71.3 (1)	62.0 (4)	81.7 (3)
TARG. + holders	11.9 (1)	84.3 (1)	83.6 (1)	-	-	-	-	-
TARG. + exp.	11.6 (1)	85.0 (0)	83.4 (0)	-	-	-	-	-
TARG. + full	10.5 (2)	84.8 (1)	83.8 (1)	-	-	-	-	-
PRED. + holders	12.1 (2)	**86.2 (0)**	84.6 (0)	-	-	-	-	-
PRED. + exp.	**14.9 (1)**	84.7 (1)	84.5 (1)	-	-	-	-	-
PRED. + full	13.0 (3)	85.5 (1)	84.3 (1)	-	-	-	-	-
+POL. BiLSTM-CRF	13.9 (1)	85.2 (1)	83.7 (1)	73.6 (1)	**73.7 (1)**	**74.5 (1)**	62.3 (1)	81.8 (1)
+POL. original	13.8 (1)	85.4 (1)	84.3 (1)	**76.9 (1)**	52.5 (1)	**71.6 (1)**	**62.9 (1)**	**83.2 (0)**
+POL. + holders	13.8 (2)	85.6 (1)	84.4 (1)	-	-	-	-	-
+POL. + exp.	13.5 (2)	85.4 (1)	84.3 (0)	-	-	-	-	-
+POL. + full	12.0 (1)	86.0 (1)	**84.6 (0)**	-	-	-	-	-

Table 3: Average token-level F1 scores for the target extraction task across five runs, (standard deviation in parenthesis). **Bold** numbers indicate the best model per dataset, while blue and pink highlighting indicates an improvement or loss in performance compared to the original data, respectively.

5.3 Polarity classification with predicted annotations

The expression models achieve modest F_1 scores when trained and tested on the same dataset – between 15.0 and 47.9 –, and poor scores when transferred to a different dataset – between 0.9 and 14.9 (further details shown in Table 7 in Appendix A). The lexicons often provide better cross-dataset F_1 than the expression models trained on another dataset, as they have relatively good precision on general sentiment terms.

Figure 3 shows a heatmap of improvements (blue) and losses (red) on the eight datasets (x-axis) when augmenting the input text with expression tags from the expression models and lexicons (y-axis). We compare the expression augmented results to the original results for each pooling technique and take the average of these improvements and losses. For a full table of all results, see Table 5 in Appendix A.

Augmenting the input text with predicted sentiment expressions leads to losses in 41 out of averaged 56 experiments shown in Figure 3 (or in 173 out of 280 experiments in Table 5). Curiously, the experiments that use an expression model trained on the same dataset as the classification task, *e.g.,* MPQA predicted expressions on the MPQA classification task, have the largest losses – the largest of which is MPQA (-2.78 on average). This seems to indicate that the mismatch be-

tween the train prediction, which are near perfect, and the rather poor test predictions is more problematic than cross-dataset predictions, which are similar on train and test.

The best expression prediction model is the one trained on MPQA, improving the performance on Darmstadt Universties, Open, and SemEval Restaurants. This is likely due to the fact that MPQA has the largest number of annotated expressions, and that the domain is more general, leading to expression predictions that generalize better. The expression models trained on Darmstadt Services leads to small benefits on two corpora and the expression model trained on Darmstadt Universities only leads to losses

The datasets that receive the most benefit from expression annotations are Darmstadt Universities (6/7 experiments) and the TDParse dataset (5/7). In both cases, the lexicon-based expression models provide more consistent benefits than the trained expression prediction models. The fact that the dataset that benefits most is the TDParse dataset suggests that expression information is most useful when there are multiple targets with multiple polarities.

There is no significant correlation between the performance of the expression prediction model and the performance on the classification task on the three fine-grained datasets. In fact, there is a small but insignificant negative correlation (-0.33

