Better Document-Level Natural Language Understanding through Data-Driven Applications of Discourse Theories

by

Patrick Huber

B.Eng., The Cooperative State University Karlsruhe, 2014
M.Sc., The Karlsruhe Institute of Technology, 2018

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(COMPUTER SCIENCE)

The University of British Columbia
(Vancouver)

October 2022

© Patrick Huber, 2022
The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

**Better Document-Level Natural Language Understanding through Data-Driven Applications of Discourse Theories**

submitted by **Patrick Huber** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Computer Science**.

**Examining Committee:**

Giuseppe Carenini, Professor, Computer Science, UBC
*Supervisor*

Leonid Sigal, Professor, Computer Science, UBC
*Supervisory Committee Member*

Kevin Leyton-Brown, Professor, Computer Science, UBC
*Supervisory Committee Member*

Rachel Pottinger, Professor, Computer Science, UBC
*University Examiner*

Miikka Silfverberg, Professor, Linguistics, UBC
*University Examiner*

Nicholas Asher, Professor, Institut de Recherche en Informatique de Toulouse (IRIT), Université Toulouse
*External Examiner*
Abstract

A discourse constitutes a locally and globally coherent text in which words, clauses and sentences are not solely a sequence of independent statements, but follow a hidden structure, encoding the author’s underlying communicative goal(s). As such, the meaning of a discourse as a whole goes beyond the meaning of its individual parts, guided by the latent semantic and pragmatic relationships holding between parts of the document. Clearly falling into the area of Natural Language Understanding (NLU), discourse analysis augments textual inputs with structured representations following linguistic formalisms and frameworks. Annotating documents following these elaborate formalisms has led to the computationally inspired research area of discourse parsing, aiming to generate robust and general discourse annotations for arbitrary documents through automated approaches.

With computational discourse parsers having great success at inferring valuable structures and supporting prominent real-world tasks such as sentiment analysis, text classification, and summarization, discourse parsing has been established as a valuable source of structured information. However, a significant limitation preventing the broader application of discourse-inspired approaches, especially in the context of modern deep learning models, is the lack of available gold-standard data, caused by the tedious and expensive human annotation process.

To overcome the prevalent data sparsity issue in the areas of discourse analysis and discourse parsing, it is imperative to find new methods to generate large-scale and high-quality discourse annotations, not relying on the restrictive human annotation process. Along these lines, we present a set
of novel computational approaches to (partially) overcome the data sparsity issue by proposing distantly and self-supervised methods to automatically generate large-scale, high-quality discourse annotations in a data-driven manner. In this thesis, we provide detailed insights into our technical contributions and diverse evaluations. Specifically, we show the competitive and complementary nature of our discourse inference approaches to human-annotated discourse information, partially outperforming gold-standard discourse structures on the important task of “inter-domain” discourse parsing. We further elaborate on our generated discourse annotations in regard to their ability to support linguistic theories and downstream tasks, finding that they have direct applications in linguistics and Natural Language Processing (NLP).
Lay Summary

To understand natural language, its underlying structure needs to be uncovered. Discourse analysis provides the means to reveal this structure through well-defined linguistic theories. However, due to the tedious and expensive process to obtain high-quality discourse annotations, available discourse datasets are restrictively small, impairing the application of modern computational approaches. In this work, we propose a suite of models and methods to overcome this limitation by exploiting readily available data from related domains, including reviews, summaries, structured text, and general language. We show that the available data from related domains never completely covers all facets of complex textual structures. However, they encode important constructs of discourse, support real-world tasks (e.g., sentiment analysis), and show potential to extend linguistic theories.
Preface

The work presented henceforth is original, independent work by the author, Patrick Huber, conducted at the Natural Language Processing (NLP) research group, Faculty of Computer Science, at the University of British Columbia, Point Grey campus. Parts of this work have been previously published at Artificial Intelligence, Machine Learning and Natural Language Processing conferences as stated below.

A version of Chapter 3 has been published at EMNLP 2019 [59] and EMNLP 2020 [61]. Work in Chapter 6 was published and presented at NAACL 2022 [63] and the work described in Chapter 7 has been published at AAAI 2021 [62]. Parts of Chapter 9 have been published at COLING 2020 [60]. For all work mentioned, I was the lead investigator, responsible for all major areas of concept formation, statement of research questions, data collection, implementation as well as paper composition. Giuseppe Carenini was the supervisory author on all projects and was involved throughout the project in concept formation, discussions, and paper composition.

Chapter 4 was published and presented at AAAI 2022 [65] and Chapter 10 is based on our ACL 2021 publication [64]. In both projects, I was the lead investigator, working with fellow Ph.D. students Wen Xiao and Linzi Xing, respectively. I was partially responsible for the concept formation, formulation of research questions, data collection, and implementation of discourse-related components and evaluations. I further guided the paper composition and wrote substantial parts of the publications. Giuseppe Carenini supervised and supported both projects.

A version of Chapter 5 has been published at NAACL 2021 [171] and
parts of Chapter 8 have been published at COLING 2020 [51]. In both projects, I have been the second author (along with, in order of appearance: Grigorii Guz, Wen Xiao), contributed to the process of defining project goals and key research questions, supported the implementation, and took stakes in the design of the experiments. I further contributed to the paper itself in all cases. Giuseppe Carenini was the supervisory author for both projects.

Portions of the abstract, Chapter 1, and Chapter 11 are an aggregation of the publications described above.
Table of Contents

Abstract ......................................................... iii
Lay Summary ...................................................... v
Preface ............................................................. vi
Table of Contents ................................................ viii
List of Tables .................................................... xv
List of Figures ..................................................... xxii
Glossary ........................................................... xxxvii
Acknowledgments ................................................ xxxviii

1 Introduction ................................................... 2

2 Related Work .................................................. 10
  2.1 Discourse Analysis ........................................ 10
  2.2 Discourse Treebanks and Metrics ......................... 13
    2.2.1 Discourse Treebanks ................................. 13
    2.2.2 Discourse Evaluation Metrics .................... 15
  2.3 Discourse Parsing ......................................... 15
    2.3.1 Supervised Discourse Parsing .................... 16
    2.3.2 Distantly Supervised Discourse Parsing .......... 18
4.2 Related Work ........................................... 63
4.3 Methodology ........................................... 63
  4.3.1 The Topic Segmentation Model ................. 64
  4.3.2 Tree Generation .................................. 66
4.4 Evaluation ............................................ 68
  4.4.1 Datasets ........................................ 69
  4.4.2 Baselines ....................................... 69
  4.4.3 Experimental Design ............................ 70
  4.4.4 Experiments and Results ...................... 71
4.5 Contributions ....................................... 76

5 Discourse Inference from Summarization Annotations . . . 78
  5.1 Motivation .......................................... 78
  5.2 Related Work ...................................... 80
  5.3 Our Discourse Inference Method .................. 81
    5.3.1 Framework Overview .......................... 81
    5.3.2 Parsing Algorithms ............................ 82
  5.4 Evaluation ........................................ 86
    5.4.1 The Summarization Task ..................... 86
    5.4.2 Discourse Datasets ............................ 87
    5.4.3 Evaluation Metric ............................. 87
    5.4.4 Main Results ................................ 88
    5.4.5 Performance of Heads ....................... 91
    5.4.6 Analysis of Generated Trees ............... 92
    5.4.7 Model Sensitivity and Summarizer Quality 94
  5.5 Contributions ..................................... 95

II Discourse Inference from Self-Supervised Language Models ........................................... 97

6 Discourse Inference from Pre-Trained Language Models . 99
  6.1 Motivation ......................................... 99
  6.2 Related Work ...................................... 101
10.3 The W-RST Treebank Generation ........................................ 177
  10.3.1 Weighted Trees from Sentiment ................................. 177
  10.3.2 Weighted Trees from Summarization .......................... 178
10.4 The W-RST Weight Integration ........................................ 180
  10.4.1 Weight-based Discourse Applications .......................... 180
  10.4.2 Weight Alignment with Human Annotation ..................... 183
10.5 Evaluation ......................................................................... 185
  10.5.1 Experimental Setup .................................................. 185
  10.5.2 Results and Analysis ................................................ 186
10.6 Contributions .................................................................... 191

11 Conclusion ......................................................................... 193

Bibliography ........................................................................... 202

A Discourse Inference from Distant Supervision ......................... 229
  A.1 Qualitative Analysis of Generated Discourse Trees from Sentiment Annotations ........................................ 229
  A.2 Complete Quantitative Results of Generated Discourse Trees from Topic Segmentation Annotations ............ 235
  A.3 Qualitative Analysis of Generated Discourse Trees from Topic Segmentation Annotations on GUM ............ 240
  A.4 Detailed Performance of the Discourse Augmented Summarization Task ..................................................... 241
  A.5 Full Results on Discourse Inference from Summarization Annotations ......................................................... 242
  A.6 Results on Sensitivity to Initialization for Discourse Inference from Summarization Annotations ................ 243

B Discourse Inference from Self-Supervision ............................... 246
  B.1 Huggingface Models for Discourse Inference from Pre-Trained Language Models ........................................ 246
  B.2 PLM Discourse Inference Test Set Results on RST-DT and GUM .............................................................. 247
List of Tables
Table 3.1

Upper bound of spatial complexity for different beam sizes
and unconstrained CKY (∞), assuming 1Byte per unit in
memory. KB = 103 , MB = 106 , GB = 109 , PB = 1015 , SB
= 1054 . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

47

Table 3.2

Size, length and vocabulary diversity of the datasets . . . .

