Natural Language Engineering (2021), **27**, pp. 249–269
doi:10.1017/S1351324920000510

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Improving sentiment analysis with multi-task learning of negation

Jeremy Barnes*, Erik Velldal, and Lilja Øvrelid

Language Technology Group, University of Oslo, Oslo, Norway
*Corresponding author. E-mail: jeremycb@ifi.uio.no

(Received 22 May 2019; revised 28 September 2019; accepted 23 December 2019; first published online 11 November 2020)

Abstract

Sentiment analysis is directly affected by compositional phenomena in language that act on the prior polarity of the words and phrases found in the text. *Negation* is the most prevalent of these phenomena, and in order to correctly predict sentiment, a classifier must be able to identify negation and disentangle the effect that its scope has on the final polarity of a text. This paper proposes a multi-task approach to explicitly incorporate information about negation in sentiment analysis, which we show outperforms learning negation implicitly in an end-to-end manner. We describe our approach, a cascading and hierarchical neural architecture with selective sharing of Long Short-term Memory layers, and show that explicitly training the model with negation as an auxiliary task helps improve the main task of sentiment analysis. The effect is demonstrated across several different standard English-language data sets for both tasks, and we analyze several aspects of our system related to its performance, varying types and amounts of input data and different multi-task setups.

Keywords: sentiment analysis; negation detection; multi-task

1. Introduction

The goal of sentiment analysis is to assign a polarity (either categorical or real valued) to text and has become a popular task in natural language processing thanks to a growing interest in automatically processing the large amount of opinionated text available on the internet. Consider the following example sentence from a movie review, taken from the Stanford Sentiment Treebank (Socher *et al*. 2013), where we have added annotations to indicate words with prior positive polarity (blue boxes), negation cues (bold face), and the scopes of the cues (underlined):

(1) Being unique **doesn't** necessarily equate to being good, **no matter** how admirably the filmmakers have gone for broke.

In this short sentence, there is a subtle negative sentiment expressed toward the movie through the negation of the phrase "necessarily equate to being good". This example points out how the sentiment of a sentence is not merely the sum of the polarity of the words and phrases found in the text, but rather depends on a number of compositional phenomena that act on indicators of polarity. *Negation* is one of the most pervasive of these phenomena.

In order to adequately deal with the phenomenon of negation in sentiment analysis, it is not enough to simply detect single words indicating negation, so-called negation cues, as the scope of this negation is equally important. In Example (2), there is negation, but the relevant polar adjectives "unique" and "well-crafted" are not within its scope (the red box indicates prior negative polarity).

© The Author(s), 2020. Published by Cambridge University Press.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

(2) It's **not** so much a work of entertainment as it is a unique, well-crafted psychological study of grief.

A sentiment classification system that takes a naive view of negation would likely classify the sentence in (2) as negative, as negation cues often lead models to predict more negative sentiment (Wiegand *et al.* 2010; Barnes, Øvrelid, and Velldal 2019a). Previous research also demonstrates the need and utility for incorporating negation information in sentiment models (Wiegand *et al.* 2010; Lapponi, Read, and Øvrelid 2012a; Cruz, Taboada, and Mitkov 2016). Approaches that use negation information to improve sentiment analysis can largely be divided into three broad categories:

1. approaches that use *heuristic polarity modification* where the prior polarity of a word is modified if found within some given radius of a negation cue (Hu and Liu 2004; Taboada *et al.* 2011),
2. approaches that *augment the classification feature space* with negation-relevant features (Pang, Lee, and Vaithyanathan 2002; Das and Chen 2007; Lapponi *et al.* 2012a),
3. or *end-to-end approaches* where the model is assumed to capture the effects of negation without being provided explicit negation annotations (Socher *et al.* 2013; Irsoy and Cardie 2014).

However, most of the previous approaches to incorporating negation information into sentiment modeling do not take full advantage of the large body of work that exists on negation detection as a task of its own, both in terms of modeling (Morante and Daelemans 2009; Read *et al.* 2012; Fancellu, Lopez, and Webber 2016) and data sets (Morante and Blanco 2012; Konstantinova *et al.* 2012). One likely reason for this is that it is not obvious how to best incorporate negation information into state-of-the-art sentiment models. In this paper, we apply multi-task learning to incorporate information from data sets explicitly annotated for negation in order to improve the performance of sentiment classifiers on English-language data sets.

Contributions: In this work, we make the following contributions:

1. we propose a hierarchical multi-task learning approach to incorporate negation information into a sentiment classifier,
2. we show that multi-task learning can lead to improvements despite a difference in the relevant units of classification (e.g. sentence-level sentiment and sequence-labeled negation scope), and
3. we provide a detailed analysis of the effects of multi-task learning of negation for sentiment analysis.

We additionally make the data and code available[a] in order to encourage reproducibility. In the remainder of the paper, we first discuss related work (Section 2), then describe the data used in all experiments (Section 3), and detail our proposed cascading multi-task model in Section 4. We then describe the results of the main experiment (Section 5) and perform a thorough analysis of the most important variables in Section 6. Finally, we discuss the implications of our findings and future work in Section 7.

2. Related work

This section first outlines some of the previous work done on handling negation – both as a part of sentiment analysis and as a separate task in itself. We then review some relevant previous work

[a]https://github.com/ltgoslo/multitask_negation_for_sa.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

	There	is	no	flowery	dialog	,	and	time	is	n't	wasted	...
negation labels:	O	O	CUE[1]	N[1]	N[1]	O	O	O	CUE[2]	CUE[2]	N[2]	
BIO labels:	O	O	B_cue	B_neg	I_neg	O	O	O	B_cue	I_cue	B_neg	

Figure 1. An example of negation detection annotation on a sample sentence from the SFU data set.

on sentiment analysis more generally and finally provide some background on previous work on multi-task learning in Natural Language Processing (NLP).

2.1 Negation in sentiment models

Negation is a frequent linguistic phenomenon which has a direct impact on sentiment analysis (Wiegand *et al*. 2010). Within the framework of lexicon-based sentiment analysis, researchers first attempted to model negation with simple heuristics, such as reversing (Hu and Liu 2004; Polanyi and Zaenen 2006; Kennedy and Inkpen 2006) or modifying (Taboada *et al*. 2011) the polarity signal of a negated word. This approach to tackle contextual valence shifting generally assumes that the final polarity of a text is some function of the prior polarities of adjectives, verbs, and nouns found in the text. The scope of negation is determined heuristically, by finding common negation cues and assuming all words between the cue and the next punctuation are in scope (Hu and Liu 2004) or based on the distance from the cue (Taboada *et al*. 2011).

Early machine learning approaches to sentiment analysis also used heuristics, such as attaching a negation tag ("_neg") to words assumed to be in scope (Pang *et al*. 2002; Das and Chen 2007). This approach, however, leads to an increase in sparsity and varying results, as the sentiment model is not able to explicitly connect the original and negated features. Other research has used negation detection systems to enhance the feature space of sentiment models (Councill *et al*. 2010; Lapponi *et al*. 2012a; Cruz *et al*. 2016), leading to improved results on sentiment classification. Additionally, certain negation cues contribute to higher shifts in polarity than others (Zhu *et al*. 2014), which indicates they should be modeled separately.

More recent advances in negation detection, both in terms of modeling and data annotation, have not been incorporated into sentiment classification models so far, to the best of our knowledge.

2.2 Negation detection

Previously reported approaches to negation analysis commonly break it down to (at least) two sub-tasks, performing (i) negation cue detection, followed by (ii) scope detection. The example in Figure 1 shows the negation annotation of the following sentence (in the first row):

(3) There is **no** flowery dialog, and time **isn't** wasted.

For our example in Figure 1, the cue detection component would locate the negation cues in the sentence, that is, the negative determiner "no", and the copula with its negative contraction "is n't", whereas the scope detection module would recognize the noun phrase "flowery dialog" and the verb phrase "wasted" as the scopes of these cues, respectively. Depending on the specific annotation scheme, subjects may or may not be part of the scope of negation.

