

Fine-grained Sentiment Analysis as Dependency Graph Parsing

Anonymous ACL-IJCNLP submission

Abstract

Fine-grained sentiment analysis attempts to extract full opinion tuples from a text, but over time this task has been subdivided into smaller and smaller sub-tasks, *e.g.* target extraction or targeted polarity classification. We argue that this division has become counterproductive and propose a new unified framework to remedy the situation. We cast the fine-grained sentiment problem as dependency graph parsing, where the nodes are spans of sentiment holders, targets and expressions, and the arcs are the relations between them. We perform experiments on five datasets in four languages (English, Norwegian, Basque, and Catalan) and show that this approach leads to strong improvements over state-of-the-art baselines. Our analysis shows that refining the sentiment graphs with syntactic dependency information further improves results.

1 Introduction

Fine-grained sentiment analysis can be theoretically cast as an information extraction problem in which one attempts to find all of the opinion tuples $O = O_1, \dots, O_n$ in a text. Each opinion O_i is a tuple (h, t, e, p) where h is a **holder** who expresses a **polarity** p towards a **target** t through a **sentiment expression** e , implicitly defining pairwise relationships between elements of the same tuple. Liu (2012) argues that all of these elements¹ are essential to fully resolve the sentiment analysis problem. However, most research on sentiment analysis focuses either on a variety of sub-tasks, which avoids performing the full task, or on simplified and idealized tasks, *e.g.* sentence-level binary polarity classification.

We argue that the division of fine-grained sentiment into these sub-tasks has become counterproductive, as reported experiments are often not

sensitive to whether a given addition to the pipeline improves the overall resolution of sentiment, or do not take into account the inter-dependencies of the various sub-tasks. As such, we propose a unified approach to *fine-grained sentiment* which jointly predicts all elements of an opinion tuple and their relations. Moreover, we cast sentiment analysis as a *dependency graph parsing problem*, where the sentiment phrase is the root node, and the other elements have arcs which model the relationships between them. This methodology also enables us to take advantage of recent improvements in semantic dependency parsing (Dozat and Manning, 2018; Oepen et al., 2020; Kurtz et al., 2020) to efficiently learn a sentiment graph parser.

This perspective also allows us to unify a number of approaches, including targeted, and opinion tuple mining. We aim to answer **RQ1**: whether graph-based approaches to fine-grained sentiment outperform state-of-the-art sequence labeling approaches, and **RQ2**: how to best encode fine-grained sentiment as parsing graphs. We perform experiments on five standard datasets in four languages (English, Norwegian, Basque, Catalan) and show that graph-based approaches outperform state-of-the-art baselines on all datasets on several standard metrics, as well as our proposed novel (unlabeled and labeled) sentiment graph metrics. We further propose methods to inject linguistic structure into the sentiment graphs using syntactic dependencies. Our main contributions are therefore 1) proposing a holistic approach to fine-grained sentiment through sentiment graph parsing, 2) introducing new evaluation metrics for measuring model performance, and 3) extensive experimental results that outperform state-of-the-art baselines. Finally, we release the code and datasets² to enable future work on this problem.

¹Liu (2012)’s definition replaces sentiment expression with the time when the opinion was expressed.

²www.blinded.for.review

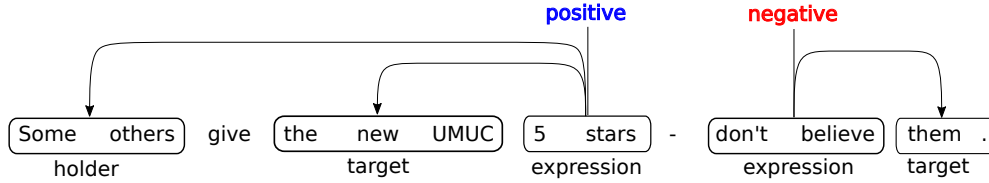


Figure 1: A fine-grained sentiment graph is composed of a holder, target, sentiment expression, their relationships and a polarity attribute. Holders and targets can be null.

2 Related Work

Fine-grained sentiment analysis can be broken down into five sub-tasks: i) sentiment expression extraction, ii) sentiment target extraction, iii) sentiment holder extraction, iv) defining the relationship between these elements, and v) assigning polarity. Previous work on information extraction has used pipeline methods which first extract the holders, targets, and expressions (tasks i - iii) and subsequently predict their relations (task iv), mostly on the MPQA dataset (Wiebe et al., 2005). CRFs and a number of external resources (sentiment lexicons, dependency parsers, named-entity taggers) (Choi et al., 2006; Yang and Cardie, 2012) are strong baselines. Given the small size of the training data and the complicated task, these techniques often still outperform neural models, such as BiLSTMs (Katiyar and Cardie, 2016). Transition-based end-to-end approaches have shown some potential (Zhang et al., 2019). However, all of this work ignores the polarity classification subtask.

Targeted sentiment analysis only concentrates on extracting sentiment targets (task ii) and classifying the polarity directed towards them (task iv) (Jiang et al., 2011; Mitchell et al., 2013). Recent shared tasks on *Aspect-Based Sentiment Analysis* (ABSA) (Pontiki et al., 2014, 2015, 2016) also include target extraction and polarity classification subtasks. Joint approaches perform on par with pipeline methods (Li et al., 2019a) and multitask models can perform even better (He et al., 2019). Finally, pretrained language models (Devlin et al., 2019) can also lead to improvements on the ABSA data (Li et al., 2019b).

End2End sentiment analysis is a recently proposed subtask which combines targeted sentiment (tasks ii and v) and sentiment expression extraction (task i), without requiring the resolution of relationships between targets and expressions. Wang et al. (2016) augment the ABSA datasets with sentiment expressions, but provide no details on the annota-

tion process or any inter-annotator agreement. He et al. (2019) make use of this data and propose a multi-layer CNN (IMN) to create hidden representations h which are then fed to a target and opinion extraction module (AE), which is also a multi-layer CNN. This module predicts \hat{y}^{ae} , a sequence of BIO tags³ that predict the presence or absence of targets and expressions. After jointly predicting the targets and expressions, a second multi-layer CNN with a final self-attention network is used to classify the polarity, again as sequence labeling task (AS). This second module combines the information from h and \hat{y}^{ae} by incorporating the predicted probability of a token to be a target in the formulation of self-attention. Finally, an iterative message-passing algorithm updates h using the predictions from all the modules at the previous timestep.

Chen and Qian (2020) instead propose Relation-Aware Collaborative Learning (RACL). This model creates task specific representations by first embedding a sentence, passing through a shared feed-forward network and finally a task-specific CNN. This approach then models interactions between each pair of sub-tasks (target extraction, expression extraction, sentiment classification) by creating pairwise weighted attention representations. These are then concatenated and used to create the task-specific predictions. The authors finally stack several RACL layers, using the output from the previous layer as input for the next.

Both models perform well on the augmented SemEval data, but it is unlikely that these annotations are adequate for full fine-grained sentiment, as Wang et al. (2016) only provide expression annotations for sentences that have targets, generally only include sentiment-bearing words (not phrases), and do not specify the relationship between target and expression.