49

Table 3.3

Vocabulary overlap between datasets . . . . . . . . . . . .

50

Table 3.4

Results of the micro-averaged precision measure using the
original Parseval method (Par.) and RST Parseval (R-Par.).
Inter-domain subscripts identify the training set. Interdomain results averaged over 10 independent runs. Models
with stochastic components are averaged over 3 distinct
generation processes. Best performance per sub-table is
bold. * Results taken from Morey et al. [115], † statistically
significant with p-value ≤ .05 to the best inter-domain baseline (Bonferroni adjusted), – non-published values, × not
feasible combinations, TS=Two-Stage, Branch=Branching.

Table 3.5

53

Ablation study showing the influence of nuclearity and
stochasticity on the overall performance, measured as the
micro-average precision using original Parseval (Par.) and
RST Parseval (R-Par.). Results averaged over 10 runs (using 3 distinct generation processes if a stochastic component
is included). The best performance is bold. TS=Two-Stage. 54

xv


Table 3.6 Confusion Matrices for our MEGA-DT-trained model evaluated on RST-DT (left) and Instr-DT (right).

Table 4.1 Average topic segmentation performance ± standard deviation over 5 runs, using the $P_k$ error score [11] (lower values indicate better performance). Best performance per column in **bold**.

Table 4.2 Statistics of the three training/testing datasets.

Table 4.3 Evaluation results using the RST Parseval micro-average precision measure. Subscripts indicate the training dataset. TS = Topic Segmentation Model. * = Average performance over 10 runs. Best performance per sub-table underlined, best performance per column **bold**.

Table 4.4 RST Parseval micro-average precision on paragraph-to-document-level for genres in the GUM corpus. Sample(s) in “Vlog” & “Conversation” only contain a single paragraph and are therefore omitted. RB=Right-Branching, 2S=Two-Stage parser, TS=Topic Segmentation model.

Table 5.1 Key RST-style discourse dataset dimensions of the three evaluation corpora used in this chapter.

Table 5.2 RST Parseval Scores of generated constituency trees on the three datasets, expressed as 'Avg. ± Std’. ↓=Result is worse than Random. Results for Random are obtained by applying the parser to random matrices 10 times. $A0|1$ are the first two layers.

Table 5.3 Unlabeled Attachment Scores of dependency trees generated by the Eisner algorithm. ↓=Result is worse than Random. $A0|1$ are the first two layers.

Table 5.4 Unlabeled Attachment Scores of dependency trees generated by the CLE algorithm. ↓=Result is worse than Random. $A0|1$ are the first two layers.
Table 5.5 Measurements on the locality of the generated dependency trees. Numbers are %. Corr.=correct predictions, GT=ground-truth trees, Ours=generated tree.

Table 5.6 Statistics of our generated trees and the gold standard trees in terms of the average branch width, average height, average leaf ratio (micro), average normalized arc length of the trees and percentage of the Vacuous trees.

Table 6.1 The seven fine-tuning datasets used in this work along with the underlying tasks and domains.

Table 6.2 Original parseval (Span) and Unlabelled Attachment Score (UAS) of the single best-performing self-attention matrix of the BERT and BART models compared with baselines and previous work. ↑, ●, ↓ indicate better, same, or worse performance compared to the PLM. rand. init=Randomly initialized transformer model of similar architecture as the PLM, RB=Right-Branching, LB=Left-Branching, Chain⁻¹=Inverse chain.

Table 7.1 Results of the micro-average precision measure, evaluated on the RST-DT corpus. Subscripts identify training sets. Best model in each subset is bold.

Table 7.2 Five-class sentiment accuracy scores trained and tested on the Yelp’13 dataset, subscripts in model-names indicate dataset for self-supervised training. Best model is bold.

Table 7.3 Representationally similar/different document-encodings based on the cosine similarity. For more examples of the representational similarity and additional tree structure comparisons see Appendix B.7 and B.8 respectively.
Table 8.1 Micro-averaged F1-scores for structure and nuclearity prediction using the original Parseval measure as proposed in Morey et al. [115], evaluated on the RST-DT and Instr-DT corpora. Best performance per column is bold. Subscripts on results indicate standard deviation, – non-published values. . . . . . . . . . . . . . . . . . . . . . . . 151

Table 8.2 Micro-averaged F1-scores for structure and nuclearity prediction using RST Parseval, evaluated on the RST-DT and Instr-DT corpora. Best performance per column is bold. Subscripts on results indicate standard deviation, – non-published values. . . . . . . . . . . . . . . . . . . . . . . . 152

Table 9.1 Final evaluation on the Yelp’13 dataset. Subscripts in model names indicate discourse augmentation treebanks used to generate discourse trees. Best model for each metric is bold. ≃Performance statistically equivalent to HAN model, †MEGA-DT treebank significantly better than RST-DT with p-value 0.05. ‡MEGA-DT treebank marginally significantly better than RST-DT with p-value 0.05-0.1, *Statistically significant to best model on the metric. All significance computations are Bonferroni adjusted. . . . . . 169

Table 10.1 Statistics on consistently and inconsistently annotated samples of the 1,354 structure-aligned sub-trees generated by two distinct human annotators. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186

Table 10.2 Confusion Matrices based on human annotation showing the absolute weight spread using the Sentiment (left) and Summarization (right) tasks on 620 and 791 sub-trees aligned with the human structure prediction, respectively. Cell upper value: Absolute weight spread for the respective combination of human-annotated nuclearities. Lower value (in brackets): Support for this configuration. . . . . . . . . 188
Table 10.3 Confusion Matrices based on human annotation showing the weight spread relative to the task-average for Sentiment (left) and Summarization (right), aligned with the human structure prediction, respectively. Cell value: Relative weight spread as the divergence from the average spread across all cells in Table 10.2. $\varnothing$=Average value of absolute scores. ............................................ 188

Table A.1 Evaluation results using the **RST Parseval** micro-average precision measure on the RST-DT dataset. Subscripts indicate the training dataset. TS=Topic Segmentation Model. Disc-Seg=Discourse Segmentation Model by Wang et al. [161]. $\times$=Not feasible combination. *=Average performance over 10 runs. Best performance per sub-table underlined, best performance per column **bold** ....... 236

Table A.2 Evaluation results using the **RST Parseval** micro-average precision measure on the GUM dataset. Subscripts indicate the training dataset. TS=Topic Segmentation Model. Disc-Seg=Discourse Segmentation Model by Wang et al. [161]. $\times$=Not feasible combination. *=Average performance over 10 runs. Best performance per sub-table underlined, best performance per column **bold** .. ... 237

Table A.3 Evaluation results using the **original Parseval** micro-average precision measure on the RST-DT dataset. Subscripts indicate the training dataset. TS=Topic Segmentation Model. Disc-Seg=Discourse Segmentation Model by Wang et al. [161]. $\times$=Not feasible combination. *=Average performance over 10 runs. Best performance per sub-table underlined, best performance per column **bold** ....... 238
Table A.4 Evaluation results using the original Parseval micro-average precision measure on the GUM dataset. Subscripts indicate the training dataset. TS=Topic Segmentation Model. Disc-Seg=Discourse Segmentation Model by Wang et al. [161]. ×=Not feasible combination. *=Average performance over 10 runs. Best performance per sub-table underlined, best performance per column bold.

Table A.5 In-domain performance of the summarizers on the Rouge-1, Rouge-2 and Rouge-L scores.

Table A.6 The RST Parseval Scores of generated constituency trees and Unlabeled Attachment Score of generated dependency trees by the Eisner and CLE algorithms on the three datasets. The numbers in each cell are represented as the performance of (Layer 0 / Layer 1). Random results are obtained by applying the parser on randomly generated matrices 10 times, and are represented as “Average (Std)”.

Table A.7 The average RST Parseval Scores of generated constituency trees and average Unlabeled Attachment Scores of generated dependency trees using the Eisner and CLE algorithms, on the three datasets with 5 random initialization. Numbers in parenthesis are the standard deviation across different runs.

Table B.1 Huggingface pre-trained and fine-tuned model links.

Table B.2 Original parseval (span) and Unlabelled Attachment Score (UAS) of the single best-performing oracle self-attention matrix and validation set picked head (in brackets) of the BERT and BART models compared with baselines and previous work. “rand. init”=Randomly initialized transformer model of similar architecture as the PLM.

Table B.3 Minimum, median, mean and maximum performance of the self-attention matrices on RST-DT and GUM for the BERT model.
Table B.4 Minimum, median, mean and maximum performance of the self-attention matrices on RST-DT and GUM for the BART model. 251
Table B.5 Representationally similar/different document-encodings based on the cosine similarity. 255
Table B.6 Representationally similar/different document-encodings based on the cosine similarity. 256
Table B.7 Representationally similar/different document-encodings based on the cosine similarity. 257
Table C.1 Results of the W-RST approach compared to threshold-based nuclearity assignments and supervised training on RST-DT. 260
List of Figures

Figure 1.1 Corpus size of commonly used discourse treebanks compared to datasets for the sentiment analysis and summarization tasks. The size of circles indicate dataset sizes, i.e. the number of documents.