A large portion of early work on negation detection (Morante, Liekens, and Daelemans 2008; Morante and Daelemans 2009; Velldal *et al*. 2012) has been done within the biomedical domain due to the availability of the BioScope corpus (Vincze *et al*. 2008), which is annotated for negation cues and scopes. Interest in the task was further spurred by the *SEM shared task (Morante and Blanco 2012), which focused on the detection of negation cues and scopes, in addition to detection of negated events and their so-called focus. The shared task made available the ConanDoyle-neg

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

corpus, which is described in Section 3. A number of systems were submitted for this task, employing a wide variety of strategies. For example, the best performing systems for the closed track and open track employed, respectively, Support Vector Machine (SVM)-based ranking of constituent (sub-)trees (Read *et al.* 2012) and conditional random field (CRF)-based sequence labeling using dependency features (Lapponi *et al.* 2012b).

Traditional approaches to the task of negation detection have typically employed a wide range of hand-crafted features describing a number of both lexical, morphosyntactic, and even semantic properties of the input text. Syntactic parsing has often been used to analyze the input prior to negation detection and has been based on both constituency-based (Read *et al.* 2012; Packard *et al.* 2014) and dependency-based representations (Lapponi *et al.* 2012a; White 2012; Enger, Velldal, and Øvrelid 2017). It is also possible to combine an existing system (Read *et al.* 2012) with an additional layer of manually defined rules over Minimal Recursion Semantics structures created by an Head-driven Phrase Structure Grammar (HPSG) parser (Packard *et al.* 2014).

There are a few previous studies that investigate neural modeling for the task of negation detection. Among these, we find a CNN model for the negation scope detection on the abstracts section of the BioScope corpus, which operates over syntactic paths between the cue and candidate tokens (Qian *et al.* 2016). Fancellu *et al.* (2016) present and compare two neural architectures for the task of negation scope detection on the ConanDoyle-neg corpus: a simple feed-forward network and a bidirectional LSTM (BiLSTM). Note that these more recent neural systems disregard the task of cue detection altogether (Fancellu *et al.* 2016; Qian *et al.* 2016; Fancellu *et al.* 2017), relying instead on gold cues and focusing solely on the task of scope detection.

While syntactic information has often been found useful for scope resolution, the task of cue detection appears to only require simpler surface information. Velldal *et al.* (2012) present an approach to cue detection which treats the set of cue words as a closed class and apply a disambiguation-based approach to the problem of cue detection, showing that simple lexical features based on a narrow context window are sufficient to achieve good performance.

As further detailed in Section 4.1, in the current paper, we model cue detection and scope resolution concurrently as a sequence labeling task, using a BIO label encoding which is illustrated in the final row of Figure 1. BIO labeling for negation detection has been employed in previous work, following the early work of Morante *et al.* (2008). Our joint modeling of cue and scope detection differs from previous approaches to negation detection as reviewed above that handle cue and scope resolution as two separate tasks (Morante and Blanco 2012; Lapponi *et al.* 2012b; Read *et al.* 2012; Fancellu *et al.* 2016; Cruz *et al.* 2016; Qian *et al.* 2016). Moreover, motivated by the assumption that downstream tasks like sentiment analysis only need information about which words are within the scope of some negation, regardless of which particular cue it relates to (in cases where more than one cue is present), we do not attempt to explicitly retain this coupling. Similar to Fancellu *et al.* (2016), we use a BiLSTM-based model, relying not only on word embeddings as input, but also adding a CRF for the prediction layer. When it comes to incorporating information about negation to our sentiment model, our approach takes advantage of the representation learning capabilities of neural models: rather than passing on the negation predictions output by the final CRF layer, we pass on the intermediate representations learned by the BiLSTM. The details of this cascading architecture are further described in Section 4.

2.3 Sentiment analysis

Approaches to sentiment analysis have moved from lexicon-based methods (Turney 2002; Hu and Liu 2004; Taboada *et al.* 2011), to machine learning methods based on hand-derived features (Pang *et al.* 2002; Pang and Lee 2008) and finally to neural networks that learn to extract useful features in an end-to-end fashion (Socher *et al.* 2013; Tang *et al.* 2014; Tai, Socher, and Manning 2015). While some of these neural architectures have been tailored to suit specific tasks better

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

(Irsoy and Cardie 2014; Lei *et al*. 2018), two recent end-to-end architectures have shown competitive results on a large number of natural language processing tasks: BiLSTMs Networks (Graves and Schmidhuber 2005) and Self-Attention Networks (SANs) (Vaswani *et al*. 2017). Variants of these two architectures give state-of-the-art results on document-level (Howard and Ruder 2018), sentence-level (Peters *et al*. 2018; Devlin *et al*. 2018), and aspect-level (Xu *et al*. 2019) sentiment analysis tasks.

The claim made by the proponents of end-to-end learning is that the models implicitly learn compositional functions (Socher *et al*. 2013; Irsoy and Cardie 2014), thereby removing the need to explicitly provide information about inter-word dependencies, negation, or speculation in the form of hand-crafted features. Recent research, however, challenges the idea that end-to-end learning is able to fully capture compositional effects (Verma, Kim, and Walter 2018; Barnes *et al*. 2019a). It is therefore worth asking whether we can help the model by providing some form of explicit training on compositional phenomena in sentiment.

2.4 Multi-task learning

Multi-task learning (MTL) (Caruana 1993) stems from the idea that learning-related tasks simultaneously allow a machine learning algorithm to incorporate a useful inductive bias by restricting the search space of possible representations to those that are predictive for both tasks. MTL assumes that features that are useful for a certain task should also be predictive for similar tasks, and in this sense, MTL also effectively acts as a regularizer, as it prevents the weights from adapting too much to just one task. Under some circumstances, multi-task learning can also be seen as a kind of data augmentation, where an MTL model takes advantage of extra training data available in an auxiliary task to improve the main task (Kshirsagar *et al*. 2015; Plank 2016; Fares, Oepen, and Velldal 2018). MTL is particularly well-suited for neural models, given the possibilities for modular design and representation learning. Below we outline some of the different ways that multi-task learning can be set up, while also reviewing previous MTL efforts in NLP.

Hard parameter sharing (Caruana 1993), which assumes that all layers are shared between tasks except for the final predictive layer, is the simplest way to implement a multi-task model. When the main task and auxiliary task are closely related, this approach has been shown to be an effective way to improve model performance (Collobert *et al*. 2011; Peng and Dredze 2017; Martínez Alonso and Plank 2017; Augenstein, Ruder, and Søgaard 2018). Other research (Søgaard and Goldberg 2016), on the other hand, finds that it is better to make predictions for low-level auxiliary tasks at lower layers of a multi-layer MTL setup. They also suggest that under the hard-parameter framework, auxiliary tasks need to be sufficiently similar to the main task for MTL to improve over the single-task baseline.

There have also been several effective implementations of *soft parameter sharing*, where two models have both shared and private task-specific parameters, such as including a gating mechanism that allows a MTL model to select which information to share across tasks (Liu, Qiu, and Huang 2016; Ruder *et al*. 2019). Their results suggest that hard-parameter sharing is only beneficial for low-level tasks, while for high-level tasks, it is better to learn how much to share at each layer and subspace of parameters in the network. Furthermore, they find that MTL is more beneficial when there is less training data for the main task and that modeling subspaces explicitly helps in almost all domains.

What characteristics of an auxiliary task are necessary to improve a main task is still largely unknown. Some research suggests high-level semantic auxiliary tasks generally help more than low-level auxiliary tasks as MTL tends to work when the main task learning plateaus quickly and the auxiliary task learning does not (Bingel and Søgaard 2017). Others find that auxiliary tasks with compact, uniform label distributions are preferable (Martínez Alonso and Plank 2017). Additionally, choosing a suitable auxiliary task is still vital, as introducing an unsuitable auxiliary task can actually hurt performance (Augenstein and Søgaard 2017).