Graph parsing: Syntactic dependency graphs are regularly used in applications, supplying them

³The tags include {BIO}-{target, expression}

		sentences		holders			targets			expressions			polarity		
		#	avg.	#	avg.	max	#	avg.	max	#	avg.	max	+	neu	-
NoReC_{Fine}	train	8634	16.7	898	1.1	12	6778	1.9	35	8448	4.9	40	5684	0	2756
	dev	1531	16.9	120	1.0	3	1152	2.0	15	1432	5.1	31	988	0	443
	test	1272	17.2	110	1.0	3	993	2.0	20	1235	4.9	30	875	0	358
MultiB_{CA}	train	1174	15.6	169	1.1	4	1695	2.4	18	1981	2.6	19	1272	0	708
	dev	168	13.3	15	1.5	7	211	2.3	10	258	2.6	9	151	0	107
	test	336	14.7	52	1.1	5	430	2.6	12	518	2.7	14	313	0	204
MultiB_{EU}	train	1064	10.5	205	1.1	6	1285	1.4	9	1684	2.2	10	1406	0	278
	dev	152	10.7	33	1.1	2	153	1.3	6	204	2.5	8	168	0	36
	test	305	10.7	58	1.1	2	337	1.4	8	440	2.2	9	375	0	65
MPQA	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_{Unis}	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

Table 1: Statistics of the datasets, including number of sentences and average length (in tokens) per split, 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.

with necessary grammatical information (Mintz et al., 2009; Cui et al., 2005; Björne et al., 2009; Johansson and Moschitti, 2012; Lapponi et al., 2012). The dependency graph structures used in these systems are predominantly restricted to trees. While trees are sufficient to encode syntactic dependencies, they are not expressive enough to handle *meaning representations*, that require nodes to have multiple incoming arcs, or having no incoming arcs at all (Kuhlmann and Oepen, 2016). While much of the early research on parsing these new structures (Oepen et al., 2014, 2015) focused on specialized decoding algorithms, Dozat and Manning (2018) presented a neural dependency parser that essentially relies only on its neural network structure to predict any type of dependency graph without restrictions to certain structures. Using the parser’s ability to learn arbitrary dependency graphs, Kurtz et al. (2020) phrased the task of negation resolution (Morante and Blanco, 2012; Morante and Daelemans, 2012) as a graph parsing task. This transformed the otherwise flat representations to dependency structures that directly encode the often overlapping relations between the building blocks of multiple negation instances at the same time. In a simpler fashion, Yu et al. (2020) exploit the parser of Dozat and Manning (2018) to predict spans of named entities.

3 Datasets

We here focus on datasets that annotate the full task of fine-grained sentiment as described initially. We perform experiments on five fine-grained datasets in four languages, the statistics of which are shown in Table 1. The largest available fine-grained dataset is the **NoReC_{Fine}** dataset (Øvrelid et al., 2020), a multi-domain dataset of professional reviews in Norwegian, annotated for fine-grained sentiment. **MultiB_{EU}** and **MultiB_{CA}** (Barnes et al., 2018) are hotel reviews in Basque and Catalan, respectively. **MPQA** (Wiebe et al., 2005) annotates news wire text in English. Finally, **DS_{Unis}** (Toprak et al., 2010) annotate English reviews of online universities and e-commerce. In our experiments, we use only the university reviews, as the e-commerce reviews have a large number of ‘polar targets’, *i.e.* targets with a polarity, but no accompanying sentiment expression.

While all the datasets annotate holders, targets, and expressions, the frequency and distribution of these vary. Regarding holders, **MPQA** has the most (2,054) and **DS_{Unis}** has the fewest (94), whereas **NoReC_{Fine}** has the largest proportion of targets (8,923) and expressions (11,115). The average length of holders (2.6 tokens) and targets (6.1 tokens) in **MPQA** is also considerably higher than the others.

It is also worth pointing out that **MPQA** and **DS_{Unis}** additionally include neutral polarity. In the case of **MPQA** the neutral class refers to verbs which are subjective but do not convey polarity, *e.g.* ‘say’, ‘opt for’. In **DS_{Unis}**, however, the neutral label tends to indicate expressions that could entail mixed polarity or are polar under the right conditions, *e.g.* ‘the classes were not easy’ is considered neutral, as it is possible for difficult classes to be desirable at a university. **MultiB_{EU}**, and **MultiB_{CA}** also have labels for strong positive and strong negative, which we map to positive and negative, respectively. Finally, **NoReC_{Fine}** includes intensity annotations (strong, normal, slight), which we disregard for the purposes of these experiments.

4 Modeling

This section describes how we define and encode sentiment graphs, detail the neural dependency graph models, as well as two state-of-the-art baselines for end-to-end sentiment analysis (target and expression extraction, plus polarity classification).

4.1 Graph Representations

Fine-grained sentiment graphs as in Figure 1 are directed graphs, that are made up of a set of labeled nodes and a set of unlabeled edges connecting pairs of nodes. Nodes in the fine-grained sentiment graphs can span over multiple tokens and may have multiple incoming edges. The resulting graphs can have multiple entry points (*roots*), are not necessarily connected, and not every token is a node in the graph. The sentence’s sentiment expressions correspond to the roots of the graphs, connecting explicitly to their respective holders and targets. In order to apply the algorithm of Dozat and Manning (2018), we simplify these structures into bi-lexical dependency graphs visualized in Figure 2. Here, nodes correspond one-to-one to the tokens of the sequence and follow the same linear order. The edges are drawn as arcs in the half-plane above the sentence, connecting *heads* to *dependents*. Similarly to the source structures, the graphs can have multiple roots and nodes can have multiple or no incoming arcs. For some rare instances of fine-grained sentiment graphs, the reduction to dependency graphs is lossy, as they do not allow multiple arcs to share the same head and dependent. This results in a slight mismatch of the learned and aimed-for representations.

Metric Name	Level	Strictness	+/-
Holder F₁	Token-level	Partial	No
Target F₁	Token-level	Partial	No
Exp. F₁	Token-level	Partial	No
Targeted F₁	Token-level	Exact	Yes
UF₁	Graph arcs	Exact	No
LF₁	Graph arcs	Exact	Yes
NSF₁	Sentiment-graph	Exact graph, partial token	No
SF₁	Sentiment-graph	Exact graph, partial token	Yes

Table 2: Metrics used to evaluate performance. Column +/- indicates whether polarity is included or not.

The choice of how to encode the sentiment graphs as parsing graphs opens for several alternate representations depending on the choice of head/dependent status of individual tokens in the target/holder/expression spans of the sentiment graph. We here propose two simple parsing graph representations: head-first and head-final, which are shown in Figure 2. For **head-first**, we set the first token of the sentiment expression as a root node, and similarly set the first token in each holder and token span as the head of the span with all other tokens within that span as dependents. The labels simply denote the type of relation (target/holder) and for sentiment expressions, additionally encode the polarity. **Head-final** is similar, but instead sets the final token of spans as the heads, and the final token of the sentiment expression as the root node.