Figure 1.2 Thesis overview aligning chapters along the axis of supervision signals (as columns) and objectives (as rows). Ling=Application to improve linguistic theories, Parse=Application to discourse parsing itself, Aux=Applications to auxiliary tasks. Publication venues are in brackets.

Figure 1.3 Project overview highlighting the relationships and dependencies between projects, colour-coded by supervision signals, clustered according to their role in this thesis. Lang Mod=Language Modelling from raw text. Others=Question-Answering, MNLI. Publication venues are in brackets.

Figure 2.1 Partially overlapping sub-tasks to generate complete constituency discourse trees.

Figure 2.2 Constituency (left) to dependency (right) tree conversion for an example discourse tree with 6 EDUs following the conversion algorithm by Li et al. [93].

Figure 2.3 Shift-Reduce algorithm generating a tree structure as a sequence of shift and reduce actions on the stack (orange) and queue (yellow). EDUs in purple, internal nodes in blue.
Figure 2.4 Example of a sentiment annotated document tree structure:
[It could have been a great movie] [It does have beautiful scenery] [some of the best since Lord of the Rings] [The acting is well done] [and I really liked the son of the leader of the Samurai] [He was a likable chap] [and I hated to see him die] [But, other than all that, this movie is nothing more than hidden rip-offs].

Green=positive sentiment, Red=negative sentiment. N=Nucleus, S=Satellite. Adopted from [14].

Figure 2.5 Significance of high-level discourse constituency structures for dependency conversions.

Figure 2.6 Example of the increasing data sparsity issue for predicting higher-level discourse structures.

Figure 3.1 Example of a negative review in the Yelp'13 corpus:
[Panera bread wannabes] [Food was okay and coffee] [was eh] [Not large portions for the price] [The free chocolate chip cookie was a nice touch] [and the orange scone was good] [Broccoli cheddar soup was pretty good] [I would not come back].

Green=positive sentiment, Red=negative sentiment.

Figure 3.2 Left: First stage, training the MIL model on the document-level sentiment prediction task. Right: Second stage, using the neural MIL model to retrieve fine-grained sentiment and attention scores (star/triangle), used for the CKY computation to generate the optimal discourse tree.

Figure 3.3 MIL Network Topology. We omit the second subscript of the hidden representations $H_{w_i}$ and $H_{E_i}$ for readability.

Figure 3.4 All 14 projective discourse trees annotated with sentiment for a 5 EDU document (using a simplified colour scheme, green = positive, red = negative, grey = neutral, omitting the attention attribute).
Figure 3.5 Standard beam-search approach (left) picking the top $B = 2$ tree candidates with the smallest distance $|p_t - gl|$ in every CKY cell. The stochastic beam-search approach (right) calculates the Boltzmann–Gibbs distribution with the tree-coverage dependent temperature $\tau$, modulating the sub-tree sampling process of the tree candidates. (For readability, we only show a maximum of 4 sub-trees per CKY cell).

Figure 3.6 Performance trend over increasingly large subsets tested on RST-DT (left) and Instr-DT (right). Each sample is generated as the average performance over 10 random subsets, drawn from 3 independently created treebanks.

Figure 3.7 Teaser for a tree analyzed in Appendix A.1 containing 28 EDUs and neutral document-level sentiment.

Figure 4.1 Textual levels in a well-structured document and their influence on downstream tasks and available gold-label data.

Figure 4.2 Overview of our approach to generate high-level discourse structures on sentence-to-document-level (S-D) from distant topic segmentation supervision.

Figure 4.3 The high-level architecture of the neural topic segmentation model proposed in Xing et al. [174].

Figure 4.4 Greedy top-down tree generation example for a balanced, sentence-level discourse tree.

Figure 4.5 Problematic CKY aggregation without the use of discounting. All trees receive the same final score.

Figure 4.6 Positive example (wsj_1346) with 100% structural overlap between prediction and RST-DT gold-label annotation. Gold paragraphs $p_n$ are indicated by solid lines. Leaves are sentences.
Figure 4.7 Random example (wsj_1380) with 80.77% structural overlap between prediction (left) and RST-DT gold-label annotation (right). Gold paragraphs $p_n$ are indicated by solid lines. Leaves are sentences.

Figure 4.8 Negative example (wsj_1365) with 55.56% structural overlap between prediction (left) and RST-DT gold-label annotation (right). Gold paragraphs $p_n$ are indicated by solid lines. Leaves are sentences.

Figure 5.1 Overview of our tree inference method based on transformer self-attention matrices trained on the summarization task.

Figure 5.2 Small-scale example of a CKY parsing matrix (left) aggregated according to the attention matrix (right). To aggregate the sub-tree containing EDUs 3-5, the green ((34)5) and blue (3(45)) aggregations are considered based on their reciprocal self-attention scores in the self-attention matrix.

Figure 5.3 The Chu-Liu-Edmonds dependency tree generation algorithm with added sentence constraints. First, a sentence-level graph structure is defined (sub-figures (b)-(c)), before determining the intra-sentence aggregations (sub-figures (d)-(e)) and combining both parts (sub-figure (f)).

Figure 5.4 The Unlabeled Attachment Score of trees generated by the attention matrix per head on three datasets under two conditions with the $CNNDM-6-8$ model.

Figure 6.1 Small-scale example of the discourse extraction approach. Purple=EDUs, green=sub-word embeddings, red=input slices of size $t_{max}$, orange=PLM, blue=self-attention values, grey-scale=frequency count.
Figure 6.2 Constituency (top) and dependency (bottom) discourse tree evaluation of BERT (a) and BART (b) models on GUM. Purple=high score, Blue=low score. Left-to-right: self-attention heads, top-to-bottom: high layers to low layers. + indicates fine-tuning dataset.

Figure 6.3 PLM discourse constituency (left) and dependency (right) structure overlap with baselines and gold trees according to the original parseval and UAS metrics.

Figure 6.4 Nested aggregation approach for discourse similarity. (a) Grey cells contain same head, white cells indicate different heads. (b) Grey cells contain same model, white cells indicate different models. Column indices equal row indices.

Figure 6.5 BERT self-attention similarities on GUM. Top: Visual analysis of head-aligned (I & III) and model-aligned (II & IV) heatmaps. Yellow=high structural overlap, purple=low structural overlap. Bottom: Aggregated similarity of different models, same models, different heads and same heads showing the min, max and quartiles of the underlying distribution. *Significantly better than respective ≠Head / ≠Model performance with p-value < 0.05.

Figure 7.1 T-AE (Tree-AutoEncoder) topology for self-supervised tree inference. Inputs and outputs are dense encodings, \( \hat{\text{Enc}}_x \) represents reconstruction of spans. \( \varphi \) represents the pointer-network, \( g \sim G(0, 1) \) denotes the Gumbel-softmax (in the forward-pass with an additional straight-through computation, not shown here). Grey/Dashed components represent actions outside the computational path chosen. Red=Model inputs/outputs, blue=TreeLSTM cells, green=Discrete structure selector as in Choi et al. [21], yellow=Hidden sub-tree encodings, orange=Hidden state \( z \) of the complete input.
Figure 8.1 (a) Example Shift action – Top element of the queue gets moved to the top of the stack. (b) Example Reduce action – Top 2 elements of a stack are assembled into a sub-tree. (c) Example of input to our classifier, consisting of the RoBERTa string-encoding and structural features. Note that since EDUs 1 and 2 form a sub-tree, their spans are concatenated.

Figure 9.1 Sentiment annotated discourse tree example for non-trivial document containing 13 clauses with positive and negative constituents, taken from MEGA-DT [61]. The gold-label sentiment is negative. Dashed lines indicate supplementary information, solid lines indicate primary importance. Full text: [I’ve been a member for a month now,]_1, [and I guess]_2, [I’m able to get my workout done,]_3, [I do find myself annoyed]_4, [how cramped it is at the weights.]_5, [The equipment is older,]_6, [but it suffices.]_7, [I worked out at another studio on 3rd]_8, [and it was amazing!]_9, [It was so clean, nice, and new -]_10, [TV’s on every cardio machine.]_11, [When i came back to this location. ]_12, [I felt bad.]_13.

Figure 9.2 Our proposed sentiment analysis framework, containing three phases of training/inference to integrate discourse parsing and sentiment analysis.

Figure 9.3 Topology of our hybrid approach using sequential HAN components (blue) in combination with an attention-extended discourse-inspired TreeLSTM (green) aggregation on the dependency discourse tree. Inputs and outputs are red.

Figure 9.4 Top: Example transformation of a constituency tree to the respective dependency tree. Bottom: Conditional attention module, weighting the importance of child node LSTM encodings $LSTM_{cell}(h_{ei})$, given the initial head node hidden state $h_{head}$.
Figure 9.5 Accuracy and F1-score over document lengths aggregated into 5 bins on the Yelp’13 dataset.