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Table 1. Overview of the data sets for sentiment and negation. Note that for SST, SFU, and Conan Doyle Neg, we show the number of sentences, while for SemEval 2013, we show the number of tweets

Task		Data set	Train	Dev.	Test
Sentiment	Socher *et al.* (2013)	SST-binary	6920	872	1821
		SST-fine	8455	1101	2210
	Nakov *et al.* (2013)	SemEval2013-binary	3056	491	1262
		SemEval2013-fine	6021	890	2376
Negation	Konstantinova *et al.* (2012)	SFU review corpus	800	71	96
	Morante and Blanco (2012)	Conan Doyle neg	842	144	235

In this work, we propose that MTL is an appropriate framework to incorporate negation detection in a sentiment classifier. Unlike previous approaches in sentiment analysis (Councill *et al.* 2010; Lapponi *et al.* 2012a; Cruz *et al.* 2016), our method does not rely on incorporating negation information as explicit features, but rather uses a cascading architecture where the intermediate representations learned for predicting negation feeds into subsequent layers (along with skip connections) for predicting sentiment. While the final layers of the network hierarchy are dedicated to the sentiment task, the lower layers are shared and supervised by both tasks. The components of the architecture are further detailed in Section 4. Note that, for comparison, we also explore using other auxiliary tasks beyond negation.

In parallel work to this, Barnes *et al.* (2019b) showed that a similar cascading or hierarchical MTL architecture could be used for incorporating information from sentiment lexicons to improve models for sentence-level sentiment classification. Also in parallel work, Sanh, Wolf, and Ruder (2019) apply hierarchical MTL to learn shared representations for a set of semantic tasks where lower-level task like named entity recognition and entity mention feeds into higher-level tasks like coreference resolution and relation extraction.

3. Data sets

As outlined above, we propose to model both sentiment classification and negation detection in a multi-task learning setup. Unlike much previous work in MTL (Bingel and Søgaard 2017; Augenstein and Søgaard 2017; Bjerva 2017; Ruder *et al.* 2019) which assumes several prediction tasks annotated on the same data set with the same output units (token-level sequence labeling), we take auxiliary data from different data sets and domains and with different units of classification across tasks: We experiment with sentence- and tweet-level classification of sentiment as a main task, while learning sequence labeling of negation cues and scopes based on two different negation data sets as an auxiliary task. This section describes the different data sets we use (shown in Table 1).

SFU Review Corpus: This corpus (Konstantinova *et al.* 2012) contains 400 reviews from eight domains (books, cars, computers, cookware, hotels, movies, music, and phones) which have been annotated for sentiment at document level, as well as negation and speculation at sentence level. Although the data set contains sentiment annotations, we do not use these to evaluate the sentiment models, but rather choose to focus on sentence- and tweet-level classification, as compositional effects will have a more direct bearing on the prediction on these finer-grained tasks. The example in Figure 1 illustrates the annotation scheme found in the SFU corpus (top rows). The annotation scheme is based principally on the guidelines developed for the biomedical BioScope corpus (Vincze *et al.* 2008), which largely employ syntactic criteria for the determination

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

of negation scope, choosing the maximal syntactic unit that contains the negated content. Unlike BioScope, however, negation cues are not included within the scope. The SFU corpus does not annotate affixal cues, for example, *im-* in *impossible*. This corpus, however, has the advantage that it stems from the same domain (reviews) as our main task. As there is not a predefined test split, we take 800 sentences annotated for negation as training, 71 for development, and 96 for testing.

ConanDoyle-neg (CD): This widely used corpus contains Conan Doyle stories manually annotated for negation cues, scopes, and events (Morante and Daelemans 2012) and was employed in the 2012 *SEM shared task on negation detection (Morante and Blanco 2012). The shared task version of the data set contains a training set of 3640 sentences, of which 848 sentences contain negation, a development set consisting of 787 sentences, of which 144 are negated, as well as a held-out test set which was constructed specifically for the shared task, consisting of 1089 additional sentences, of which 235 sentences contain negation. The annotation scheme is also based on those employed for the BioScope corpus (Vincze *et al*. 2008), but with some important modifications. In ConanDoyle-neg (CD hereafter), the cue is not included in the scope and it annotates a wide range of cue types, that is, both sub-token (affixal), word-based and multi-word negation cues. Scopes may furthermore be discontinuous, often an effect of the requirement to include the subject within the negation scope. This is in contrast to the annotation scheme found in the SFU corpus, where subjects are not included in negation scope, as is clear from the example in Figure 1, where the subject *time* is not included in the scope of the negation cue *isn't*. In our experiments, we use the pre-defined train, development, and test splits from the shared task.

Stanford Sentiment Treebank (SST): The SST data (Socher *et al*. 2013) contain 11,855 sentences taken from English-language movie reviews. It was annotated for fine-grained sentiment (strong negative, negative, neutral, positive, and strong positive) which we refer to as SST-fine and can also be mapped to a binary setting (SST-binary), where the neutral class is removed and strong and normal examples are merged (9613 sentences). We perform experiments with both setups, using the pre-defined train, development, and test splits.

SemEval 2013: The SemEval 2013 shared task on tweet-level sentiment analysis (Nakov *et al*. 2013) contains 9287 tweets annotated for three-way sentiment (positive, neutral, and negative), which we refer to as SemEval-fine. Additionally, we remove the tweets with neutral labels to give a binary setup (SemEval-binary). We use the train, development, and test splits given in the shared task.

4. A cascading multi-task model for negation and sentiment

This section details our neural architecture for multi-task learning of negation and sentiment, as shown in Figure 2. We adopt a cascading architecture where the lower layers are used to perform the auxiliary task – in our case negation cue and scope prediction – and the higher layers are dedicated to the main task – in our case polarity prediction. Adopting the terminology of Goldberg (2017), 'cascading' here refers to the fact that rather than passing on the negation predictions as such, the lower layers pass on the intermediate representations learned for making these predictions. The multi-task learning setup means that the shared lower layers will receive supervision signals from both the sentiment and negation tasks. This setup also aligns well with the findings of Søgaard and Goldberg (2016) that MTL models tend to benefit more from lower-level auxiliary tasks at lower layers of the network. We detail the different components in more detail below.

4.1 Negation model

We start by discussing the part of the model responsible for detecting negation cues and scopes, including how it relates to some of the previously reported approaches that are most directly relevant.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

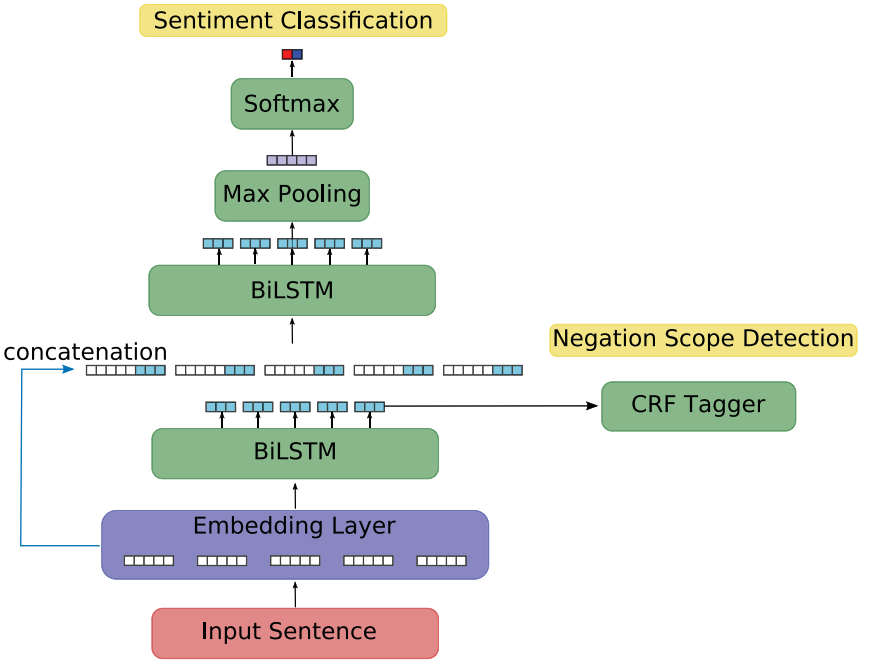


Figure 2. Our proposed multi-task model.

Similarly to Fancellu *et al.* (2016), we use a BiLSTM network (Graves and Schmidhuber 2005) to extract features from the embedding layer, but where Fancellu *et al.* (2016) use a linear softmax layer for prediction, we use a linear-chain CRF with Viterbi decoding to find the most probable assignment of labels, as CRFs can better model sequences. Moreover, while Fancellu *et al.* (2016) assume gold cues, encoded as separate cue embeddings concatenated to the word embeddings provided as input, we here let the BiLSTM to predict both cues and scopes – performed in one pass.