4.2 Proposed model

The neural graph parsing model used in this work is a reimplementation of the neural parser by Dozat and Manning (2018) which was used by Kurtz et al. (2020) for negation resolution. The parser learns to score each possible arc to then finally predict the output structure simply as a collection of all positively scored arcs. The base of the network structure is a bidirectional LSTM (BiLSTM), that processes the input sentence both from left-to-right and right-to-left, to create contextualized representations $c_1, \dots, c_n = \text{BiLSTM}(w_1, \dots, w_n)$ where w_i is the concatenation of a word embedding, POS tag embedding, lemma embedding, and character embedding created by a character-based LSTM for the i th token. In our experiments, we further augment the token representations with pre-trained contextualized embeddings from multilingual BERT (Xu et al., 2019). We use multilingual BERT as several languages did not have available

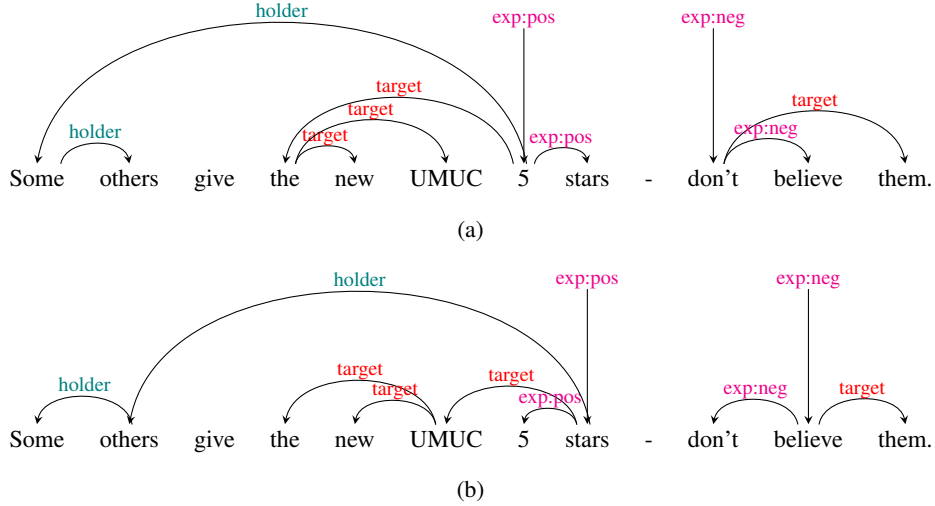


Figure 2: Two parsing graph proposals to encode the sentiment graph: (a) **head-first**, where the first token of any span is the head, and (b) **head-final**, where the final token is the head.

monolingual BERT models at the time of the experiments (Catalan, Norwegian).

The contextualized embeddings are then processed by two feedforward neural networks (FNN), creating specialized representations for potential heads and dependents, $h_i = \text{FNN}_{\text{head}}(c_i)$ and $d_i = \text{FNN}_{\text{dep}}(c_i)$. The scores for each possible arc-label combination are computed by a final bilinear transformation using the tensor U . Its inner dimension corresponds to the number of sentiment graph labels plus a special NONE label, indicating the absence of an arc, which allows the model to predict arcs and labels jointly, $\text{score}(h_i, d_j) = h_i^\top U d_j$.

4.3 Baselines

We compare our proposed graph prediction approach with three state-of-the-art baselines⁴ for extracting targets and expressions and predicting the polarity: **IMN**⁵, **RACL**⁶, as well as **RACL-BERT**, which also incorporates contextualized embeddings. Instead of using $\text{BERT}_{\text{Large}}$, we use the cased BERT-multilingual-base in order to fairly compare with our own models. Note, however, that our model does not update the mBERT representations, putting it at a disadvantage to RACL-BERT.

⁴Despite having state-of-the-art results on **MPQA**, we do not compare with Katiyar and Cardie (2016) as they use different dataset splits, 10-fold cross-validation, and their code is not available.

⁵IMN code available at <https://github.com/ruidan/IMN-E2E-ABSA>.

⁶<https://github.com/NLPWM-WHU/RACL>.

5 Evaluation

As we are interested not only in extraction or classification, but rather in the full fine-grained sentiment task, we propose metrics that capture the relations between all predicted elements, while enabling comparison with previous state-of-the-art models which only perform part of the full task.

Token-level F_1 for Holders, Targets, and Expressions To easily compare our models to pipeline models, we evaluate how well these models are able to identify the elements of a sentiment graph with token-level F_1 .

Targeted F_1 This is a common metric in targeted sentiment analysis (also referred to as F_{1-i} (He et al., 2019) or ABSA F_1 (Chen and Qian, 2020)). A true positive requires the combination of exact extraction of the sentiment target, and the correct polarity.

Parsing graph metrics We additionally compute graph-level metrics to determine how well the models predict the unlabeled and labeled arcs of the parsing graphs: Unlabeled F_1 (UF_1), Labeled F_1 (LF_1). These measure the amount of (in)correctly predicted arcs and labels, as the harmonic mean of precision and recall (Oepen et al., 2014). These metrics inform us of the local properties of the graph, and do not overly penalize a model if a few edges of a graph are incorrect.

Sentiment graph metrics The main metrics are Non-polar Sentiment Graph F_1 (NSF_1) and Sen-

Dataset	Model	Spans			Targeted F ₁	Parsing Graph		Sent. Graph	
		Holder F ₁	Target F ₁	Exp. F ₁		UF ₁	LF ₁	NSF ₁	SF ₁
NoReC _{Fine}	IMN	-	35.9	48.7	18.0	-	-	-	-
	RACL	-	45.6	55.4	20.1	-	-	-	-
	RACL-BERT	-	47.2	<u>56.3</u>	30.3	-	-	-	-
	Head-first	51.1	50.1	54.4	30.5	39.2	31.5	37.0	29.5
	Head-final	<u>60.4*</u>	<u>54.8</u>	55.5	<u>31.9</u>	<u>48.0*</u>	<u>37.7*</u>	<u>39.2*</u>	<u>31.2*</u>
MultiB _{EU}	IMN	-	48.2	65.2	39.5	-	-	-	-
	RACL	-	55.4	70.7	48.2	-	-	-	-
	RACL-BERT	-	59.9	72.6	56.8	-	-	-	-
	Head-first	60.4	<u>64.0</u>	<u>73.9</u>	<u>57.8</u>	<u>64.6</u>	<u>60.0</u>	<u>58.0</u>	<u>54.7</u>
	Head-final	<u>60.5</u>	<u>64.0</u>	72.1	56.9	60.8	56.0	<u>58.0</u>	<u>54.7</u>
MultiB _{CA}	IMN	-	56.3	60.9	32.5	-	-	-	-
	RACL	-	65.4	67.6	49.1	-	-	-	-
	RACL-BERT	-	67.5	70.3	52.4	-	-	-	-
	Head-first	<u>43.0</u>	<u>72.5</u>	<u>71.1*</u>	<u>55.0*</u>	<u>66.8*</u>	<u>62.1*</u>	<u>62.0</u>	<u>56.8</u>
	Head-final	37.1	71.2	67.1	53.9	62.7	58.1	59.7	53.7
MPQA	IMN	-	24.3	29.6	1.2	-	-	-	-
	RACL	-	32.6	37.8	11.8	-	-	-	-
	RACL-BERT	-	20.0	31.2	17.8	-	-	-	-
	Head-first	43.8	<u>51.0</u>	<u>48.1</u>	<u>33.5*</u>	40.0	36.9	24.5	17.4
	Head-final	<u>46.3</u>	49.5	46.0	18.6	<u>41.4</u>	<u>38.0</u>	<u>26.1</u>	<u>18.8</u>
DS _{Unis}	IMN	-	33.0	27.4	17.9	-	-	-	-
	RACL	-	39.3	40.2	22.8	-	-	-	-
	RACL-BERT	-	<u>44.6</u>	38.2	27.3	-	-	-	-
	Head-first	28.0	39.9	40.3	26.7	35.3	31.4	31.0	25.0
	Head-final	<u>37.4</u>	42.1	<u>45.5*</u>	<u>29.6</u>	<u>38.1</u>	<u>33.9</u>	<u>34.3*</u>	<u>26.5</u>