		MPQA	DS. Services	DS. Unis	Challenge	SemEval R.	SemEval L.	Open	TDParse
	Previous Results	n/a	n/a	n/a	70.3	80.1	78.3		
[CLS]	original	63.5 (2)	57.3 (1)	57.6 (4)	84.3 (0)	74.1 (2)	72.8 (1)	54.6 (1)	48.8 (1)
	+ holders	63.1 (2)	57.1 (1)	60.5 (0)	-	-	-	-	-
	+ exp.	64.0 (3)	56.4 (0)	62.9 (4)	-	-	-	-	-
	+ full	61.9 (2)	56.6 (1)	62.8 (2)	-	-	-	-	-
FIRST	original	64.3 (2)	57.8 (1)	58.7 (4)	**84.4 (1)**	75.6 (1)	74.3 (1)	55.6 (2)	46.6 (1)
	+ holders	63.4 (2)	57.7 (2)	60.5 (3)	-	-	-	-	-
	+ exp.	**64.8 (2)**	57.0 (1)	63.7 (2)	-	-	-	-	-
	+ full	64.0 (1)	55.2 (1)	**65.7 (4)**	-	-	-	-	-
MEAN	original	63.5 (2)	57.3 (1)	60.2 (4)	**84.4 (1)**	74.1 (2)	72.8 (1)	56.8 (3)	46.1 (1)
	+ holders	63.1 (2)	57.8 (1)	56.7 (5)	-	-	-	-	-
	+ exp.	64.3 (2)	56.2 (1)	64.1 (3)	-	-	-	-	-
	+ full	64.2 (2)	56.3 (1)	63.7 (2)	-	-	-	-	-
MAX	original	60.8 (4)	58.2 (1)	57.8 (3)	81.4 (1)	73.9 (2)	**74.5 (2)**	**61.4 (5)**	**49.0 (3)**
	+ holders	61.9 (4)	57.9 (1)	53.9 (1)	-	-	-	-	-
	+ exp.	64.3 (2)	57.4 (1)	61.5 (6)	-	-	-	-	-
	+ full	62.7 (3)	57.9 (1)	54.5 (2)	-	-	-	-	-
MAXMM	original	59.3 (2)	57.8 (1)	55.2 (3)	81.3 (1)	**77.2 (1)**	**74.5 (1)**	60.2 (5)	48.5 (5)
	+ holders	61.3 (1)	57.8 (1)	54.7 (3)	-	-	-	-	-
	+ exp.	64.1 (2)	**59.8 (3)**	54.0 (2)	-	-	-	-	-
	+ full	63.9 (1)	57.7 (1)	54.4 (4)	-	-	-	-	-

Table 4: Average macro F_1 scores for polarity classification across five runs (standard deviation in parenthesis) on gold targets, also adding information about holders and expressions. **Bold** indicates the best model per dataset, while blue and pink highlighting indicates an improvement or loss in performance compared to the original (targets only) data, respectively.

p=0.13, -0.16 p=0.48, -0.26 p=0.25 for macro Precision, Recall, or F_1 respectively, as measured by Pearson's correlation between the expression performances and the F_1 of the classification models augmented with these predicted expressions). It seems that the possible benefits depends more on the target dataset than the actual expression model used.

6 Conclusion

In this work we have explored the benefit of augmenting targeted sentiment models with holder and sentiment expressions. The experiments have shown that although augmenting text with holder and expression tags (**RQ1 a**) or simultaneously predicting them (**RQ1 b**) have no benefit for target extraction, predicting collapsed BIO + polarity tags consistently improves target extraction (**RQ1 c**). Furthermore, augmenting the input text with gold expressions generally improves targeted polarity classification (**RQ2 a**), although it is not clear which target representation strategy is best (**RQ2 b**). Furthermore, we have found benefits of

including lexicon-based expressions for the more complex targeted datasets (**RQ2 c**).

The rather poor performance of the learned expression models and the difference between augmenting with gold or predicted expressions reveals the need to improve expression prediction approaches, both by creating larger corpora annotated with sentiment expressions, as well as performing further research on the modeling aspect. Any future work interested in modelling more complex sentiment phenomena should therefore be aware that we may first require more high-quality annotated data if we wish to do so with current state-of-the-art machine learning approaches.