Note that there might be several instances of negation in the same sentence, as in the example of Figure 1. In the setup of Fancellu *et al.* (2016), each instance is multiplied out into a separate example, effectively duplicating the sentence for each pair of cue and scope. In our setup, all instances will be treated in the same pass. In the CRF model of Lapponi *et al.* (2012b) too, all scopes are predicted in one pass – although cues are there predicted in a preceding step using an SVM classifier as in Read *et al.* (2012) – but then post-processing heuristics are applied to assign the identified negation tokens to their respective cues.

A simplifying assumption made in our model is that we do not care about explicitly preserving the links between particular cues and scopes in our output; intuitively, the important information for a downstream task like sentiment analysis is whether a token is within the scope of negation, regardless of the identity of the negation cue. Additionally, since we do not incorporate sub-token information in our model, we treat any token annotated with morphological negation, for example *un-* or *-less*, as a negation cue.

As shown in Figure 2, given a sequence of tokens, our negation model first embeds these in an embedding layer and then uses a BiLSTM to create a contextualized representation of each token. This representation is then used as features in the CRF. In our experiments, we use Viterbi decoding to find the most probable assignment of labels and train the model to minimize the negative log likelihood.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

4.2 Sentiment model

The sentiment model uses the same embedding layer and the first BiLSTM layer to create the contextualized representation of the input tokens. We make use of skip connections where we concatenate each of the original embeddings to the contextualized representations. This sequence then serves as input to a second sentiment-specific BiLSTM layer.

Finally, we perform a max pooling operation on the output of the sentiment-specific BiLSTM and pass this max-pooled representation to a softmax layer to compute the class probabilities. We then minimize the cross entropy loss of the sentiment predictions with respect to the true sentiment.

During training, the model alternates between training one epoch on the main task and one epoch on the auxiliary task. Preliminary experiments showed that more complicated training strategies (alternating training between each batch or uniformly sampling batches from the two tasks) did not lead to improvements. Note that we do not upsample negation data. We train the model for 10 epochs using Adam (Kingma and Ba 2014), performing early stopping determined by accuracy on the development set. We regularize[b] with dropout before the BiLSTM layers (0.5), between BiLSTM layers (0.3), apply batch norm, and L2 regularization (0.0001). As neural models are sensitive to the random initialization of their parameters, we perform five runs with different random seeds and show the mean and standard deviation as the final result for each model.[c]

5. Results and comparison with state of the art

Table 2 shows the mean accuracy and standard deviation of single-task sentiment models (STL), multi-task models with SFU auxiliary negation data (MTL-SFU), and multi-task models with ConanDoyle-neg auxiliary negation data (MTL-CD) over five runs. One important design decision in these experiments is that, in order to isolate the effects of multi-task learning, we make sure all models have the same capacity in terms of number of parameters: The single-task models also include the lower BiLSTM layers, the difference being that they are supervised by the sentiment task only.

It is important to note that the objective of the current paper is not to achieve new state-of-the-art results for sentiment analysis, but rather to gauge the relative contribution of negation as an auxiliary task using MTL. Nonetheless, we also include a comparison with the following sentiment models:

- BOW: a L2-regularized logistic regression model trained on a bag-of-words representation (Barnes, Klinger, and Schulte im Walde 2017).
- CNN: a one-layer convolutional neural network with one convolutional layer on top of pre-trained word embeddings (Barnes *et al.* 2017).
- BiLSTM: a bidirectional LSTM creates a hidden representation from pre-trained word embeddings, which is then mean pooled and fed to a feed-forward network (Barnes *et al.* 2017).
- SAN+RPR: a SAN with relative position representations (Ambartsoumian and Popowich 2018).
- Tree-LSTM: a recursive LSTM that uses parse trees annotated for sentiment at each node as input (Tai *et al.* 2015).
- BERT: a large SAN pre-trained on a cloze-like language modeling task and then fine-tuned on the main task (Devlin *et al.* 2018).

[b] These optimal values were chosen by observing performance on the development set when training only on the main task and kept stable through all experiments.

[c] We use the same five random seeds for all experiments to ensure a fair comparison between models.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Table 2. Mean accuracy and standard deviation of STL and MTL models over five runs on the main sentiment task. The MTL model trained with negation outperforms the single-task baseline in both fine-grained and binary setups. Underlined results indicate the best overall approach, while **bold** results show where the MTL model outperforms the STL. A star (*) indicates that the model performs significantly better ($p < 0.01$), according to approximate randomization tests. We do not report results for SemEval-binary for baseline models or SemEval-fine for Tree-LSTM because the previous work does not report results on this data

Model	SST-binary	SST-fine	SemEval-binary	SemEval-fine
BOW	80.7	40.3	–	65.5
CNN	81.3 (1.1)	39.8 (0.7)	–	63.5 (1.3)
BiLSTM	82.6 (0.7)	45.6 (0.7)	–	65.1 (0.9)
SAN + RPR	84.2 (0.4)	48.1 (0.4)	–	72.2 (0.8)
Tree-LSTM	88.0 (0.3)	51.0 (0.5)	–	–
BERT	94.9	53.0	–	75.1
HEUR	83.78 (0.3)	45.29 (1.6)	82.59 (1.4)	65.59 (1.3)
STL	84.57 (1.0)	46.49 (0.7)	84.00 (1.0)	67.26 (0.6)
MTL-SFU	**86.04 (0.3)***	**46.75 (0.8)**	**84.02 (0.9)**	67.03 (1.0)
MTL-CD	**85.43 (0.9)***	**47.33 (0.6)***	83.52 (1.6)	**67.75 (1.0)***

- HEUR: this model is identical to the STL model, but incorporates three negation embeddings (one each for negation cue, in-scope, and out-of-scope), which are learned during training and are concatenated to the word embeddings before being passed to the LSTM modules. The decision for which embedding to concatenate comes from performing a heuristic negation processing where any token from a negation cue to the next punctuation mark is considered in scope.

The single-task model (STL) achieves an average accuracy of 84.57 on SST-binary, 46.49 on SST-fine, 84.0 on SemEval-binary, and 67.26 on SemEval-fine. These results are better than standard performance for a Bidirectional LSTM model (82.6/45.6/-/65.1) and competitive with similar models. The improvement most likely derives from the extra BiLSTM layer, skip connections, and the max-pooling operation before the softmax layer. Previous state-of-the-art BiLSTM models (Barnes *et al.* 2017) instead use a single-layer BiLSTM with mean pooling. The STL model outperforms the SAN+RPR model on SST-binary, but performs worse on the SST-fine and SemEval-fine tasks. The Tree-LSTM outperforms the STL model on the SST data sets, while the BERT model is the best performing system overall. Note that these final two approaches have access to a much larger quantity of data than the others, either in the form of phrase-level annotations for the Tree-LSTM or language model pre-training on more than three billion words in the case of BERT. The HEUR method, however, performs poorly, not even reaching the performance of STL.