Table 3: Experiments comparing our sentiment graph approaches (Head-first/Head-final) using mBERT with the sequence-labeling baselines (IMN, RACL, RACL-BERT). **Underlined** numbers indicate the best result for the metric and dataset. * indicates approach is significantly better than second best ($p < 0.05$), as determined by a bootstrap with replacement test.

timent Graph F₁ (SF₁), which attempt to capture how well a model is able to capture the full sentiment graph (see Figure 1). For NSF₁, each sentiment graph is a tuple of (holder, target, expression), while for SF₁ we include polarity (holder, target, expression, polarity). A true positive is defined as an exact match at graph-level, weighting the overlap in predicted and gold spans for each element, averaged across all three spans. For precision we weight the number of correctly predicted tokens divided by the total number of predicted tokens (for recall, we divide instead by the number of gold tokens). We allow for empty holders and targets.

6 Experiments

All sentiment graph models use token-level mBERT representations in addition to word2vec skip-gram embeddings openly available from the NLPL vector repository⁷ (Fares et al., 2017). We

⁷Nordic Language Processing Laboratory vector repo.: <http://vectors.nlpl.eu/repository/>. We used

train all models for 100 epochs and keep the model that performs best regarding LF₁ on the dev set (Targeted F₁ for the baselines). We use default hyperparameters from Kurtz et al. (2020) (see Appendix) and run all of our models five times with different random seeds and report the mean (standard deviation shown as well in Table 7 in the Appendix). We calculate statistical difference between the best and second best models through a bootstrap with replacement test (Berg-Kirkpatrick et al., 2012). As there are 5 runs, we require that 3 of 5 be statistically significant at $p < 0.05$. Table 3 shows the results for all datasets.

On NoReC_{Fine}, the baselines IMN, RACL, and RACL-BERT perform well at extracting targets (35.9, 45.6, and 47.2 F₁, respectively) and expressions (48.7/55.4/56.3), but struggle with the full tar-

300-dimensional embeddings trained on English Wikipedia and Gigaword for English (model id 18 in the repo.), and 100-dimensional embeddings trained on the 2017 CoNLL corpora for all others; Basque (id 32), Catalan (id 34), and Norwegian Bokmål (id 58).

geted sentiment task (18.0/20.1/30.3). The graph-based models extract targets better (50.1/54.8) and have comparable scores for expressions (54.4/55.5). The holder extraction scores have a similar range (51.1/60.4). These patterns hold throughout the other datasets, where the proposed graph models nearly always perform best on extracting spans, although RACL-BERT achieves the best score on extracting targets on **DS_{Unis}** (44.6 vs. 42.1). The graph models also outperform the strongest baseline (RACL-BERT) on *targeted sentiment* on all 5 datasets, although this difference is often not statistically significant (**NoReC_{Fine}** Head-first, **MultiB_{EU}** Head-final) and RACL-BERT is better than Head-first on **DS_{Unis}**.

Regarding the Graph metrics, the results depend highly on the dataset, with UF_1 and LF_1 ranging from 35.3/31.4 (**DS_{Unis}** Head-first) to 66.8/62.1 (**MultiB_{CA}** Head-first). Sentiment Graph metrics NSF_1 and SF_1 have a similar, though slightly lower range (24.5/17.7 – 62.0/56.8). The graph and sentiment graph metrics do not correlate perfectly, however, as UF_1 and LF_1 on **MPQA** are relatively good (40.0/36.9 and 41.4/38.0 for Head-first and Head-final, respectively), but the NSF_1 and SF_1 are poor (24.5/17.4 and 26.1/18.8).

On average IMN is the weakest baseline, followed by RACL and then RACL-BERT. The main improvement that RACL-BERT gives over RACL on these datasets is seen in the Targeted metric, *i.e.* the contextualized representations improve the polarity classification more than the extraction task. The proposed graph-based models are consistently the best models across the metrics and datasets.

Regarding graph representations, the differences between Head-first and Head-final are generally quite small. Head-first performs better on **MultiB_{CA}** and slightly better on **MultiB_{EU}**, while for the others (**NoReC_{Fine}**, **MPQA**, and **DS_{Unis}**) Head-final is better. This suggests that the main benefit is the joint prediction of all spans and relationships, and that the specific graph representation matters less.

7 Analysis

In this section we perform a deeper analysis of the models in order to answer the research questions.

	#	H.first	H.final	RACL
NoReC_{Fine}	147	63.3	67.8	65.6
MultiB_{EU}	45	68.9	65.9	29.2
MultiB_{CA}	74	72.2	73.7	28.2
MPQA	40	55.4	58.5	28.8
DS_{Unis}	10	56.9	43.1	31.4

Table 4: Number of sentences with multiple targets (#) and Macro F_1 on the target extraction task for Head-final and RACL. Head-final is consistently better than RACL on extracting multiple targets.

7.1 Do syntactically informed sentiment graphs improve results?

Our two baseline graph representations, Head-first and Head-final, are crude approximations of linguistic structure. In syntactic and semantic dependency graphs, heads are often neither the first or last word, but rather the most salient word according to various linguistic criteria. First, we enrich the dependency labels to distinguish edges that are internal to a holder/target/expression span from those that are external and perform experiments by adding an ‘in label’ to non-head nodes within the graph, which we call **+inlabel**. We further inform the head selection of the parsing graphs with syntactic information in the **Dep. edges** parsing graphs, where we compute the dependency graph for each sentence⁸ and set the head of each span to be the node that has an outgoing edge in the corresponding syntactic graph. As there can be more than one such edge, we default to the first. A manual inspection showed that this approach sometimes set unlikely dependency label types as heads, *e.g.* *punct*, *obl*. Therefore, we suggest a final approach, **Dep. labels**, which filters out these unlikely heads. The full results are shown in Table 7 in the Appendix. The implementation of the graph structure has a large effect on all metrics, although the specific results depend on the dataset. We plot the average effect of each implementation across all datasets in Figure 3, as well as each individual dataset (Figures 4–8 in the Appendix). **+inlabel** tends to improve results on the non-English datasets, consistently increasing target and expression extraction and targeted sentiment. It also generally improves the graph scores UF_1 and LF_1 on the non-English datasets.

⁸We use SpaCy (Honnibal et al., 2020) for English, Stanza (Qi et al., 2020) for Basque and Catalan and UDPipe (Straka and Straková, 2017) for Norwegian.