Figure 10.1 Document wsj_0639 from the RST-DT corpus with inconsistent importance differences between N-S attributions. (The top-level satellite is clearly more central to the overall context than the lower-level satellite. However, both are similarly assigned the satellite attribution by at least one annotator). Top relation: Annotator 1: N-S, Annotator 2: N-N.

Figure 10.2 Three phases of our approach to generate weighted RST-style discourse trees. Left and center steps are described in Section 10.3, right component is described in Section 10.4. †=As in Huber and Carenini [61], ‡=As in Marcu [109], *=Sentiment prediction component is a linear combination, mapping the aggregated embedding to the sentiment output. The linear combination has been previously learned on the training portion of the dataset.

Figure 10.3 Three phases of our approach. Left/Center: Detailed view into the generation of weighted RST-style discourse trees using the sentiment analysis downstream task. Right: Sentiment discourse application evaluation.

Figure 10.4 Three phases of our approach. Left/Center: Detailed view into the generation of weighted RST-style discourse trees using the summarization downstream task. Right: Summarization discourse application evaluation.

Figure 10.5 Three phases of our approach. Left: Generation of W-RST-Sent/Summ discourse trees. Right: Linguistic evaluation.
Figure 10.6 Top: Sentiment Analysis accuracy of the W-RST model compared to the standard RST framework with different thresholds. Bottom: Average ROUGE score (ROUGE-1, -2 and -L) of the W-RST summarization model compared to different thresholds. Full numerical results are shown in Appendix C.1.
Figure A.1 Accurately predicted example, Gold-label polarity: 0.5, Predicted polarity: 0.345, Discourse: [it has been a hell of a work week], [and friday could not have come any sooner], [this week was rough especially with the announcement], [that we are officially in a recession], [and the ambiguity was hitting me from all sides], [man, did], [i need some distraction from all my worries], [so the bf ( bacon ) and i decided on thai], [but wanted to venture out from the norm], [and we are very glad], [the thai hut exceeded our expectations], [we were a little skeptical at first with so many lackluster reviews], [but we hit the jackpot on our night], [this place has a great vibe !], [we were seated immediately], [and staff was beyond courteous and attentive], [we were approached by a few staff members], [which gave us the feeling of true teamwork], [our server was attentive], [and even sparked up some conversation throughout our meal], [we started with a hot pot of chicken tom kha kai], [and this soup hit the spot], [my worries were vanishing with every spoonful], [they say], [chicken soup will cure a cold], [and that menudo will feed a hangover], [well, i], [now believe], [tom kha kai is the cure for the blues], [because it sure made me happy !], [for our main dish we shared the red chicken curry], [a little heavy on the red and green peppers but very tasty and was the perfect match with the soup], [so we will definitely be back], [and will be sharing this place with some of our closest friends], [i’ve already made lunch plans for next wednesday].
Figure A.2 Accurately predicted example, Gold-label polarity: $-0.5$, Predicted polarity: $-0.391$, Discourse: [stopped in here for a friday happy hour with co-workers], [the beer was decently], [priced for happy hour], [the appetizers were decently priced], [which would be awesome], [if they were good], [the chicken strips were terrible], [i have never eaten something so greasy and yet dry all at once], [they are beer battered (like fish)], [which could be good], [but the execution on this was terrible], [the outside was really greasy], [which took away all of the crispy goodness], [that usually happens], [when things are battered and deep fried], [the chicken itself was dry as a bone], [we also got an order of fries], [that came out cold], [and were just below mediocre], [the place was really warm], [which could be attributed to the summer heat], [but we were sitting inside], [so there is a fair assumption], [that air conditioning would be involved], [i’ll pass next time], [my coworkers are planning a trip here], [i’d be better off], [eating at mcdonald’s]
Figure A.3 Inaccurately predicted example, Gold-label polarity: 1, Predicted polarity: $-0.098$, Discourse: [upon first moving here 2 years ago], [i had the worse experience], [attempting to get an airbrush spray tan at this salon], [they had only 2 people at specific times], [that could spray you custom], [no problem showed up], [and the tech could not figure out how to use the gun], [so awkward enough], [him being a male], [and standing there naked, i to get my money back], [after waiting 20 min], [well a couple months back], [they ran a deal for versa], [which is a booth spray tan], [i love this booth], [it is like airbrushing but private], [and this spray tan absolutely does not smell or stain your sheets], [i found this], [upon leaving denver, co. and just], [until i saw it online on living deals for amazon. :)], [one down two], [to go], [all for $29 :) love !], [as far as the gym goes], [never used it!]

xxxii
Figure A.4  Inaccurately predicted example, Gold-label polarity: $-0.5$,
Predicted polarity: $0.085$, Discourse: [i hate having]$_1$, [to write a poor review for this joint!]$_2$, [the owner is a really great guy]$_3$, [and the service was excellent.]$_4$, [the place is decorated well]$_5$, [and has a clean finished look.]$_6$, [i really wanted to love the pudding]$_7$, [but it really didn’t work out for my wife and i. from first glance]$_8$, [the pudding was all very soupy]$_9$, [and while it tasted]$_{10}$, [okay, was not anything to write home about.]$_{11}$, [the shop is trying too hard]$_{12}$, [to be an ice cream or gelato setup.]$_{13}$, [i think all the flavors and take away from their core business model.]$_{14}$, [i think]$_{15}$, [they should focus on making the rice pudding more solid]$_{16}$, [and have a couple]$_{17}$, [warm pudding options.]$_{18}$, [i can envision a warm rice pudding with some nuts and raisins with some brown sugar or cinnamon on top]$_{19}$, [yum !]$_{20}$, [shoot for rich, creamy and full of flavor.]$_{21}$, [my best wishes go out to them]$_{22}$, [and hope]$_{23}$, [that the masses will enjoy it more than we did.]$_{24}$, [they are good folks]$_{25}$, [and deserve to be successful.]$_{26}$  233
Figure A.5  Inaccurately predicted example, Gold-label polarity: $-0.5$, Predicted polarity: $-0.006$, Discourse: [i’ve been here a couple of times in the past.], [usually at someone else’s suggestion.], [i can’t say], [that i recommend this place.], [unless you like], [your lunch served up with a lot of attitude.], [the lady], [that takes the orders at the counter], [is usually abrasive and rude.], [i am the type of person], [who will kill the meanest person with kindness.], [but there are places], [where i draw the line.], [so, i have drawn the line with rome’s pizza.], [the funny part about it all is], [that my line is often zig-zag and curvy.], [so i still go here], [when someone else wants to go.], [hehe.], [my friends like the abuse i guess.], [one friend says], [the lady is nice to him.], [the plus side], [of going here is the fact], [that they serve an average pizza by the slice with your custom toppings.], [they also make hoagies and some other dishes.], [they have a nice lunch special], [that includes soda for a few bucks.], [they also serve some typical american favorites like hot wings.], [i usually order the of pizza lunch special], [and get the unsweetened tea.], [i’m not sure], [why i forget.], [but their tea tastes horrible], [because the water from the fountain tastes terrible !], [but it never fails, i forget], [that i need to of water with me.], [all in all, this place is a dive.], [give it a try.]

Figure A.6  Positive example (GUM_bio_jesper) with 76.92% structural overlap between prediction (left) and GUM gold-label annotation (right).

Figure A.7  Random example (GUM_voyage_oakland) with 70.83% structural overlap between prediction (left) and GUM gold-label annotation (right).
Figure B.5  Top: Visual analysis of sorted heatmaps. Yellow=high score, purple=low score. Bottom: Aggregated similarity of same heads, same models, different heads and different models. *=Head/=Model significantly better than ≠Head/≠Model performance with p-value < 0.05.

Figure B.6  Our generated tree (left) compared to the gold-standard tree (right) for document wsj_1395

Figure B.7  Our generated tree (left) compared to the gold-standard tree (right) for document wsj_1198

Figure B.8  Our generated tree (left) compared to the gold-standard tree (right) for document wsj_1998
Glossary

Several important terms used in this thesis have been overloaded with inconsistent meaning in the literature. To be as clear as possible, we define their meaning in the context of this work by providing a set of definitions for the most essential terms.

**discourse** constitutes a locally and globally coherent piece of text (i.e., a document) with a well-defined syntactic, semantic and pragmatic structure, guided by the underlying communicative goal(s) of the author [74][148].

**discourse analysis/processing** describes the investigation and annotation of a complete discourse in regard to the underlying, communicative goal(s) driven by semantic and pragmatic relationships within a document [148].

**discourse information/annotations** are the output of performing a discourse analysis. Discourse information/annotations can be represented in manifold ways (e.g., structured or unstructured/descriptive), however, are most commonly expressed through holistic/partial trees or graph structures, connecting text spans according to their coherence relationship [148].

**discourse parsing** is the process of (computationally) obtaining structured discourse information/annotations following their coherence relations [74].
**discourse structures** refer to the structured sub-set of *discourse information/annotations*. As such, the term “discourse structure” refers to either structural discourse representations (e.g., trees, graphs) or, in the context of the RST discourse theory, to a plain constituency tree, without additional nuclearity and relation attributes [107, 148].