The MTL models outperform the STL models on six of the eight experiments. The MTL-SFU model achieves accuracies of 86.04 (+1.47 percentage points (ppt.)), 46.75 (+0.26 ppt.), 84.02 (+0.02 ppt.), and 67.03 (−0.23 ppt.), improving over the STL on the first three tasks, while the MTL-CD model data have an accuracy of 85.43 (+0.86 ppt.), 47.33 (+0.84 ppt.), 83.53 (−0.48 ppt.), and 67.75 (+0.48 ppt.). Interestingly, the MTL-SFU model is the best performing model on both binary tasks, while the MTL-CD model is the best on both fine-grained tasks. Note that while SAN, Tree-LSTM, and BERT perform better than our proposed models, we choose to use BiLSTMs as it is both easier and faster to perform multi-task learning, allowing for a

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

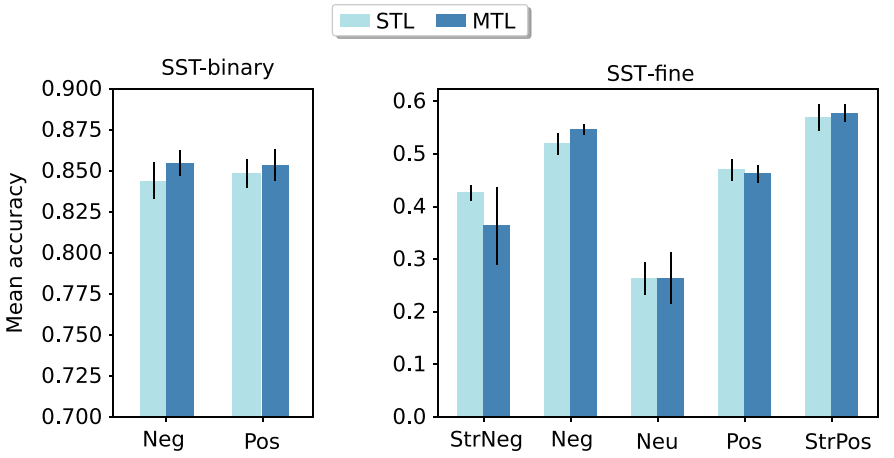


Figure 3. Mean accuracy and standard deviation of STL and MTL-CD model on the SST-binary and SST-fine tasks, broken down across the two (five) classes.

deeper analysis. Given the results, however, it is clear that follow-up work should concentrate on incorporating the negation information into more advanced models.

We test the significance by performing approximate randomization testing (Yeh 2000) with 10,000 iterations pairwise between the results of each of the five runs.[d] We consider results significant if the difference between models in at least three of the five runs[e] is statistically significant ($p < 0.01$ which corresponds to a Bonferroni correction for five hypotheses). MTL models perform significantly better than the STL baseline in four of the eight experiments.

6. Model analysis

In this section, we include detailed analyses of several aspects of our model. The first analysis is an error analysis that gives a more qualitative view of the results (Section 6.1). We then perform an analysis of the impact of data set size (both for the main and auxiliary tasks), and data set composition (Sections 6.2–6.5). Finally, we evaluate several components of the multi-task learning setup (Sections 6.6–6.7).

6.1 Error analysis

A per class evaluation of the SST-binary and SST-fine tasks (Figure 3) shows what effect multi-task learning has on each sentiment class.[f] On SST-binary, the MTL model improves on both positive and negative classes. On the SST-fine task, however, the model improves only on the negative and strong positive classes, performing worse on strong negative and positive, while performing nearly the same on neutral. An analysis of the data shows that the negative class contains the largest percentage of negated sentences (27%), while the strong positive has the least (13%). It is possible that the MTL model is able to better discriminate relevant and non-relevant negation. In the example from the SST-fine task (4) below, the STL model assigned the sentence a positive label

[d] We use a re-implementation of the sigf package (Padó 2006).
[e] Although *t*-tests are common in such situations, we opted against this as the independence assumptions do not hold.
[f] We show both results from the MTL-CD model, in order to isolate the effects of the multitask training from differences in data.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Table 3. Mean accuracy and standard deviation of STL and MTL models on the negated and non-negated subsets of the SST test data. Bold results show the best model on each data sub-set.

Data set	Model	Sub-set accuracy	
		Negated	Non-negated
SST-binary	STL	78.9 (1.2)	85.8 (0.9)
	MTL	**81.1 (1.5)**	**87.1 (0.2)**
SST-fine	STL	**41.1 (1.4)**	47.6 (0.4)
	MTL	40.7 (1.2)	**47.9 (0.9)**

due largely to the number of positive tokens, while the MTL-CD model correctly predicted the negative label, as it was able to resolve the negation.

(4) Accuracy and realism are terrific, but if your film becomes boring, and your dialogue isn't smart, then you need to use more poetic license.

The previous analysis suggests that the multi-task setup is beneficial for sentiment analysis, but does not confirm that the model is actually learning better representations for negated sentences. Here, we look at how each model performs on negated and non-negated sentences.

As we do not have access to gold negation annotations on the main task sentiment data, we create silver data by assuming that any sentence that has a negation cue (taken from SFU) is negated. We then extract the negated and non-negated sentences from the SST fine-grained (397 negated/1813 non-negated) and binary (319/1502) test sets. While this inevitably introduces some noise, it allows us to observe general trends regarding these two classes of sentences.

Table 3 shows the results of the analysis. On the binary task, the MTL model performs better on both the negated (+2.2 ppt.) and non-negated (+1.3 ppt.) subsets. On the fine-grained task, however, the STL model outperforms the MTL model on the negated subsection (−0.4 ppt.), while the MTL model performs better on the non-negated subsection (+0.3 ppt.). More detailed analysis would be needed to explain why the binary-task STL model outperforms the MTL model on our silver-standard negated sample. However, when it comes to the better performance of the MTL model on the non-negated sample – for both the binary and fine-grained task – one possible explanation is that learning about negation also enables the model to make more reliable predictions about sentiment bearing words that it has perhaps only seen in a negated context during training, but outside of negation during testing.

6.2 Impact of data size

In order to better understand the effects of multi-task learning of negation detection, we compute learning curves with respect to the negation data for the SST-binary setup. The model is given access to an increasing number of negation examples from the SFU data set (from 10 to 800 in intervals of 100), and accuracy is calculated for each number of examples. Figure 4 (left) shows that the MTL model improves over the baseline with as few as 10 negation examples and plateaus somewhere near 600. An analysis on the SST-fine setup showed a similar pattern. There is nearly always an effect of diminishing returns when it comes to adding training examples, but if we were to instead plot this learning curve with a log-scale on the *x*-axis, that is doubling the amount data for each increment, it would seem to indicate that having more data could indeed still prove useful, as long as there were sufficient amounts. In any case, regardless of the amount of data, exposing

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

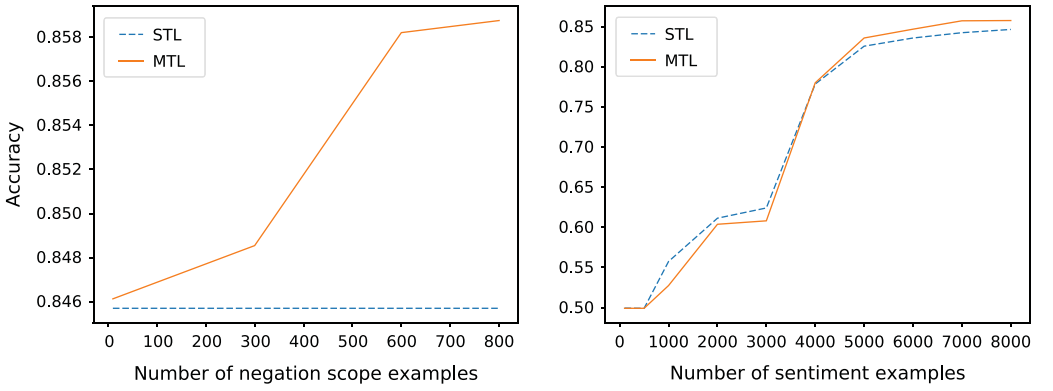


Figure 4. Mean accuracy on the SST-binary task when training MTL negation model with differing amounts of negation data from the SFU data set (left) and sentiment data (right).

the model to a larger *variety* of negation examples could also prove beneficial – we follow up on this point in the next subsection.

While the previous experiment shows that models already improve with as few as 10 auxiliary examples, here we investigate whether a sentiment model benefits from multi-task learning more when there is limited sentiment data, as previous research has shown for other tasks (Ruder *et al*. 2019). We keep the amount of auxiliary training data steady and instead vary the sentiment training data from 100–8000 examples.

Figure 4 (right) shows that the performance of the STL model begins to plateau at around 5000 training examples (although note the comment about diminishing returns above). Until this point, the MTL model performs either worse or similarly. From 5000 on, however, the MTL model is always better. Therefore, negation detection cannot be used to supplement a model when there is lacking sentiment data, but rather can improve a strong model. This may be a result of using a relevant auxiliary task which has a different labeling unit (sequence labeling vs. sentence classification), as other research (Ruder *et al*. 2019) suggests that for similar tasks, we should see improvements with less main task data.