	Spans			Targeted	Graph		Sent. Graph	
	Holder F ₁	Target F ₁	Exp. F ₁	F ₁	UF ₁	LF ₁	NSF ₁	SF ₁
NoReC_{Fine}	1.2	5.0	3.4	4.2	2.8	2.7	4.6	4.0
MultiB_{EU}	2.9	0.6	0.8	1.1	1.0	1.4	1.2	1.4
MultiB_{CA}	0.4	1.6	1.6	2.1	2.0	1.8	3.3	2.8
MPQA	8.2	8.8	5.2	7.2	6.6	7.3	5.4	5.1
DS_{Unis}	7.9	1.2	4.3	6.4	3.9	5.7	3.6	6.0

Table 5: Average gains in percentage points by including mBERT representations.

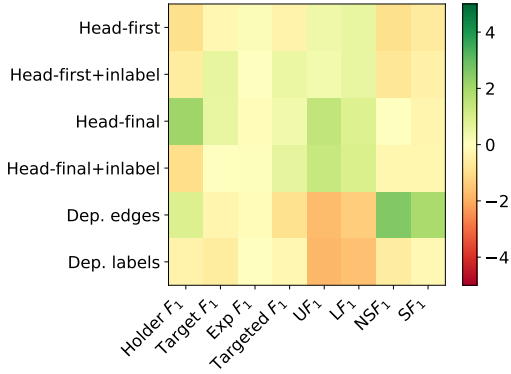


Figure 3: Average benefit of each graph annotation scheme (y-axis) on the evaluation metrics (x-axis) in percentage points. The results are averaged across datasets.

Dep. edges has the strongest positive effect on the NSF₁ and SF₁ (an avg. 2.52 and 2.22 percentage point (pp) over Head-final, respectively). However, this average is pulled down by poorer performance on the English datasets. Removing these two, the average benefit is 5.2 and 4.2 for NSF₁ and SF₁, respectively. On span extraction and targeted sentiment, however, Dep. edges leads to poorer scores overall. Dep. labels does not lead to any consistent improvements. These results indicate that incorporating syntactic dependency information is particularly helpful for the full fine-grained sentiment task, but that these benefits do not always show at a more local level, *i.e.* span extraction.

7.2 Do graph models perform better on sentences with multiple targets?

We hypothesize that predicting the full sentiment graph may have a larger effect on sentences with multiple targets. Therefore, we create a subset of the test data containing sentences with multiple targets and reevaluate Head-first, Head-final, and RACL-BERT on the target extraction task. Table 4 shows the number of sentences with multiple targets and the Target span extraction score for each

model. On this subset, Head-first and Head-final outperform RACL-BERT on 9 of 10 experiments, confirming the hypothesis that the graph models improve on examples with multiple targets.

7.3 How much does mBERT contribute?

We also perform experiments without mBERT (shown in Table 6 in the Appendix) and show the average gains (over all 6 graph setups) of including it in Table 5. Adding the mBERT features leads to average improvements in all experiments: for extracting spans an average gain of 4.1 pp for holders, 3.4 for targets, and 3.1 for expressions. For targeted sentiment there is a larger gain of 4.2 pp, while for the parsing graph metrics UF₁ and LF₁ the gains are more limited (3.3 pp/ 3.8 pp) and similarly for NSF₁ and SF₁ (3.6 pp/3.9 pp). The gains are largest for the English datasets (**MPQA**, **DS_{Unis}**) followed by **NoReC_{Fine}**, and finally **MultiB_{CA}** and **MultiB_{EU}**. This corroborates the bias towards English and similar languages that has been found in multilingual language models (Artetxe et al., 2020; Conneau et al., 2020) and motivates the need for language-specific contextualized embeddings.

8 Conclusion

In this paper, we have proposed a dependency graph parsing approach to fine-grained sentiment analysis and shown that these models outperform state-of-the-art sequence labeling models on five benchmark datasets. Using dependency parsing as part of a pipeline approach to fine-grained sentiment has shown promise in the past (Almeida et al., 2015). In the future, we would like to better exploit the similarities between dependency parsing and sentiment graph parsing, either by augmenting the token-level representations with contextualized vectors from their heads in a dependency tree (Kurtz et al., 2020) or by multi-task learning to dependency parse. We would also like to explore different graph parsing approaches, *e.g.* PERIN (Samuel and Straka, 2020).

References

- Mariana S. C. Almeida, Cláudia Pinto, Helena Figueira, Pedro Mendes, and André F. T. Martins. 2015. [Aligning opinions: Cross-lingual opinion mining with dependencies](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 408–418, Beijing, China. Association for Computational Linguistics.
- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. [On the cross-lingual transferability of monolingual representations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.
- Jeremy Barnes, Toni Badia, and Patrik Lambert. 2018. [MultiBooked: A corpus of basque and Catalan hotel reviews annotated for aspect-level sentiment classification](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).
- Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An Empirical Investigation of Statistical Significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.
- Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting Complex Biological Events with Rich Graph-based Feature Sets. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task, BioNLP '09*, pages 10–18, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zhuang Chen and Tieyun Qian. 2020. [Relation-aware collaborative learning for unified aspect-based sentiment analysis](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3694, Online. Association for Computational Linguistics.
- Yejin Choi, Eric Breck, and Claire Cardie. 2006. [Joint extraction of entities and relations for opinion recognition](#). In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Sydney, Australia. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. [Question answering passage retrieval using dependency relations](#). In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05*, pages 400–407, Salvador, Brazil. Association for Computing Machinery.
- 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.
- Timothy Dozat and Christopher D. Manning. 2018. [Simpler but more accurate semantic dependency parsing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.
- Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. [Word vectors, reuse, and replicability: Towards a community repository of large-text resources](#). In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.
- 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.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength Natural Language Processing in Python](#).
- Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. [Target-dependent twitter sentiment classification](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 151–160, Portland, Oregon, USA. Association for Computational Linguistics.
- Richard Johansson and Alessandro Moschitti. 2012. Relational Features in Fine-Grained Opinion Analysis. *Computational Linguistics*, 39(3):473–509.

- 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.
- Marco Kuhlmann and Stephan Oepen. 2016. Towards a Catalogue of Linguistic Graph Banks. *Computational Linguistics*, 42(4):819–827.
- Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. 2020. [End-to-end negation resolution as graph parsing](#). In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 14–24, Online. Association for Computational Linguistics.
- Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. UiO2: Sequence-labeling Negation Using Dependency Features. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, SemEval '12*, pages 319–327, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6714–6721.
- Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019b. [Exploiting BERT for end-to-end aspect-based sentiment analysis](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 34–41, Hong Kong, China. Association for Computational Linguistics.
- Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.
- 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.
- Roser Morante and Eduardo Blanco. 2012. *SEM 2012 Shared Task: Resolving the Scope and Focus of Negation. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 265–274, Montréal, Canada. Association for Computational Linguistics.
- Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg: Annotation of negation cues and their scope in Conan Doyle stories. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1563–1568, Istanbul, Turkey. European Language Resources Association (ELRA).
- Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Herscovich, Bin Li, Tim O’Gorman, Nianwen Xue, and Daniel Zeman. 2020. [MRP 2020: The Second Shared Task on Cross-Framework and Cross-Lingual Meaning Representation Parsing](#). In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. [SemEval 2015 Task 18: Broad-Coverage Semantic Dependency Parsing](#). *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. [SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing](#). *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72.
- Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. [A fine-grained sentiment dataset for Norwegian](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryigit. 2016. [SemEval-2016 task 5: Aspect based sentiment analysis](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. [SemEval-2015 task 12: Aspect based sentiment analysis](#). In *Proceedings of the 9th International*

- Workshop on Semantic Evaluation (SemEval 2015), pages 486–495, Denver, Colorado. Association for Computational Linguistics.
- 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.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- David Samuel and Milan Straka. 2020. [ÚFAL at MRP 2020: Permutation-invariant semantic parsing in PERIN](#). In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 53–64, Online. Association for Computational Linguistics.
- Milan Straka and Jana Straková. 2017. [Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe](#). In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. 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.
- Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. [Recursive neural conditional random fields for aspect-based sentiment analysis](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626, Austin, Texas. Association for Computational Linguistics.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.
- 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.
- Bishan Yang and Claire Cardie. 2012. [Extracting opinion expressions with semi-Markov conditional random fields](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1335–1345, Jeju Island, Korea. Association for Computational Linguistics.
- Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named Entity Recognition as Dependency Parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.
- Meishan Zhang, Qiansheng Wang, and Guohong Fu. 2019. [End-to-end neural opinion extraction with a transition-based model](#). *Information Systems*, 80:56 – 63.