Furthermore, we introduce a common format for eight standard English datasets in fine-grained sentiment analysis and release the scripts to download and preprocess them easily. We plan to include further datasets in our script in the future, as well as extending our work to other languages with available fine-grained corpora.
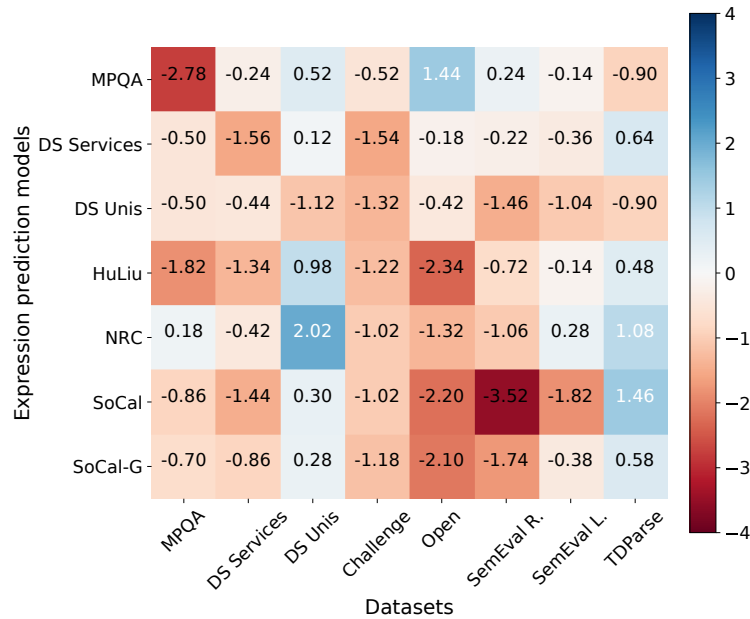
Figure 3: Heatmap of average improvements (blue) and losses (red) on the target classification tasks (x-axis) when augmenting the input text with predicted sentiment expressions from the expression prediction models (y-axis).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

David Etter, Francis Ferraro, Ryan Cotterell, Olivia Buzek, and Benjamin Van Durme. 2013. Nerit: Named Entity Recognition for Informal Text. Technical Report 11, Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, Maryland.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515, Florence, Italy. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 537–546, Florence, Italy. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, USA.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6279–6284, Hong Kong, China. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2016. Investigating LSTMs for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929, Berlin, Germany. Association for Computational Linguistics.

Roman Klinger and Philipp Cimiano. 2013. Bi-directional inter-dependencies of subjective expressions and targets and their value for a joint model. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 848–854, Sofia, Bulgaria. Association for Computational Linguistics.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019)*, pages 6714–6721, Honolulu, Hawaii. AAAI Press.

Bing Liu. 2015. *Sentiment analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, Cambridge, United Kingdom.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, pages 63–70.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654, Seattle, Washington, USA. Association for Computational Linguistics.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a Word-Emotion Association Lexicon. *Computational Intelligence*, 29(3):436–465.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2020. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4933–4941, Marseille, France. European Language Resources Association.

Maite Taboada, Caroline Anthony, and Kimberly Voll. 2006. Methods for Creating Semantic Orientation Dictionaries. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, Baltimore, Maryland. Association for Computational Linguistics.

Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. 2010. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 575–584, Uppsala, Sweden. Association for Computational Linguistics.

Bo Wang, Maria Liakata, Arkaitz Zubiaga, and Rob Procter. 2017. TDParse: Multi-target-specific sentiment recognition on twitter. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 483–493, Valencia, Spain. Association for Computational Linguistics.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2/3):164–210.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 612–621, Lisbon, Portugal. Association for Computational Linguistics.