**discourse trees** describe specific *discourse structures*, where the *discourse* is represented as a complete or partial tree structure.

For further clarification on related terms, we refer readers to Stede [148] and Jurafsky and Martin [74].
Acknowledgments

First and foremost, I want to thank my supervisor Giuseppe Carenini for his efforts, patience and time in guiding me through the past four years of the Ph.D. program, for passionately discussing research problems, encouraging me in times of difficulties, while always keeping an eye on the broader goal.

I thank my supervisory committee (Giuseppe Carenini, Kevin Leyton-Brown and Leonid Sigal), my University Examiners (Rachel Pottinger and Miikka Silfverberg) and the External Examiner (Nicholas Asher) for the critical comments and helpful suggestions in their respective roles.

In addition, I want to extend my sincere gratitude to Xilun Chen and Armen Aghajanyan for their support and guidance during my short time interning at Facebook. Despite the short duration, the obtained experience has been pivotal in many regards.

I want to thank my colleagues and collaborators at UBC, which over the years have become friends. I want to thank Wen Xiao, Linzi Xing, Grigorii Guz, Hyeju Jang, Raymond Li, Felipe Gonzalez-Pizarro, Chuyuan Li and everyone else in the Natural Language Processing research group for your help, interesting discussions and all the great experiences inside and outside the lab.

I want to thank my closest friends during these difficult years, namely Jackson Tong, Hayden Leung, Kieran Haberstock, Nathan Tannar, Leilani Ruffini and Annie Yang. I could always count on you no matter what and I’m grateful for your support through some of the hardest times.

Lastly, I want to thank my parents, Siegfried and Cornelia Huber, and siblings, Tobias and Samira Huber, for their encouragement and help through-
out this time. Thank you for having faith in me, supporting me in all my
decisions and always having my back. This accomplishment would have not
been possible without you. Thank you.

I am grateful for the funding received for my Ph.D. degree from the
Natural Sciences and Engineering Research Council of Canada (NSERC),
the BC Graduate Scholarship (BCGS), the Four Year Doctoral Fellowship
and the Huawei Language & Speech Innovation Lab of Cloud BU.

In the end, if I have seen further, it is by standing on the shoulders of
Giants. – Isaac Newton
Introduction and Related Work
Chapter 1

Introduction

The field of Natural Language Processing (NLP) aims to better understand and generate natural language through automated systems. Researchers thereby study and build computational models using algorithms and linguistic formalisms to enable automated systems to perform real-world tasks involving human language. The field of NLP can traditionally be further subdivided into two sub-fields:

The area of Natural Language Understanding (NLU) extracts insights from textual data to form meaningful representations, e.g., structured representations, in form of graphs, trees, or sparse relations (such as co-reference relationships), generalized representations such as topic labels, sentiment scores or named entities, or, more recently, dense encodings. On the other hand, pure Natural Language Generation (NLG) systems use previously generated or naturally occurring representations to generate textual outputs. In line with these definitions, well-known tasks such as text classification (e.g., sentiment prediction) are aligned with the area of NLU, while others, such as image-captioning, constitute a pure NLG system. Despite some important tasks clearly falling into either the area of NLU or NLG, most real-world applications combine components of both (e.g., in a “text-to-text” framework), such as machine translation, summarization, and question-answering, just to name a few. In this thesis, we explore NLU-related problems and solutions concerning structured representations.
Further differentiating types of (linguistically inspired) structured knowledge representations, approaches can be broadly separated according to four partially overlapping and non-exclusive objectives:

- **Morphologic Representations**, primarily concerned with the structure of words and their compositionality

- **Syntactic Representations**, concerning the grammatical structure of text within sentences [22]

- **Semantic Representations**, encoding structures regarding the logical form of sentences and larger textual constituents [163]

- **Pragmatic Representations**, encoding the document-level structure of a discourse in regard to coherence, context and pragmatics [107, 130]

In this thesis, we focus on monologue text (as opposed to speech and dialogue text), investigating the semantic/pragmatic structure of natural language at the document level. The exploration of document-level semantic and pragmatic relationships thereby generally falls into the research areas of **discourse analysis** and **discourse parsing** [18, 68, 72, 149, 160, 183]. As such, discourse annotations have shown great potential for a variety of real-world tasks, such as text classification [69], sentiment analysis [14, 58, 119], summarization [109], argument mining [19] and natural language generation [10]. For example, tackling the downstream task of argument mining, structural representations have shown to support the reliable extraction of arguments from unstructured text, used for example in computational debaters [143]. For natural language generation, on the other hand, discourse awareness is of essence to properly plan and structure the generation of text [50].

Maybe even more important, **discourse analysis**, one of the most popular approaches to encode document-level structural information, plays a significant role in many contextualized models by enabling the effective use of global (document-level) features. In this context, the reliable extraction of semantic and pragmatic insights at the document level has shown to be one of the central objectives in the field of NLU and NLP.
Figure 1.1: Corpus size of commonly used discourse treebanks compared to datasets for the sentiment analysis and summarization tasks. The size of circles indicate dataset sizes, i.e. the number of documents.

Despite the wide range of downstream tasks and the great impact on document-level contextualizations, the manual generation of structured representations on the semantic and pragmatic level constitutes a severe bottleneck. In contrast to many other forms of meaning representations, such as topic or sentiment representations, where natural annotations exist, no naturally annotated data is available for document-level discourse structures. As a result, discourse processing is heavily influenced by linguistic research, defining the necessary frameworks to generate consistent annotations. However, building complex formalisms to describe language and its usage centred around humans leads to tedious and time-consuming annotation processes, in turn resulting in high annotations costs and small-scale datasets (see Figure 1.1 for a dataset-size comparison of discourse treebanks with common NLP tasks). While this holds true for linguistic formalisms at large, the annotation of semantic/pragmatic structures underlying discourse analysis is especially difficult, given the ambiguity and complexity of contextualized structures at document level.

As a result, to be able to leverage recent advances in the computational modelling of language, the data sparsity issue underlying the current discourse annotation process needs to be lifted. To overcome the prevalent data sparsity issue, we state three research questions as part of this work:

(RQ1) How can we effectively generate discourse annotations at scale,
without the current limitations inherent to the human annotation process? This is an important question given the current lack of training data in both size and domain, not scaling to the requirements of modern deep learning models. As a result, this research question aims at the development and evaluation of novel computational models and methods to obtain discourse annotations by replacing the human component, which is not scalable to the extent required by many deep learning approaches.

(RQ2) Can large-scale, automatically generated discourse annotations enhance real-world auxiliary tasks? Aiming to further quantify the potential of automatically generated large-scale discourse structures, we frame this question to assess the ability of our models and methodologies to predict the task-dependent organization of documents for downstream applications.

(RQ3) Can large-scale, automatically generated discourse annotations augment linguistic theories of discourse? Here, we explore the potential of our generated annotations to refine discourse theories with data-driven insights, given that existing discourse formalisms (e.g., RST) are inherently imperfect, make significant simplifications, and are likely not fully aligned with the use of natural language “in the wild” As a result, we investigate if our models and methodologies can effectively augment the weak spots of existing linguistic theories of discourse.

To address the research questions above, we state the central hypothesis of this thesis as:

\[ \text{Context-sensitive NLP tasks internally encode some notion of discourse information to prioritize, structure, and connect parts of the input.} \]

Build around the research questions and our central hypothesis, we propose a set of novel computational models and methods to automatically infer discourse annotations at scale. To this end, we relax the discourse treebank bottleneck, currently restraining modern deep learning approaches to utilize more linguistically inspired structural representations on document level and, in turn, enable NLP researchers to better integrate discourse annotations into state-of-the-art models. Figure 1.2 gives a visual overview of our con-
tributions aligned with supervision signals and objectives, pointing towards the relevant chapters and prior publication venues of our proposals. Figure 1.3 provides a more detailed view on the supervision signals, dependencies between projects, and their role in this thesis. We colour-code supervision signals in Figure 1.3, encode the training datasets through the database symbol and present the generated labels/structures as a hexagon, indicating project outputs. A legend defining the symbols used in Figure 1.3 is shown in the bottom right corner.

The remainder of this thesis is divided into three parts: Part I explores different tasks to infer discourse annotations through distantly supervised approaches, Part II investigates self-supervised methods, solely relying on unlabelled, plain textual data and Part III shows applications of our proposed generation methods for discourse parsing, downstream tasks and linguistic theories. With the 9 projects$^1$ presented in Figure 1.3, we show a broad range of novel models and methods to tackle the currently prevailing data sparsity issue in discourse parsing (left) and present an extensive analysis of our distantly supervised discourse inference approaches (right).