A Appendix

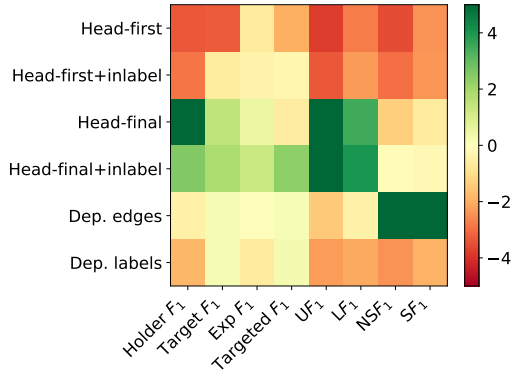


Figure 4: Average benefit of each graph annotation scheme (y-axis) on the evaluation metrics (x-axis) for **NoReC_{Fine}**.

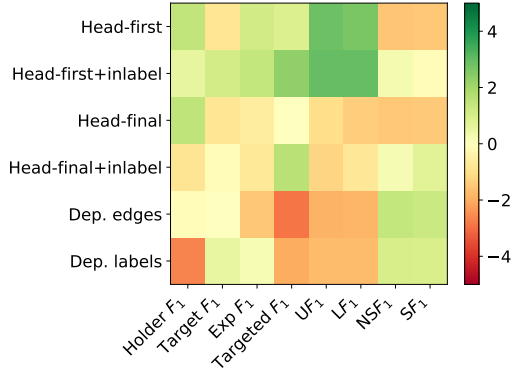


Figure 5: Average benefit of each graph annotation scheme (y-axis) on the evaluation metrics (x-axis) for **MultiB_{EU}**.

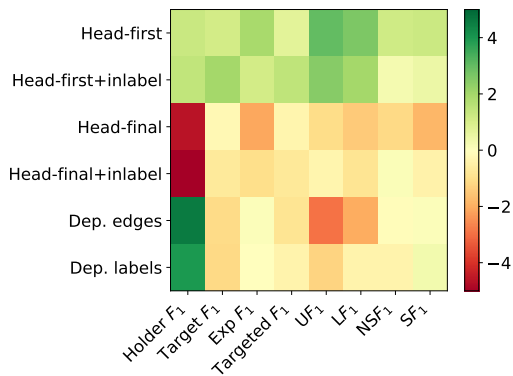


Figure 6: Average benefit of each graph annotation scheme (y-axis) on the evaluation metrics (x-axis) in percentage points for **MultiB_{CA}**.

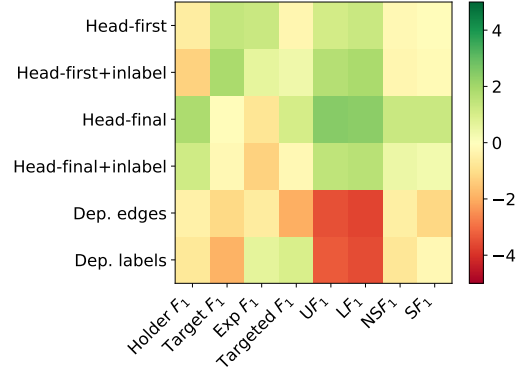


Figure 7: Average benefit of each graph annotation scheme (y-axis) on the evaluation metrics (x-axis) in percentage points for **MPQA**.

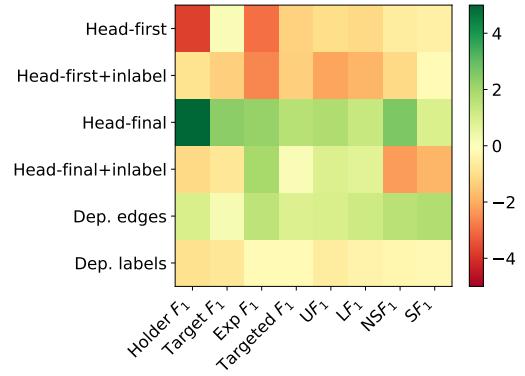


Figure 8: Average benefit of each graph annotation scheme (y-axis) on the evaluation metrics (x-axis in percentage points) for **DS_{Unis}**.

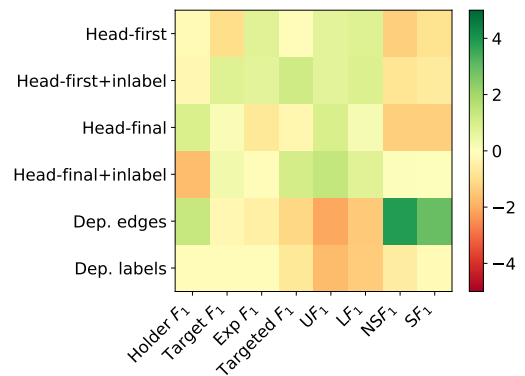


Figure 9: Average benefit of each graph annotation scheme (y-axis) on the evaluation metrics (x-axis) in percentage points. The results on **NoReC_{Fine}**, **MultiB_{EU}**, **MultiB_{CA}**.