Appendix

Appendix A   Additional tables

		MPQA	DS. Services	DS. Unis	Challenge	Open	SemEval R.	SemEval L.	TDParse
original		[Cls]	63.5 (2)	57.3 (1)	57.6 (4)	84.3 (0)	54.6 (1)	74.1 (2)	72.8 (1)	48.8 (1)
		First	**64.3 (2)**	57.8 (1)	58.7 (4)	**84.4 (1)**	55.6 (2)	75.6 (1)	74.3 (1)	46.6 (1)
		Mean	63.5 (2)	57.3 (1)	60.2 (4)	**84.4 (1)**	56.8 (3)	74.1 (2)	72.8 (1)	46.1 (1)
		Max	60.8 (4)	**58.2 (1)**	57.8 (3)	81.4 (1)	61.4 (5)	73.9 (2)	74.5 (2)	49.0 (3)
		MaxMM	59.3 (2)	57.8 (1)	55.2 (3)	81.3 (1)	60.2 (5)	**77.2 (1)**	74.5 (1)	48.5 (5)
Predicted Expressions	MPQA	[Cls]	60.3 (2)	57.0 (1)	61.3 (5)	83.1 (1)	57.5 (4)	74.2 (2)	72.2 (1)	47.5 (2)
		First	61.6 (2)	57.0 (2)	59.8 (3)	83.5 (1)	55.2 (3)	77.1 (1)	73.9 (2)	45.2 (1)
		Mean	60.3 (2)	57.0 (1)	61.3 (5)	83.1 (1)	57.5 (4)	74.2 (2)	72.2 (1)	47.5 (2)
		Max	59.1 (2)	58.1 (0)	57.0 (4)	82.3 (0)	**63.7 (1)**	75.0 (2)	74.7 (1)	48.5 (2)
		MaxMM	56.2 (5)	58.1 (1)	52.7 (2)	81.3 (1)	61.9 (3)	75.6 (2)	75.2 (1)	45.8 (4)
	DS. Services	[Cls]	63.6 (1)	56.3 (1)	60.6 (1)	82.4 (1)	53.4 (4)	72.1 (2)	72.2 (1)	49.4 (2)
		First	61.3 (2)	54.5 (0)	59.4 (3)	82.6 (1)	56.2 (9)	76.3 (1)	74.8 (1)	45.4 (2)
		Mean	64.1 (2)	56.0 (0)	60.1 (5)	82.4 (1)	56.1 (3)	73.0 (2)	72.4 (1)	49.8 (2)
		Max	61.4 (1)	56.7 (1)	55.2 (2)	80.3 (1)	63.0 (2)	76.5 (1)	74.5 (1)	48.1 (3)
		MaxMM	58.5 (2)	57.1 (1)	54.8 (5)	80.4 (1)	59.0 (2)	75.9 (2)	73.2 (2)	49.5 (4)
	DS. Unis	[Cls]	63.1 (1)	57.0 (1)	60.3 (3)	82.8 (1)	54.7 (2)	73.6 (3)	72.8 (1)	48.1 (3)
		First	64.1 (2)	56.9 (1)	58.2 (2)	82.6 (1)	55.2 (3)	70.8 (3)	72.9 (2)	44.4 (1)
		Mean	62.3 (1)	57.0 (1)	59.7 (3)	82.8 (1)	54.7 (2)	73.6 (3)	72.8 (1)	48.1 (3)
		Max	59.6 (4)	57.8 (1)	53.4 (1)	80.5 (1)	62.0 (1)	74.8 (2)	71.0 (1)	46.9 (2)
		MaxMM	59.8 (3)	57.5 (1)	52.3 (4)	80.5 (0)	59.9 (4)	74.8 (1)	74.2 (1)	47.0 (4)
Lexicon Expressions	HuLiu	[Cls]	60.3 (2)	56.2 (1)	60.8 (3)	82.8 (1)	54.0 (3)	73.6 (1)	73.2 (1)	**50.6 (1)**
		First	61.2 (2)	55.0 (1)	61.2 (1)	82.3 (1)	46.5 (3)	74.8 (1)	74.2 (1)	43.8 (1)
		Mean	60.3 (2)	56.2 (1)	60.8 (3)	82.8 (1)	54.0 (3)	73.6 (1)	73.2 (1)	**50.6 (1)**
		Max	59.7 (3)	57.2 (1)	56.4 (2)	81.0 (1)	61.1 (2)	75.5 (2)	73.7 (3)	47.2 (3)
		MaxMM	60.8 (3)	57.1 (1)	55.2 (2)	80.8 (1)	61.3 (3)	73.8 (3)	73.9 (1)	49.2 (4)
	NRC	[Cls]	64.0 (2)	56.9 (1)	63.0 (2)	83.1 (1)	54.8 (3)	72.0 (1)	73.0 (1)	49.4 (1)
		First	63.7 (2)	56.9 (1)	61.1 (2)	83.3 (1)	49.1 (5)	74.9 (3)	**74.9 (0)**	46.1 (2)
		Mean	64.0 (2)	56.9 (1)	**63.0 (2)**	83.1 (1)	54.8 (3)	72.0 (1)	73.0 (1)	49.4 (1)
		Max	61.1 (3)	58.0 (1)	55.6 (2)	80.4 (1)	62.0 (1)	75.3 (2)	74.6 (2)	49.7 (3)
		MaxMM	59.5 (3)	57.6 (1)	56.9 (4)	80.8 (1)	61.3 (2)	75.4 (2)	74.8 (1)	49.8 (4)
	SoCal	[Cls]	63.2 (2)	56.6 (1)	60.5 (4)	83.0 (0)	51.5 (4)	69.8 (1)	71.0 (1)	50.0 (1)
		First	61.8 (2)	53.7 (2)	59.9 (4)	81.8 (0)	51.4 (6)	72.8 (1)	73.0 (1)	45.4 (1)
		Mean	63.2 (2)	56.6 (1)	60.5 (4)	83.0 (0)	51.5 (4)	69.8 (1)	71.0 (1)	50.0 (1)
		Max	59.2 (2)	57.8 (1)	54.5 (2)	79.3 (1)	62.3 (2)	71.5 (4)	71.8 (2)	49.5 (4)
		MaxMM	59.7 (2)	56.5 (2)	55.6 (1)	79.1 (1)	60.9 (3)	73.4 (3)	73.0 (1)	51.4 (2)
	SoCal-Google	[Cls]	62.6 (3)	56.5 (1)	60.0 (3)	83.0 (1)	53.2 (3)	71.5 (1)	72.9 (1)	50.2 (1)
		First	62.1 (1)	56.2 (1)	60.8 (5)	82.5 (1)	49.7 (6)	74.9 (2)	74.0 (1)	46.2 (0)
		Mean	62.6 (3)	56.5 (1)	60.0 (3)	83.0 (1)	53.2 (3)	71.5 (1)	72.9 (1)	50.2 (1)
		Max	60.0 (3)	57.8 (0)	55.5 (5)	80.9 (1)	61.5 (3)	74.8 (2)	74.5 (2)	49.5 (4)
		MaxMM	60.6 (4)	57.1 (1)	54.6 (3)	80.5 (1)	60.5 (3)	73.5 (4)	72.7 (2)	45.8 (5)