6.3 Can we combine negation data despite differences in annotation?

The previous experiment suggests that more negation data will not necessarily lead to large improvements. However, the model trained on the SFU negation data set performs better on the SST-binary task, while the CD negation model is better on SST-fine. In this section, we ask whether a combination of the two negation data sets will give better results, despite the fact that they have conflicting annotations, see Section 3 above.

We train an MTL model on the concatenation of the SFU and CD train sets (MTL-Combined) using the same hyperparameters as in the previous experiments. The results in Table 4 show that MTL-Combined performs worse than the MTL-SFU model on SST-binary, while it is the best performing model by a small margin ($p > 0.01$ with approximate randomization tests as described in Section 5) on SST-fine. This shows that simply combining the negation data sets does not necessarily lead to improved MTL results, which is most likely due to the differences in the annotation schemes.

6.4 Scopes or cues

As described in the initial sections, negation is usually represented by a negation cue and its scope. One interesting question to address is whether both of these are equally important for downstream use in our multi-task setup. Here, we investigate whether it is enough to learn to identify only cues

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Table 4. Combining the SFU and CD negation data (MTL-Combined) does not lead to large improvements. Bold results show the best performing model on each dataset.

Setup	STL	MTL-SFU	MTL-CD	MTL-Combined
SST-binary	84.57 (1.0)	**86.04 (0.3)**	85.43 (0.9)	85.54 (0.7)
SST-fine	46.49 (0.7)	46.75 (0.8)	47.33 (0.6)	**47.42 (0.3)**

Table 5. Mean accuracy and standard deviation on the sentiment task for the STL model, MTL models trained to predict only negation cues or only negation scope, and finally the MTL model trained to predict both scopes and cues. Bold results show the best performing model on each dataset.

Setup	STL	MTL-Cues	MTL-Scopes	MTL-Both
SST-binary	84.57 (1.0)	85.94 (0.2)	85.60 (1.3)	**86.04 (0.3)**
SST-fine	46.49 (0.7)	46.43 (1.1)	46.48 (1.2)	**47.33 (0.6)**

Table 6. Number of training, development, and test examples for the Stanford Sentiment Treebank phrase-level data

Setup	Train	Dev.	Test	Total
SST-binary	75,646	10,099	19,972	105,717
SST-fine	155,019	20,173	40,195	215,387

or only scopes. We train the MTL model from Section 4 to predict only cues or only scopes and compare their results with the STL model and the original MTL model which predicts both.

The results of each experiment are shown in Table 5. Learning to predict only cues or only scopes performs worse than the MTL model trained to predict both. Additionally, learning to predict only one of the two elements also performs worse than STL on the fine-grained setup. This indicates that it is necessary to learn to predict both scopes and cues. One likely explanation for this is that the cue predictions in turn benefits scope predictions. On the SST-binary, the MTL-Cues model performs better than the MTL-Scopes model, while the opposite is true for the SST-fine task, indicating that it is more important to correctly predict the negation scope for the fine-grained setting than the binary.

6.5 Can models trained on phrase-level data improve with MTL?

Besides the sentence-level annotations we have used so far, the SST also contains sentiment annotations at each node of a constituent tree (i.e., its constituent phrases) for all sentences (statistics are shown in Table 6). The sentence "It was fun to watch", for example, would contain the constituents "It", "was", "fun", "to", "watch", "was fun to watch", "fun to watch", "to watch", each with its corresponding polarity label.

Although originally intended to enable recursive approaches to sentiment, it has also been shown that training a non-recursive model with these annotated phrases leads to models that are better at capturing compositionality effects (Iyyer *et al.* 2015). This is likely because, given a sentence such as "The movie was not great", models are explicitly shown that "great" is positive, while "not great" is negative. A relevant question is therefore whether this phrase-level annotation of sentiment reduces the need for explicit negation annotation. In this section, we compare

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Table 7. Mean accuracy and standard deviation of STL, MTL, and STL-Phrase-level models. Bold results show the best performing model on each dataset.

Setup	STL	MTL	STL-phrase	MTL-phrase
SST-binary	84.57 (1.0)	86.04 (0.3)	87.82 (0.4)	**88.18 (0.5)**
SST-fine	46.49 (0.7)	47.33 (0.6)	49.10 (1.2)	**49.71 (0.4)**

training on these sentiment-annotated phrases to multi-task learning on negation annotations, and also the combination of these two approaches.

We train the STL model from Section 4 on the phrase-level SST data (STL-phrase) and compare with the MTL model trained with phrase-level SST data and with negation as an auxiliary task (MTL-phrase). In order to fairly compare with models trained only on sentence-level annotation, we test on the sentence-level SST data described in Section 3. The results in Table 7 show that even though the largest gains are found by training on the phrase-level data, multi-task learning (MTL) of negation still provides small but consistent gains. This indicates that while end-to-end models may learn some compositional functions implicitly when trained on phrase-level data, there is still room for further improvements by combining this with training on explicit negation annotations in addition. However, while the MTL approach is generally applicable to any sentiment data set, the phrase-level annotations are particular to the SST data.

6.6 Evaluating the negation component: A case for transfer learning?

Although our interest in *negation* modeling in this paper is primarily tied to its influence on sentiment analysis, we do, however, also want to evaluate negation performance in isolation, just to make sure the model is reasonable. There are a number of evaluation metrics used for negation detection. For example, F_1 scores can be computed with respect to cues or scopes or both, either requiring an exact match of predicted spans or allowing for partial matches, or evaluating on the token level. A range of different metrics were implemented for the *SEM 2012 shared task on negation, see Morante and Blanco (2012) for an overview. In this section, we report F_1 for cues and scopes separately, both on the token level. The latter corresponds to the measure called 'scope tokens' in Morante and Blanco (2012) and Fancellu *et al.* (2016).

Table 8 compares our best performing MTL sentiment models with a setup where the negation component of the architecture – corresponding to only the first-layer BiLSTM+CRF as shown in Figure 2 – is trained as a single-task model for negation prediction. The single-task negation model achieves a token-level scope F_1 score of 89.23 on the SFU data and 75.38 on the CD data, while the MTL model reaches 74.69 and 63.81, respectively. As we are optimizing the MTL models for sentiment, the single-task models achieve much better token-level scope F_1 scores (14.5 ppt. on the SFU data, 11.6 on CD). For comparison, the best performing system (Read *et al.* 2012) on CD with respect to the same metric in the *SEM 2012 shared task (Morante and Blanco 2012) achieved an F_1 of 85.26.

An analysis of the common errors shows that neither the STL nor MTL models generalize well to morphological negation cues, for example "unlikely", that have not been seen in training. This is not surprising, given that neither model has access to subtoken information. Of course, this also affects the scopes, as the models rely on predicted cues. Additionally, the MTL model has difficulty in identifying long scopes.

Given that the lower BiLSTM(+CRF) component does not achieve strong results for the auxiliary negation task when trained in the multi-task setup, it is logical to ask if better sentiment predictions can be obtained by starting from a better performing negation model. To test this,

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Table 8. Token-level F_1 for the SFU and ConanDoyle-neg (CD) negation tasks. The single-task negation models outperform the multi-task (MTL) models. Note that the MTL models are not tuned to optimize negation, this being the auxiliary task

Setup	Evaluation	SFU	CD
STL	Scopes	89.23 (0.4)	75.38 (0.9)
	Cues	97.84 (1.0)	86.25 (0.2)
MTL	Scopes	80.45 (7.5)	63.81 (2.8)
	Cues	94.84 (2.2)	84.14 (1.8)

Table 9. Mean accuracy and standard deviation of single-task, multi-task, and transfer-learning models. Bold results show the best performing model on each dataset.

Setup	STL	MTL	Transfer
SST-binary	84.57 (1.0)	**86.04 (0.3)**	85.57 (0.7)
SST-fine	46.49 (0.7)	**47.33 (0.6)**	47.17 (0.4)

we explore a *transfer learning* approach where we pre-train the negation component with a single-task negation objective as described above.

In contrast to the MTL setup, with transfer learning, we first optimize the negation parameters and afterward use these parameters to initialize the lower BiLSTM layer of the sentiment model (cf. Figure 2). These pre-trained parameters are then further fine-tuned when train the overall model on sentiment data, but using a reduced learning rate. Note that this continued training is no longer MTL, however, as the entire network is only supervised by the sentiment task.