		Spans			Targeted	Graph		Sent. Graph	
		Holder F ₁	Target F ₁	Exp. F ₁	F ₁	UF ₁	LF ₁	NSF ₁	SF ₁
NoReC _{Fine}	IMN	-	35.9	48.7	18.0	-	-	-	-
	RACL	-	45.6	55.4	20.1	-	-	-	-
	Head-first	48.4 (2.2)	47.1 (1.6)	52.0 (1.6)	33.0 (1.4)	37.6 (0.5)	29.8 (0.4)	32.9 (1.6)	26.1 (1.5)
	+inlabel	50.4 (4.0)	47.6 (2.5)	51.0 (1.3)	27.3 (1.1)	36.9 (0.5)	29.4 (0.8)	32.9 (1.1)	25.8 (0.5)
	Head-final	57.0 (3.3)	49.4 (0.9)	52.1 (1.8)	26.0 (0.6)	45.1 (1.2)	35.2 (1.1)	34.4 (0.7)	27.2 (0.9)
	+inlabel	57.9 (1.7)	50.1 (1.3)	52.6 (0.4)	29.6 (0.6)	45.0 (1.0)	35.2 (0.5)	35.1 (1.6)	27.0 (1.3)
	Dep. edges	54.4 (3.9)	49.0 (2.5)	51.4 (1.7)	26.7 (3.1)	39.3 (1.1)	31.5 (1.3)	47.2 (0.9)	36.0 (1.1)
	Dep. labels	51.6 (2.6)	46.5 (3.0)	50.7 (2.7)	26.7 (1.9)	36.7 (1.1)	28.3 (0.8)	33.4 (1.8)	25.4 (1.8)
	IMN	-	48.2	65.2	39.5	-	-	-	-
	RACL	-	55.4	70.7	48.2	-	-	-	-
MultiB _{EU}	Head-first	60.8 (3.8)	64.1 (1.4)	72.2 (0.7)	53.9 (1.8)	62.9 (0.6)	58.2 (0.3)	58.5 (2.3)	54.7 (2.6)
	+inlabel	59.8 (1.6)	64.3 (0.9)	71.9 (0.8)	57.9 (1.9)	62.6 (0.6)	57.5 (1.1)	57.3 (1.5)	53.6 (1.3)
	Head-final	57.0 (2.0)	66.0 (1.6)	72.2 (0.6)	55.5 (1.7)	60.2 (0.8)	55.5 (0.9)	59.6 (0.8)	56.3 (1.0)
	+inlabel	53.7 (1.2)	64.0 (2.4)	72.9 (0.6)	54.9 (2.0)	60.1 (1.5)	54.9 (1.7)	57.1 (3.2)	53.5 (3.3)
	Dep. edges	53.1 (1.9)	63.8 (1.7)	71.0 (1.1)	53.7 (1.7)	59.0 (1.3)	54.5 (1.6)	59.0 (1.6)	55.6 (1.8)
	Dep. labels	52.0 (3.8)	63.0 (1.1)	71.3 (1.9)	54.0 (1.0)	59.5 (1.1)	54.9 (1.1)	58.6 (2.8)	54.6 (2.4)
	IMN	-	56.3	60.9	32.5	-	-	-	-
MultiB _{CA}	RACL	-	65.4	67.6	53.1	-	-	-	-
	Head-first	41.9 (2.8)	69.8 (1.7)	68.9 (1.4)	57.3 (2.0)	64.2 (0.7)	59.9 (0.8)	58.2 (2.3)	53.3 (2.2)
	+inlabel	42.4 (2.6)	70.9 (0.8)	69.9 (0.9)	50.9 (1.3)	64.4 (0.9)	59.6 (0.7)	55.7 (2.0)	50.7 (2.1)
	Head-final	40.4 (2.5)	69.9 (1.5)	66.8 (0.8)	50.8 (2.6)	60.9 (0.6)	57.1 (0.8)	57.7 (1.0)	53.3 (1.3)
	+inlabel	36.4 (2.2)	69.1 (1.1)	65.4 (0.6)	52.9 (0.9)	60.6 (0.9)	57.0 (0.6)	58.0 (1.9)	53.5 (2.0)
	Dep. edges	42.6 (6.1)	69.1 (0.5)	67.3 (0.6)	50.6 (1.3)	59.3 (0.7)	55.7 (1.1)	57.5 (2.1)	52.8 (1.8)
	Dep. labels	43.8 (3.4)	70.3 (0.8)	67.2 (1.1)	50.6 (1.8)	61.0 (0.5)	57.1 (0.8)	57.8 (1.4)	52.7 (1.9)
MPQA	IMN	-	24.3	29.6	1.2	-	-	-	-
	RACL	-	32.6	37.8	11.8	-	-	-	-
	Head-first	35.2 (1.1)	40.5 (1.8)	41.7 (1.7)	22.6 (3.1)	32.2 (1.3)	28.2 (1.4)	19.4 (1.5)	12.4 (1.7)
	+inlabel	35.6 (1.4)	41.6 (1.1)	42.4 (2.3)	14.0 (0.9)	32.9 (0.8)	28.9 (0.9)	20.4 (1.0)	13.2 (1.2)
	Head-final	37.1 (1.3)	42.1 (1.0)	41.9 (0.8)	13.3 (1.9)	35.7 (0.8)	31.7 (0.5)	18.7 (0.7)	12.5 (1.6)
	+inlabel	37.0 (0.5)	42.3 (1.2)	41.6 (1.6)	15.2 (2.5)	35.5 (0.7)	31.7 (0.6)	19.6 (0.6)	12.6 (0.8)
	Dep. edges	35.4 (1.5)	39.1 (2.0)	41.6 (1.1)	12.3 (0.8)	28.9 (1.7)	24.8 (1.5)	19.0 (0.9)	11.9 (1.1)
DS _{Units}	Dep. labels	36.7 (0.6)	39.2 (2.3)	40.4 (2.1)	12.6 (1.2)	28.9 (1.6)	24.5 (1.1)	18.9 (0.8)	11.6 (1.0)
	IMN	-	33.0	27.4	17.9	-	-	-	-
	RACL	-	39.3	40.2	22.8	-	-	-	-
	Head-first	25.6 (5.1)	36.8 (3.5)	39.0 (1.5)	23.4 (1.8)	32.9 (1.7)	25.9 (1.7)	27.8 (1.4)	18.8 (2.0)
	+inlabel	22.9 (5.7)	38.6 (3.9)	38.6 (2.8)	18.3 (2.7)	33.7 (2.8)	26.4 (1.6)	25.9 (3.4)	15.2 (2.2)
	Head-final	29.2 (8.4)	38.1 (2.0)	39.5 (2.4)	21.8 (1.1)	31.1 (2.1)	26.0 (1.0)	29.1 (3.3)	20.4 (2.0)
	+inlabel	30.2 (8.9)	38.2 (2.9)	38.8 (2.0)	24.4 (2.7)	32.4 (2.1)	28.4 (2.1)	28.1 (2.9)	22.4 (3.4)
	Dep. edges	33.9 (5.4)	39.2 (2.9)	39.3 (3.4)	22.2 (2.7)	32.4 (2.1)	26.8 (2.0)	29.3 (1.5)	19.8 (1.2)
	Dep. labels	21.3 (18.1)	40.0 (1.5)	38.4 (2.4)	21.4 (4.3)	32.1 (1.5)	27.2 (1.5)	28.2 (1.1)	20.5 (3.1)

Table 6: Experiments without contextualized embeddings.