To summarize our contributions in the three parts of this thesis:

**Part I:** In the first part, we generate and evaluate discourse annotations obtained from distant supervision signals by exploiting large-scale datasets of related auxiliary tasks. We extract discourse information from sentiment annotations, topic labels and summarizations, targeting different components of complete discourse trees. We investigate the auxiliary tasks of sentiment analysis for locally inspired discourse structures (on low- and mid-levels, purple in Figure 2.1). In our experiments, we show clear improvements on the “inter-domain” discourse parsing task. Our discourse dataset and trained discourse parsers obtained from sentiment annotations are published at: [https://nlp.cs.ubc.ca/mega-dt](https://nlp.cs.ubc.ca/mega-dt). Despite its great applicability for short- and mid-range discourse structures, the auxiliary task of sentiment analysis is reasonably less aligned with long-range discourse relations holding between paragraphs and sections of a long document. Hence, we explore a plausibly more aligned auxiliary task in this scenario: topic segmentation (orange in

$^1$Please note that the EMNLP '19 and '20 projects are combined into a single chapter.
Figure 1.2: Thesis overview aligning chapters along the axis of supervision signals (as columns) and objectives (as rows). Ling=Application to improve linguistic theories, Parse=Application to discourse parsing itself, Aux=Applications to auxiliary tasks. Publication venues are in brackets.

Figure 2.1: Specifically, we show the usefulness of topic-segmentation inferred discourse structures on different tree levels, finding that while sentiment-related information is more expressive on low- and mid-levels of the tree, topic segmentation improves the performance on high levels, making the task a valuable extension to generate complete discourse trees. Orthogonal to the different levels of discourse structures, we explore the RST nuclearity attribute from the related auxiliary task of summarization by extracting discourse information from the transformer self-attention component. We find that when interpreting the residual attention between EDUs as their semantic and pragmatic “relatedness”, we are able to extract valuable nuclearity annotations using traditional tree generation algorithms.

Part II: In this part, we approach the problem of robust discourse tree generation in a self-supervised manner, exploring the ability of models to infer valid semantic and pragmatic relationships by exclusively exploiting unlabelled text. Specifically, we explore previously proposed, transformer-based, pre-trained language models (PLMs) in regard to their internal representation of discourse structures. Using our novel approach to infer discourse information for arbitrarily long documents, we find that the discourse tree structures captured in the self-attention matrices of BERT and BART are
consistent and general across a collection of fine-tuning tasks. In our second self-supervised discourse generation project, we directly tackle a key limitation of PLMs for modelling discourse: The lack of linguistic priors in the simple chain structure (i.e., the sequential nature of masked and auto-regressive models). To overcome this limitation, we extend the ability of language models to represent the hierarchical (discourse) structure of textual documents. As a result, we propose a novel, tree-style autoencoder trained on the language modelling task. In this initial work on discrete, tree-style language modelling, we find that our approach outperforms commonly used, linguistically supervised methods without making assumptions about the underlying data.
Part III: In this part, we propose a set of novel models and methods to apply our distantly supervised discourse structures to discourse parsing, downstream tasks and linguistic frameworks. More specifically: (i) We exploit large-scale “silver-standard” discourse treebanks for discourse parsing. Using the traditional shift-reduce approach in combination with neural classifiers, we infer general discourse structures in a pre-training/fine-tuning framework. Initially conditioning the model on automatically inferred discourse annotations and subsequently fine-tuning on small-scale, human-annotated structures thereby achieves state-of-the-art (SOTA) in-domain discourse parsing performance. (ii) Besides the usefulness for discourse parsing itself, we further explore the direct application of discourse structures obtained from sentiment supervision to the task of sentiment analysis. In our framework, going from sentiment annotations to sentiment predictions through distantly supervised discourse structures, we show the potential of “silver-standard” discourse trees to improve the downstream task of sentiment analysis, especially for long and nuanced documents. (iii) Lastly, we show the potential of computationally inferred, real-valued importance scores to improve the performance of downstream tasks (here: sentiment analysis and summarization) and confirm their ability to augment human annotations.
Chapter 2

Related Work

To put our work into context, we give background information on discourse analysis (§2.1), popular discourse datasets (§2.2), discourse parsing models (§2.3) and synergistic downstream tasks (§2.4) in the following sections.

2.1 Discourse Analysis

Discourse analysis is a traditionally linguistic area of research, aiming to better understand and model natural language centred around humans. In the past, multiple discourse theories have been proposed, focusing on different aspects of a complete discourse. In this thesis, we follow the RST framework as the main guiding theory to bridge the gap between the linguistic understanding of text and the application of large-scale machine learning models. We choose the RST framework over competing theories due to its extensive backing by empirical results, the potential to improve important downstream tasks and its wide acceptance as a core theory of discourse.

Developed to better describe the textual organization of complete and potentially complex documents, the Rhetorical Structure Theory (or short RST), was first introduced by Mann and Thompson [107] in 1988, constituting one of the main guiding theories for discourse analysis [18, 45, 100, 149, 183], discourse parsing [39, 68, 72, 92, 160, 180], and text planning [42, 50, 155]. To generate a complete discourse tree for a document, it is first separated
into so-called Elementary Discourse Units (or short EDUs), representing clause-like sentence fragments. Based on these atomic EDUs, progressively larger textual constituents are aggregated to form a discourse tree consisting of three components: (i) A projective\(^1\) constituency tree structure\(^2\) (shown in Figure 2.1), oftentimes referred to as the tree span. Within this tree, leaf nodes represent EDUs, while internal nodes combine their children into a single, joint constituent, based on the rhetorical coherence (i.e., two EDUs that are directly related are also closely connected within the tree structure, while EDUs that are part of different arguments are further removed in the discourse tree). (ii) Every local sub-tree consisting of a parent node and \(n\) child nodes is annotated with an additional, binary nuclearity attribute defining the local importance of the tree node (and the respective sub-tree) relative to its siblings (arrows connecting nodes in Figure 2.1). A tree node classified as a Nucleus is defined as of primary importance in the local relation, while a Satellite annotation indicates a supplementary role of the node and the underlying sub-tree. The RST framework further limits the possible

\(^1\)Projective trees contain no crossing edges when drawn on a plane.

\(^2\)Constituency trees (in comparison to dependency trees) represent a document by recursively combining textual spans (i.e., constituents) into larger subtrees.
nuclearity attributes of a local sub-tree to at least one Nucleus annotated
node and arbitrarily many Satellites (i.e., a sub-tree with all Nucleus children
is possible, however a node with only Satellite children is not well-defined).
(iii) Closely related to the structure and nuclearity attributes described above,
every internal tree node connecting either EDUs, larger text spans (i.e., sub-
trees) or a combination thereof, is assigned a relation attribute defining
the nature of the rhetorical connection, such as Elaboration, Contrast or
Motivation. The relation attribute between sub-trees is thereby defined as the
relationship of the respective sub-tree nuclei [116]. Following this definition,
the Antithesis relation shown at the top of Figure 2.1 holds between EDU_j
and EDU_j+1. Annotating the complete document with a single, connected
discourse tree, the RST framework is a prime candidate for tasks requiring
document-level semantic and pragmatic understanding. Besides the standard
constituency format of RST-style discourse trees shown in Figure 2.1, the
annotated constituency structures can be further converted into dependency
tree structures⁴ oftentimes preferred for downstream applications. Figure 2.2
shows an example transformation from a constituency tree to its respective
dependency tree. The transformation thereby follows the near isomorphic
conversion described in Li et al. [93], attaching dependents to their heads
following the nucleus-satellite structure, primed towards the left nucleus for
N-N relations⁵.

Despite using the RST discourse theory in this work, our models and
methods are generally theory agnostic and only require minor changes to be
applied to other theories, such as: (i) The lexicalized discourse framework
[162] for monologues (underlying the PDTB corpus), focusing on inter-
and intra-sentence discourse connectives [130]. In comparison to the RST
discourse theory, the lexicalized discourse framework does not create a single
discourse tree covering complete documents, but rather defines “shallow”
relations within sentences and between sentence pairs (however not connecting

---

⁴ In a dependency tree, every node refers to an input text span (i.e., an EDU), which
are connected in head-dependent relationships.

⁵ Please note that alternative constituency-to-dependency tree conversion algorithms
exist, e.g., presented in Hirao et al. [57].
sentence pairs with each other). (ii) The Segmented Discourse Representation Theory (SDRT) for dialogues [6, 7], using a graph-based framework tailored towards the important task of dialogue discourse analysis and (iii) The approach by Wolf and Gibson [167], generally following the RST theory, however, removing the tree-constraint for human annotators and, as a result, generating graph-style structures.

### 2.2 Discourse Treebanks and Metrics

#### 2.2.1 Discourse Treebanks

To enable the computational use of abstract discourse theories, discourse treebanks provide broadly accessible resources to model discourse structures for new and unseen documents. Regarding the RST discourse theory, the predominantly used data sources in English are the RST Discourse Treebank (RST-DT) [18], the Instruction Discourse Treebank (Instruction-DT) [149], and the Georgetown University Multilayer Corpus (GUM) [183].

---

[183]: The GUM treebank diverts from the standard RST annotation by introducing additional relation classes, while the Instr-DT corpus contains multi-rooted trees, both not defined in
The RST-DT dataset by Carlson et al. [18] is the largest and most frequently used English corpus to train RST-style discourse parsers. The treebank consists of 385 news articles extracted from the Wall Street Journal corpus, annotated with full constituency discourse trees, pre-segmented in a 90-10 train/test split.