Table 9 shows that while transfer learning based on initializing the sentiment model with a pre-trained negation model does show improvements over the STL (1 ppt. on binary and 0.7 on fine-grained), it performs worse than MTL. Counterintuitively, having a better negation detection model, in terms of performance on the negation data sets, does not lead to better results on the sentiment main task.

We also consider that the poor performance of the transfer learning approach may be due to overfitting to the training data, which only contains negated examples. However, further experiments training with balanced data (50% negated and 50% non-negated) give poorer performance overall (85.3 F_1 for both STL and MTL scope-level), indicating that this is not the root of the problem.

6.7 Comparing negation detection to other common auxiliary tasks

In MTL for natural language processing, it is common to employ a number of auxiliary tasks, which range from simple tasks, (predicting word frequency), to morphosyntactic tasks (chunking, dependency relation classification), to semantic tasks (semantic frame detection, super-sense tagging). In this section, we compare common auxiliary tasks and their effect on sentiment analysis. Specifically, we train the MTL model from Section 4 on three additional auxiliary tasks: POS tagging, multi-word detection (MWE) (identifying multi-word expressions, i.e. *by the way*, *cope with*), and super-sense tagging (SEM) (assigning course-grained semantic types to verbs and nouns).

The data for the auxiliary tasks come from the STREUSLE data set (Schneider and Smith 2015), which contains sentences from the REVIEW section of the English Web Treebank (Bies *et al.* 2012),

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Table 10. Train and test splits available for each auxiliary task, as well as label entropy and kurtosis

Data set	# Train	# Test	Label entropy	Label kurtosis
SFU	800	96	0.21	0.2
ConanDoyle-neg	842	235	1.02	−0.8
POS tagging	2451	272	3.04	2.8
Multi-word detection	2451	272	0.64	29.8
Super-sense tagging	2451	272	2.16	82.7

Table 11. Accuracy on SST-fine with POS tagging, multi-word detection (MWE), and super-sense tagging (SEM) as auxiliary tasks. Bold results show the best performing auxiliary task.

Data set	STL	Negation	POS	MWE	SEM
SST-fine	46.49 (0.7)	**47.33 (0.6)**	46.44 (0.9)	46.93 (0.6)	46.11 (0.5)

which have been enriched with multi-word and super-sense annotations. Table 10 shows the statistics of the data sets, as well as the entropy and kurtosis of the labels. Here, entropy indicates the amount of uncertainty in the label distribution, while kurtosis indicates the skewness. These measures have been shown to correlate well with the usefulness of auxiliary tasks in previous work (Martínez Alonso and Plank 2017; Bingel and Søgaard 2017).

The results are shown in Table 11. POS tagging and super-sense tagging perform worse than the baseline single-task model, while multi-word detection and negation detection show improvements. The MTL negation model, however, is still the best performing model, which demonstrates the importance of negation on sentiment classification. The fact that multi-word detection is helpful may correlate with the importance of multi-word idioms in expressions of sentiment (Williams *et al*. 2015; Liu *et al*. 2017; Jochim *et al*. 2018; Barnes *et al*. 2019a).

Both the SFU and CD data sets not only have low label kurtosis but also have relatively low label entropy. This partially aligns with previous research (Martínez Alonso and Plank 2017; Bingel and Søgaard 2017), which suggests that for an auxiliary task to improve the main task, the entropy of the labels should be high (implying the task should not be trivial to learn), and the kurtosis should be low (the labels should not have an overly long-tailed distribution). The fact that the multi-word task is more helpful, however, seems to indicate that the appropriateness of the auxiliary task for the main task is more important than the specific data set properties.

7. Conclusion and future work

This paper introduces a MTL approach to incorporating explicitly annotated negation information into a sentiment classifier. We employ a cascading architecture where one BiLSTM is shared between the sentiment and negation tasks and feeds into a higher-level BiLSTM dedicated only to sentiment prediction (also using skip connections). We show that using negation as an auxiliary task helps improve the main task of sentiment analysis and that the effect persists across several different standard data sets. While we only report results for English here, for future work, we plan to extend the experiments to other languages that have annotations for both tasks available, for example, Spanish.

The extensive analysis of the results reveals several effects of using negation detection as an auxiliary task. On the one hand, we find that even a small amount of annotated negation data allows

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

a multi-task learner to improve, while on the other hand, it is necessary to have enough sentiment data to achieve relatively good performance in order to see improvements. We further show that detection of both negation cues and scopes as an auxiliary task is preferable over detecting only one of these.

In this work, negation cues were always modeled on the token level, but morphological negation is another important realization of negation that our current model does not fully take into account. Adding character-level information to the network could be of interest in the future. Moreover, it may also be useful to separate the cue and scope classification, in order to improve the negation module.

We have noted several places that the two data sets for negation employed in this work operate with slightly different annotation schemes. Due to the fact that these data sets are taken from different domains and genres, it has not been possible for us to compare the effect of these differing annotation choices systematically. Another avenue for future work would therefore be to compare the effect of different annotation schemes for negation by comparing the use of the ConanDoyle-neg and the re-annotated version of this data set dubbed NegPar (Liu, Fancellu, and Webber 2018) in our multi-task setup for sentiment analysis.

Regarding MTL, we demonstrate that it is possible to use an auxiliary task with different labeling units (token-level sequence labeling) to improve the main task (sentence-level classification). Additionally, we show that negation detection is a more suitable auxiliary task for sentiment analysis than other standard auxiliary tasks, such as POS tagging, multi-word detection, or super-sense tagging. Finally, our experiments on transfer learning indicate that MTL may provide a better framework to leverage negation information, but other approaches to transfer learning, such as freeze-thaw (Felbo *et al*. 2017) or discriminative fine-tuning (Howard and Ruder 2018) may give better results. We also want to explore the combination of MTL and transfer learning (i.e. continued training of pre-trained negation layers in an MTL setup).

Although we only experiment with negation in this work, there are many other linguistic and paralinguistic phenomena, that is speculation, multi-word expressions, sarcasm, etc., which also affect sentiment classification (Cruz *et al*. 2016; Farias and Rosso 2017; Barnes *et al*. 2019a). Here, we have shown that explicit training via hierarchical MTL is a viable way to incorporate some of this information. In the future, we would like to incorporate other sources of linguistic knowledge in a similar fashion.

References

Ambartsoumian A. and Popowich F. (2018). Self-attention: A better building block for sentiment analysis neural network classifiers. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Brussels, Belgium, pp. 130–139.

Augenstein I., Ruder S. and Søgaard A. (2018). Multi-task learning of pairwise sequence classification tasks over disparate label spaces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, USA, pp. 1896–1906.

Augenstein I. and Søgaard A. (2017). Multi-task learning of keyphrase boundary classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, pp. 341–346.

Barnes J., Klinger R. and **Schulte im Walde S.** (2017). Assessing state-of-the-art sentiment models on state-of-the-art sentiment datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Copenhagen, Denmark, pp. 2–12.

Barnes J., Øvrelid L. and Velldal E. (2019a). Sentiment analysis is not solved! Assessing and probing sentiment classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy, pp. 12–23.

Barnes J., Touileb S., Øvrelid L. and Velldal E. (2019b). Lexicon information in neural sentiment analysis: A multi-task learning approach. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, Turku, Finland.

Bies A., Mott J., Warner C. and Kulick S. (2012). English web treebank. In *Technical Report LDC2012T13, Linguistic Data Consortium*, Philidelphia, PA, USA.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Bingel J. and Søgaard A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, pp. 164–169.

Bjerva J. (2017). Will my auxiliary tagging task help? Estimating auxiliary tasks effectivity in multi-task learning. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, Gothenburg, Sweden, pp. 216–220.

Caruana R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, pp. 41–48.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K. and Kuksa P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **12**, 2493–2537.

Councill I., McDonald R. and Velikovich L. (2010). What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, Uppsala, Sweden, pp. 51–59.

Cruz N.P., Taboada M. and Mitkov R. (2016). A machine-learning approach to negation and speculation detection for sentiment analysis. *Journal of the Association for Information Science and Technology* **67**(9), 2118–2136.