		Spans			Targeted	Graph		Sent. Graph	
		Holder F ₁	Target F ₁	Exp. F ₁	F ₁ .	UF ₁	LF ₁	NSF ₁	SF ₁
NoReC _{fine}	RACL-BERT	-	47.2	56.3	30.3	-	-	-	-
	Head-first	51.1 (3.2)	50.1 (3.4)	54.4 (1.6)	30.5 (2.3)	39.2 (0.5)	31.5 (0.5)	37.0 (2.6)	29.5 (2.4)
	+inlabel	51.6 (2.8)	52.7 (0.7)	54.6 (1.4)	32.2 (1.4)	39.6 (0.8)	32.0 (0.7)	37.6 (1.2)	29.5 (1.2)
	Head-final	60.4 (1.2)	54.8 (1.6)	55.5 (1.5)	31.9 (1.3)	48.0 (1.3)	37.7 (1.4)	39.2 (1.7)	31.2 (1.6)
	+inlabel	57.1 (3.0)	55.2 (1.0)	56.3 (1.3)	34.8 (1.0)	48.7 (1.2)	38.3 (1.0)	40.5 (1.1)	31.7 (1.1)
	Dep. edges	54.0 (3.4)	53.6 (1.5)	55.0 (0.9)	32.7 (1.6)	41.5 (0.7)	33.8 (0.4)	50.9 (0.3)	39.4 (0.4)
	Dep. labels	52.7 (5.6)	53.6 (0.3)	54.4 (1.5)	32.7 (1.6)	40.7 (0.8)	32.2 (0.5)	38.2 (1.4)	30.0 (1.2)
MultiB _{EU}	RACL-BERT	-	59.9	72.6	56.8	-	-	-	-
	Head-first	60.4 (2.2)	64.0 (2.4)	73.9 (1.0)	57.8 (2.4)	64.6 (1.0)	60.0 (1.6)	58.0 (1.1)	54.7 (1.6)
	+inlabel	59.6 (1.9)	65.9 (0.9)	74.2 (0.7)	59.2 (0.9)	64.7 (0.7)	60.3 (1.1)	59.8 (1.1)	56.1 (1.6)
	Head-final	60.5 (2.2)	64.0 (2.3)	72.1 (1.2)	56.9 (1.7)	60.8 (0.8)	56.0 (1.1)	58.0 (2.1)	54.7 (1.8)
	+inlabel	58.1 (2.4)	64.7 (1.1)	72.0 (0.7)	58.5 (1.4)	60.6 (1.1)	56.6 (0.7)	59.8 (1.6)	56.9 (1.8)
	Dep. edges	58.8 (4.2)	64.8 (1.4)	71.2 (0.8)	54.0 (1.8)	59.9 (0.4)	55.5 (0.7)	60.9 (1.6)	57.4 (1.6)
	Dep. labels	56.3 (2.1)	65.4 (0.9)	72.9 (1.1)	54.9 (0.8)	60.0 (0.9)	55.6 (0.8)	60.5 (1.1)	57.1 (1.1)
MultiB _{CA}	RACL-BERT	-	67.5	70.3	52.4	-	-	-	-
	Head-first	43.0 (1.3)	72.5 (1.0)	71.1 (0.8)	55.0 (0.9)	66.8 (0.5)	62.1 (0.5)	62.0 (1.1)	56.8 (0.7)
	+inlabel	43.1 (2.2)	73.4 (1.0)	70.3 (1.0)	55.8 (1.8)	66.2 (0.3)	61.5 (0.6)	61.1 (1.0)	56.0 (1.0)
	Head-final	37.1 (4.2)	71.2 (0.6)	67.1 (1.7)	53.9 (2.2)	62.7 (0.4)	58.1 (0.8)	59.7 (1.1)	53.7 (2.4)
	+inlabel	34.9 (4.1)	70.7 (1.4)	68.2 (1.0)	53.5 (0.7)	63.4 (0.5)	58.7 (0.6)	60.9 (1.1)	55.1 (1.2)
	Dep. edges	46.3 (3.1)	70.3 (0.6)	69.2 (1.4)	53.4 (1.5)	60.8 (0.4)	57.5 (0.6)	60.7 (1.0)	55.6 (0.9)
	Dep. labels	45.6 (2.9)	70.3 (1.1)	69.1 (1.7)	53.9 (1.5)	62.5 (0.6)	59.1 (0.6)	60.4 (1.0)	55.8 (1.2)
MPQA	RACL-BERT	-	20.0	31.2	17.8	-	-	-	-
	Head-first	43.8 (1.8)	51.0 (1.9)	48.1 (0.8)	33.5 (3.1)	40.0 (1.0)	36.9 (1.2)	24.5 (2.3)	17.4 (2.7)
	+inlabel	43.1 (1.5)	51.5 (1.0)	47.5 (1.1)	21.3 (0.4)	40.6 (0.5)	37.5 (0.5)	24.5 (1.3)	17.3 (1.0)
	Head-final	46.3 (1.8)	49.5 (0.9)	46.0 (1.1)	21.9 (1.4)	41.4 (0.7)	38.0 (0.5)	26.1 (0.7)	18.8 (0.7)
	+inlabel	45.6 (2.5)	49.4 (2.1)	45.6 (1.1)	20.7 (1.0)	40.4 (1.5)	37.2 (1.9)	25.2 (1.7)	17.8 (1.3)
	Dep. edges	44.0 (1.5)	48.5 (1.2)	46.3 (1.9)	18.9 (2.3)	35.4 (1.3)	31.9 (1.2)	24.2 (1.6)	16.3 (1.9)
	Dep. labels	43.7 (0.9)	47.7 (2.3)	47.5 (0.8)	21.9 (0.7)	35.6 (1.2)	32.0 (1.3)	24.0 (0.8)	17.2 (0.8)
DS _{Units}	RACL-BERT	-	44.6	38.2	27.3	-	-	-	-
	Head-first	28.0 (7.7)	39.9 (2.2)	40.3 (0.6)	26.7 (2.1)	35.3 (0.9)	31.4 (1.3)	31.0 (1.4)	25.0 (1.3)
	+inlabel	30.9 (9.9)	38.4 (3.3)	40.6 (2.9)	26.7 (2.4)	34.2 (2.1)	30.7 (2.5)	30.5 (2.1)	25.4 (2.3)
	Head-final	37.4 (11.6)	42.1 (2.7)	45.5 (2.4)	29.6 (1.7)	38.1 (1.9)	33.9 (2.3)	34.3 (4.2)	26.5 (3.5)
	+inlabel	30.6 (16.4)	38.9 (3.1)	45.2 (2.7)	28.1 (3.7)	37.3 (2.7)	33.3 (2.1)	29.4 (2.8)	23.7 (2.4)
	Dep. edges	32.7 (12.1)	39.9 (2.8)	44.8 (4.0)	28.9 (3.6)	37.3 (2.5)	33.8 (2.7)	33.2 (4.5)	27.3 (4.1)
	Dep. labels	30.8 (5.8)	38.9 (0.9)	43.1 (1.2)	27.8 (1.9)	35.7 (1.5)	32.1 (1.6)	31.3 (1.1)	25.3 (2.0)

Table 7: Experiments with mBERT.

GPU Infrastructure	NVIDIA P100, 16 GiB RAM
CPU Infrastructure	Intel Xeon-Gold 6138 2.0 GHz
Training duration	00:31:43 (MultiB _{EU}) – 07:40:54 (NoReC _{Fine})
Model implementation	https://blinded.for.review
Hyperparameter	Best assignment
embedding	Word2Vec SkipGram 100D
contextualized embedding	mBERT
embeddings trainable	False
number of epochs	100
batch size	50
beta1	0
beta2	0.95
l2	3e-09
hidden lstm	200
hidden char lstm	100
layers lstm	3
dim mlp	200
dim embedding	100
dim char embedding	80
early stopping	0
pos style	xpos
attention	bilinear
model interpolation	0.5
loss interpolation	0.025
lstm implementation	drop connect
char implementation	convolved
emb dropout type	replace
bridge	dpa+
dropout embedding	0.2
dropout edge	0.2
dropout label	0.3
dropout main recurrent	0.2
dropout recurrent char	0.3
dropout main ff	0.4
dropout char ff	0.3
dropout char linear	0.3