The GUM treebank [183] constitutes a continuously growing RST-style discourse treebank of richly annotated texts. In the current version 7.3, the dataset contains 168 documents from 12 genres, annotated with full RST-style constituency and dependency discourse trees.

The Instructional corpus (short Instr-DT) by Subba and Di Eugenio [149] comprises of 176 documents in the home-repair domain, annotated with full RST-style discourse trees, separated into training and test set with a 90-10 split.

As described above, each of the three most popular RST-style discourse treebanks in English does not exceed a minuscule number of 400 combined documents in their training, validation and test splits. The reason for the prohibitively small number of available RST-style discourse trees is threefold: First, no naturally annotated discourse structures are readily available to train and evaluate systems on. As a result, discourse trees have to be obtained through manual annotations. Second, discourse structures (and structured representations at large) have an inherently high degree of freedom, making the manual annotation of these structures tedious, time-consuming and expensive. Third, the oftentimes ambiguous nature of semantic and pragmatic structures requires a deep understanding of the text at hand, demanding sophisticated linguistic expertise and preventing the annotations to be executed through crowd-sourcing efforts. Given these factors, available discourse treebanks are sparse and small, cover few domains, and are not sufficient to robustly and generally represent natural language.

The dependency representations are generated from the respective constituency trees using the conversion algorithm proposed by Li et al. [93].

Please note, that while we focus on the inference and application of discourse trees in
visual comparison of the three described discourse treebanks with commonly used datasets for the tasks of sentiment analysis and summarization.

### 2.2.2 Discourse Evaluation Metrics

The standard evaluation metric for RST-style discourse structures is an adoption of the parseval metric, originally proposed for syntactic parsing. As such, the *RST-parseval* adaption based on right-heavy binarized discourse trees has been traditionally used to compare discourse parsing performances.

Following RST-parseval, each node in the discourse tree, except the root node, is included in the evaluation. Further, individual nodes contain a single nuclearity attribute, describing their role in the parent context (either “Nucleus” or “Satellite”). Arguing against this artificially inflated scoring mechanism, Morey et al. [115] propose to compare model performances for RST-style discourse parsers using the *original-parseval* scores, with every node expressing the nuclearity attributions of its child nodes (e.g., “N-N”, “N-S” or “S-N”), including the root-node, but excluding leaf nodes from the computation. Given the common practice of using gold-segmented EDUs in the area of discourse parsing, the switch to the original parseval score shows a more accurate picture of the true parsing performance and has been slowly transitioned to in recent years.

### 2.3 Discourse Parsing

Initiated by elaborate discourse theories, annotation guidelines, and the generation of discourse treebanks, computational applications of discourse frameworks started to evolve, leading to the creation of a new research area called *discourse parsing*.

Discourse parsing describes the automatic inference of discourse annotations for new and unseen documents based on a pre-defined discourse framework. To make discourse analysis available for arbitrary documents, discourse parsers aim to generalize on the rules and formalisms defined by the underlying linguistic theory. The overall process of discourse parsing the English language in this thesis, our presented approaches are not language specific.
thereby broadly consists of two, mostly distinct, phases: In the first step, the process of discourse segmentation separates continuous textual documents into a sequence of non-overlapping Elementary Discourse Units (EDUs), with popular systems proposed by Feng and Hirst [39], Li et al. [91], and Wang et al. [161]. Subsequently, based on the clause-like sentence fragments, the process of RST-style discourse parsing automatically generates discourse annotations for arbitrary documents. Throughout the history of discourse parsing, researchers used a variety of diverse computational approaches to find underlying patterns in natural language based on a discourse framework. Since we are focusing on RST-style discourse parsing in this thesis, we present an overview of proposed models to predict discourse for the English language separated into supervised, distantly supervised and self-supervised approaches.

2.3.1 Supervised Discourse Parsing

In line with many tasks in NLP and Machine Learning (ML) at large, traditional discourse parsing approaches have been near-exclusively focused on supervised models, typically trained and tested within the same domain using human-annotated discourse treebanks. In contrast, however, to many popular NLP tasks aiming to understand and generate natural language, human-annotated discourse treebanks are sparse and small (as described in Section 2.2). Given the small scale of the three most commonly used corpora in English, most supervised discourse parsing approaches are heavily influenced by traditional, linguistically-inspired and more data-effective machine learning approaches. In their 2017 evaluation of supervised discourse parsers, Morey et al. [115] give an overview of competitive approaches, mostly focused on traditional discourse models like DPLP [68], gCRF [39], CODRA [72] and Li et al. [92]. Despite these models following different (traditional) modelling methodologies such as Conditional Random Fields (CRFs) [39, 72] and Support Vector Machines (SVMs) [55, 68], they all model discourse by subdividing the general problem of document-level discourse parsing into smaller, more approachable
sub-tasks. Specifically, models either separate the task “vertically”, training individual model components for sentence-, paragraph- and document-level [68, 72] or split the computation “horizontally”, predicting the structure, nuclearity and relation attributes individually [160].

Generally, models learn discourse trees either: (i) Top-down, splitting the document into non-overlapping text constituents starting from the representation of the complete discourse down to individual EDUs [94], or (ii) Bottom-up, starting from the discourse-segmented list of EDUs and aggregating adjacent units. The bottom-up approach is typically realized using either chart parsing approaches (e.g., CKY dynamic programming) [72, 93], through greedy methods [55], or with the more locally inspired linear-time shift-reduce framework [49, 51]. The shift-reduce framework (initially adopted from syntactic parsing) thereby generates tree-structures through sequential actions of either shifting an EDU from the queue onto the stack, or by aggregating (i.e., reducing) the top stack elements into a sub-tree, as shown in Figure 2.3. Besides traditional models, recent approaches slowly shifted to successfully incorporate neural networks into the task of
discourse parsing, by using dense representations with custom architectures [16, 92, 97, 180] or pre-trained language models (PLMs), such as ELMo embeddings [82], XLNet [120], BERT [86], RoBERTa [51] and SpanBERT [49].

With a variety of the above mentioned models used to improve important downstream tasks, such as sentiment analysis [14, 58, 119], text classification [69] and summarization [43], they all share a common weakness: The lack of domain adaptation. In other words, supervised approaches can effectively infer discourse structures in domains similar to the training data, however, their performance significantly drops when generating discourse trees for documents in previously unseen domains [59, 61]. Combined with the very limited amount of available domains and data points, the application of available supervised discourse parsers is clearly limited.

2.3.2 Distantly Supervised Discourse Parsing

Targeting the prevalent data sparsity issue limiting the applicability of fully supervised discourse parsers for robust and domain-independent discourse parsing, researchers started to generate discourse annotations without the explicit use of human-annotated discourse through distantly supervised approaches. The general intuition is thereby tailored around the assumption that auxiliary data from contextual NLP tasks have synergistic relationships with discourse analysis, as previously shown in Bhatia et al. [14], Hogenboom et al. [58], Ji and Smith [69], Nejat et al. [119] and Gerani et al. [43]. While no single task is likely to capture the full range of discourse-related phenomena, we believe that different auxiliary tasks at least partially align with valuable discourse information. Work along these lines has been previously conducted by Nishida and Nakayama [122], proposing a novel approach to infer RST-style discourse structures in a “linguistically” supervised manner\(^8\), heavily exploiting syntactic markers in combination with general linguistic priors. Despite the promising performance of their approach, the used methodology is still heavily dataset-specific and does not solve the important

\(^8\)We consider “linguistic supervision” as a distantly supervised methodology.
issue of domain-dependency and dataset overfitting. Karimi and Tang [75] and Liu et al. [102] tackle the same problem by using latent tree induction frameworks, inferring trees from the text classification and summarization tasks, respectively. In their approaches, task-dependent discourse trees are generated as part of the neural training process, improving the performance on the auxiliary task, and completely bypassing the human annotation issue. However, follow-up work by Ferracane et al. [40] indicates that the tree structures generated by Liu et al. [102] are often trivial and shallow, not well aligned with human-annotated discourse. Hence, the problem of robustly predicting discourse annotations, while bypassing the human annotation step, is still unsolved. As a result, we propose multiple novel methods to infer discourse without requiring explicit human discourse annotations from the auxiliary tasks of sentiment analysis, topic segmentation and summarization, effectively bypassing the human annotation step, as described in Part I.

2.3.3 Self-Supervised Discourse Parsing

The area of self-supervised (sometimes also called unsupervised) RST-style discourse parsing has been mostly overlooked in the past, due to generally inferior performance compared to supervised and distantly supervised approaches. However, not relying on annotated data, self-supervised models have shown great promise for many NLP tasks in the recent past [28, 89]. The most commonly used auxiliary task for self-supervised models is language modelling, applied in different ways using either the autoencoder objective [31], masking methods [28, 89] or causal (i.e. auto-regressive) [17, 132, 133] approaches (see Section 2.4.4 for further details on the self-supervised objectives). In the area of discourse parsing, recent self-supervised approaches either take advantage of pre-trained language models [85, 169, 187], use pre-trained syntactic parsers and linguistic knowledge [122], or exploit the document hierarchy based on recursively computed dissimilarity scores [82].