Das S.R. and Chen M.Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science* **53**(9), 1375–1388.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.

Enger M., Velldal E. and Øvrelid L. (2017). An open-source tool for negation detection: A maximum-margin approach. In *Proceedings of the EACL workshop on Computational Semantics Beyond Events and Roles (SemBEaR)*, Valencia, Spain, pp. 64–69.

Fancellu F., Lopez A. and Webber B. (2016). Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pp. 495–504.

Fancellu F., Lopez A., Webber B. and He H. (2017). Detecting negation scope is easy, except when it isn't. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, pp. 58–63.

Fares M., Oepen S. and Velldal E. (2018). Transfer and multi-task learning for noun–noun compound interpretation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 1488–1498.

Farias D.H. and Rosso P. (2017). Irony, sarcasm, and sentiment analysis. In Pozzi F.A., Fersini E., Messina E. and Liu B. (eds), *Sentiment Analysis in Social Networks*, Chapter 7. Boston, USA: Morgan Kaufmann, pp. 113–128.

Felbo B., Mislove A., Søgaard A., Rahwan I. and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 1615–1625.

Goldberg Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies* **10**(1), 1–309.

Graves A. and Schmidhuber J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* **18**(5), 602–610. IJCNN 2005.

Howard J. and Ruder S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, pp. 328–339.

Hu M. and Liu B. (2004). Mining opinion features in customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, USA, pp. 168–177.

Irsoy O. and Cardie C. (2014). Deep recursive neural networks for compositionality in language. In Ghahramani Z., Welling M., Cortes C., Lawrence N.D. and Weinberger K.Q. (eds), *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., pp. 2096–2104

Iyyer M., Manjunatha V., Boyd-Graber J. and Daume III H. (2015). Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China, pp. 1681–1691.

Jochim C., Bonin F., Bar-Haim R. and Slonim N. (2018). SLIDE – a sentiment lexicon of common idioms. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan, pp. 2387–2392.

Kennedy A. and Inkpen D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence* **22**(2), 110–125.

Kingma D. and Ba J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Konstantinova N., de Sousa S.C., Cruz N.P., Maña M.J., Taboada M. and Mitkov R. (2012). A review corpus annotated for negation, speculation and their scope. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, Istanbul, Turkey, pp. 3190–3195.

Kshirsagar M., Thomson S., Schneider N., Carbonell J., Smith N.A. and Dyer C. (2015). Frame-semantic role labeling with heterogeneous annotations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China, pp. 218–224.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Lapponi E., Read J. and Øvrelid L. (2012a). Representing and resolving negation for sentiment analysis. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops*, Washington, DC, USA, pp. 687–692.

Lapponi E., Velldal E., Øvrelid L. and Read J. (2012b). UiO2: Sequence-labeling negation using dependency features. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, Montreal, Canada, pp. 319–327.

Lei Z., Yang Y., Yang M. and Liu Y. (2018). A multi-sentiment-resource enhanced attention network for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, pp. 758–763.

Liu P., Qian K., Qiu X. and Huang X. (2017). Idiom-aware compositional distributed semantics. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 1204–1213.

Liu P., Qiu X. and Huang X. (2016). Recurrent neural network for text classification with multi-task learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York, USA, pp. 2873–2879.

Liu Q., Fancellu F. and Webber B. (2018). NegPar: A parallel corpus annotated for negation. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, Miyazaki, Japan, pp. 3464–3472.

Martínez Alonso H. and Plank B. (2017). When is multitask learning effective? Semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, pp. 44–53.

Morante R. and Blanco E. (2012). *SEM 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, Montréal, Canada, pp. 265–274.

Morante R. and Daelemans W. (2009). A metalearning approach to processing the scope of negation. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, Boulder, USA.

Morante R. and Daelemans W. (2012). ConanDoyle-neg: Annotation of negation cues and their scope in Conan Doyle stories. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, Istanbul, Turkey.

Morante R., Liekens A. and Daelemans W. (2008). Learning the scope of negation in biomedical texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Waikiki, Hawaii.

Nakov P., Rosenthal S., Kozareva Z., Stoyanov V., Ritter A. and Wilson T. (2013). Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval)*.

Packard W., Bender E.M., Read J., Oepen S. and Dridan R. (2014). Simple negation scope resolution through deep parsing: A semantic solution to a semantic problem. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, USA.

Padó S. (2006). User's guide to sigf: Significance testing by approximate randomisation. https://nlpado.de/sebastian/software/sigf.shtml.

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

Pang B., Lee L. and Vaithyanathan S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, Philadelphia, USA, pp. 79–86.

Peng N. and Dredze M. (2017). Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, Vancouver, Canada, pp. 91–100.

Peters M., Neumann M., Iyyer M., Gardner M., Clark C., Lee K. and Zettlemoyer L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, USA, pp. 2227–2237.

Plank B. (2016). Keystroke dynamics as signal for shallow syntactic parsing. In *Proceedings of the 26th International Conference on Computational Linguistics*, Osaka, Japan, pp. 609–619.

Polanyi L. and Zaenen A. (2006). *Contextual Valence Shifters*. Netherlands, Dordrecht: Springer, pp. 1–10.

Qian Z., Li P., Zhu Q., Zhou G., Luo Z. and Luo W. (2016). Speculation and negation scope detection via convolutional neural networks. In *The 2016 Conference on Empirical Methods in Natural Language Processing*.

Read J., Velldal E., Øvrelid L. and Oepen S. (2012). UiO1: Constituent-based discriminative ranking for negation resolution. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, Montreal, Canada.

Ruder S., Bingel J., Augenstein I. and Søgaard A. (2019). Latent multi-task architecture learning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, Honolulu, USA.

Sanh V., Wolf T. and Ruder S. (2019). A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, Honolulu, USA.

Schneider N. and Smith N.A. (2015). A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, pp. 1537–1547.

Socher R., Perelygin A., Wu J., Chuang J., Manning C., Ng A. and Potts C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, USA, pp. 1631–1642.

Søgaard A. and Goldberg Y. (2016). Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pp. 231–235.

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press

Taboada M., Brooke J., Tofiloski M., Voll K. and Stede M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics* **37**(2), 267–307.

Tai K.S., Socher R. and Manning C.D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China, pp. 1556–1566.

Tang D., Wei F., Yang N., Zhou M., Liu T. and Qin B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1555–1565.

Turney P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, USA.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L.u. and Polosukhin I. (2017). Attention is all you need. In Guyon I., Luxburg U.V., Bengio S., Wallach H., Fergus R., Vishwanathan S. and Garnett R. (eds), *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 5998–6008.

Velldal E., Øvrelid L., Read J. and Oepen S. (2012). Speculation and negation: Rules, rankers, and the role of syntax. *Computational Linguistics* **38**(2), 369–410.

Verma R., Kim S. and Walter D. (2018). Syntactical analysis of the weaknesses of sentiment analyzers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 1122–1127.

Vincze V., Szarvas G., Farkas R., Móra G. and Csirik J. (2008). The BioScope corpus: Biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, Suppl 11.

White J. (2012). UWashington: Negation resolution using machine learning methods. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, Montreal, Canada.

Wiegand M., Balahur A., Roth B., Klakow D. and Montoyo A. (2010). A survey on the role of negation in sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, Uppsala, Sweden, pp. 60–68.

Williams L., Bannister C., Arribas-Ayllon M., Preece A. and Spasić I. (2015). The role of idioms in sentiment analysis. *Expert Systems with Applications* **42**(21), 7375–7385.

Xu H., Liu B., Shu L. and Yu P.S. (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*, Minneapolis, USA.

Yeh A. (2000). More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational Linguistics*, Saarbrücken, Germany, pp. 947–953.

Zhu X., Guo H., Mohammad S. and Kiritchenko S. (2014). An empirical study on the effect of negation words on sentiment. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, Maryland, pp. 304–313.

Cite this article: Barnes J, Velldal E and Øvrelid L (2021). Improving sentiment analysis with multi-task learning of negation. *Natural Language Engineering* **27**, 249–269. https://doi.org/10.1017/S1351324920000510

https://doi.org/10.1017/S1351324920000510 Published online by Cambridge University Press