Table 5: Macro F_1 scores for polarity classification of gold targets. **Bold** numbers indicate the best model per dataset, while blue and pink highlighting indicates an improvement or loss in performance compared to the original data (gold targets only), respectively.

	% Unique			% Overlap	
	train	dev	test	train-dev	train-test
MPQA	**85.7**	88.5	**89.2**	<u>15</u>	<u>19</u>
DS. Services	36.2	48.6	47.5	45.0	35.6
DS. Uni	35.2	52.9	45.0	**58.5**	47.6
TDParse	33	51.8	41.7	57.4	47.3
SemEval R.	36.3	59.8	49.4	56.4	33.8
SemEval L.	45.5	71.7	64.8	48.9	33.7
Open	85	**92.4**	87.1	23	24
Challenge	<u>23.1</u>	<u>39.0</u>	<u>39.7</u>	54.1	**52**

Table 6: Analysis of targets in the datasets. % Unique describes the number of targets that are found only in that split. % Overlap describes the percentage of dev/test targets that are found in the train set. We disregard partial matches, *e.g.,* "chinese food" and "food".

		MPQA	DS. Services	DS. Unis
trained	MPQA	15.0 (1.7)	1.0 (0.8)	2.2 (1.2)
	DS. Services	0.9 (0.3)	47.9 (7.3)	14.9 (1.2)
	DS. Unis	1.4 (0.6)	10.9 (1.5)	18.5 (1.5)
lexicons	HuLiu	4.7	17.9	16.0
	NRC	3.3	7.4	9.0
	SoCal	2.4	13.2	13.8
	SoCal Google	1.0	13.2	11.4

Table 7: Token-level macro F_1 scores for expression prediction models (trained) and lexicon expressions (lexicons) when tested on the three fine-grained datasets (x-axis). The trained model scores are the average and standard deviation across five runs with different random seeds. The lexicon models are deterministic and therefore only have a single score.

Appendix B Training details

GPU Infrastructure	1 NVIDIA P100, 16 GiB RAM
CPU Infrastructure	Intel Xeon-Gold 6126 2.6 GHz
Number of search trials	50
Domain training duration	2580 sec
Extraction fine-tuning duration	15381 sec
Classification fine-tuning duration	9080 sec
Model implementation	https://github.com/blinded/for/review

Hyperparameter	Assignment
number of epochs	50
max. sequence length	128
metric early stopping monitored	validation loss
batch size	32
sentiment dropout	0.3
learning rate optimiser	Bert Adam
fine-tuning learning rate	3e-5
learning rate warmup proportion	0.1
regularisation type	L2
regularisation value	0.01