Despite these works showing the usefulness of self-supervised approaches for discourse parsing, all previous approaches cast the task into a “local” problem, using only partial information through the shift-reduce framework
(see Figure 2.3) [49, 51], natural document breaks (e.g. paragraphs [88]) or by framing the task as an inter-EDU sequence labelling problem on partial documents [86]. However, we believe that the true benefit of discourse information emerges when complete documents are considered, leading us to propose new approaches to infer discourse information from: (i) Self-supervised language modelling objectives in a more “global” manner, superseding the local proxy tasks and (ii) by encoding complete documents into a single tree structure using a novel tree-style autoencoder objective. With our new approaches for self-supervised discourse parsing, we generate and evaluate data-driven discourse structures for arbitrarily long documents in Part II.

### 2.4 Discourse Synergistic Downstream Tasks

Looking at the recent literature in the field of NLP, it becomes apparent that distantly and self-supervised models have great potential in many areas. For example, distant supervision has been shown valuable for tasks such as sentiment retrieval [108, 151] and emotion classification [1], while self supervision has been shown highly beneficial for pre-training language models, such as BERT [28], BART [89] and GPT-X [17, 133]. However, a core challenge for any distantly or self-supervised model is the choice of a suitable task from which to infer information.

To improve the task of discourse parsing through distant and self-supervision, we explore auxiliary tasks which have shown to be positively influenced by discourse information, such as summarization [43], sentiment analysis [14, 58, 119], text classification [69] and topic segmentation [70]. Given the previously shown, synergistic relationships from discourse parsing to these real-world tasks, we aim to exploit the relationship in the opposite direction. In the following sections, we motivate the use of our selected distantly and self-supervised auxiliary tasks, putting them into the context of previous work.
Figure 2.4: Example of a sentiment annotated document tree structure:

[It could have been a great movie]¹ [It does have beautiful scenery.]² [some of the best since Lord of the Rings.]³ [The acting is well done.]⁴ [and I really liked the son of the leader of the Samurai.]⁵ [He was a likable chap.]⁶ [and I hated to see him die.]⁷ [But, other than all that, this movie is nothing more than hidden rip-offs.]⁸. Green=positive sentiment, Red=negative sentiment. N=Nucleus, S=Satellite. Adopted from [14]

2.4.1 Sentiment Analysis

Sentiment Analysis is a popular, yet challenging, real-world task, oftentimes strongly influenced by the document structure. For example, as shown in Figure 2.4, the importance of different clauses is essential for the determination of the overall sentiment. In this example, looking at the sentiment of individual clauses, the underlying document is seemingly positive. However, taking the discourse structure of the review into account, it becomes evident that the document sentiment is negative. Adjusting to the contextual nature of the task of sentiment annotation, Kim [79] shows the effectiveness of convolutional neural networks for the task by exploiting local neighbourhoods. Yang et al. [177] followed shortly after with their Hierarchical Attention Network (HAN) model, proposing one of the first hierarchical models for text classification by separating the task at the sentence level. Following
these initial works investigating the potential of using rudimentary structural information, researchers started to examine more sophisticated discourse structures (specifically: RST-style discourse trees) for the task of sentiment analysis. As motivated in Figure 2.4, augmenting the sentiment analysis task with discourse information can reasonably enhance the ability of systems to predict the correct sentiment label. Exploring the potential of using discourse information to improve sentiment analysis has been previously studied in multiple lines of work [14, 58, 69, 119]. For example, the model by Ji and Smith [69] uses discourse trees generated by the DPLP parser [68] trained on RST-DT in a recursive neural network to predict sentiment for multiple corpora, while Bhatia et al. [14] use handcrafted discourse features to score clauses depending on the level in the discourse tree.

Researching the synergistic relationship in the opposite direction, Choi et al. [21] showed a promising approach to generate task-depended discourse trees solely relying on sentiment information. In their model, they make use of the Gumbel-Softmax [66] (also used in similar ways in Corro and Titov [25, 26]), allowing the neural network to make discrete decisions while still using standard approaches like back-propagation to optimize the model.

Along these lines, we investigate the synergies from sentiment analysis to discourse parsing through the use of distant supervision. We show that reasonable discourse trees can be extracted from sentiment-supervised models, surpassing completely supervised discourse parsers on the “inter-domain” discourse parsing task[9] where models are trained in one domain and tested in another domain (e.g., training a model on news documents and applying the trained system to documents in the home-repair domain). We further show that the inferred discourse trees are not only improving structural metrics for discourse prediction but in turn also enhance the sentiment analysis performance for long documents. Our approaches and results exploring the synergies between large-scale sentiment annotations and the task of discourse parsing is described in Chapters 3, 9 and 10.

[9]Oftentimes also referred to as “zero-shot” evaluation.
2.4.2 Topic Segmentation

Topic segmentation aims to uncover the latent topical structure of a document by splitting it into a sequence of thematically coherent segments. Due to the shortage of annotated training data, early topic segmentation models are mostly unsupervised. As such, they traditionally exploit surface feature sets, like lexical overlap [36, 53] or topical signals contained in sentences [32, 137]. More recently, the initial data sparsity issue has been lifted by large-scale corpora sampled from Wikipedia [5, 84], containing well-structured articles with section marks as ground-truth segment boundaries. As a result, neural-based, completely supervised topic segmenters [5, 84, 174] have been heavily explored, showing robust performance improvements, potentially due to their enhanced ability to encode semantic and pragmatic information compared to methods solely relying on surface features.

Exploring the auxiliary task of topic segmentation in the context of discourse parsing, we draw inspiration from the findings in Jiang et al. [70], showing that pre-trained, neural-based topic segmenters can benefit supervised discourse parsers for the important task of high-level document structure prediction. The prediction of high-level constituency discourse structures is thereby especially important and challenging, due to the large impact on dependency tree conversions (see Figure 2.5) and the even more dire data sparsity issue for high-level discourse structures (Figure 2.6). In comparison to previous work by Jiang et al. [70], augmenting a supervised discourse parser with information obtained from topic segmentation models, we explore the potential of directly inferring high-level discourse structures.
Figure 2.6: Example of the increasing data sparsity issue for predicting higher-level discourse structures.

from the output of neural topic segmentation models, completely bypassing the data sparsity issue of supervised models in Chapter 4.

2.4.3 Summarization

The research area of summarization is typically divided into abstractive and extractive summarization methods, generating natural language summaries and selecting the most salient sentences of a document, respectively. With both types of summarization tasks having a large variety of use cases, abstractive summarization entails an additional NLG component, otherwise replaced by the sequence-labelling objective in extractive settings. With the synergistic connection from discourse parsing to extractive summarizations previously shown in the literature [109], we also focus on the extractive case in this thesis. Specifically, Marcu [109] pioneered the idea to directly apply RST-style discourse parsing to the extractive summarization task and empirically shows that RST discourse information can benefit the summarization performance by simply extracting EDUs along the nucleus path. This initial success was followed by further work leveraging discourse parsing in summarization by McDonald [112], Hirao et al. [57], and Kikuchi et al. [77]. More recently, the benefits of discourse for summarization have also been confirmed for neural summarizers, e.g. in Xiao and Carenini [170] and Cohan et al. [24] using the structure of scientific papers (i.e. sections), and in Xu et al. [176] by successfully incorporating RST-style discourse and co-reference information in the BERTSUM summarizer [98]. Interestingly, it has also
been shown that discourse enables the specification of simpler neural summarizers, without affecting their performance [171]. Similarly, Liu et al. [102] attempted to infer discourse structures from attention mechanisms, while training a neural model on the auxiliary task of summarization. However, no comparison against ground-truth discourse trees has been conducted, leaving the potential of their approach for the task of discourse parsing ambiguous.

Inspired by these findings, suggesting that transformer-based summarization models learn effective discourse representations, we explore if useful discourse structures can be inferred from transformer self-attention weights in Chapter 5.

2.4.4 Language Modelling

As presented above, a variety of valuable auxiliary tasks have previously shown to have synergistic relationships with discourse annotations, opening up great potential to overcome the crucial limitation of supervised discourse parsers depending on small-scale, human-annotated datasets. However, as per definition, distantly supervised approaches use specific auxiliary tasks to infer discourse structures. While this is a valid strategy, showing evidence for its effectiveness through performance gains on downstream tasks [97, 102], the inferred discourse structures are inherently task-depended. To generate more general discourse trees using distant supervision, a variety of diverse auxiliary tasks need to be reasonably combined, as individual tasks likely can not capture the full range of discourse phenomena.

In comparison to distantly supervised models, self-supervised approaches do not focus on a particular aspect of the data but directly exploit raw text. As such, language modelling presents a prime opportunity to extract general discourse information, not tied to a specific task. In this thesis, we propose the first set of approaches to exploit this vast and readily available data to improve the task of document-level self-supervised discourse parsing, describing two promising objectives in Chapters 6 and 7 based on pre-trained language models and autoencoder language modelling, respectively.
Pre-Trained Language Models

Pre-trained language models (PLMs) are at the base of many recent state-of-the-art (SOTA) approaches in the field of Natural Language Processing (NLP). As such, PLMs pushed the field from specific architecture designs towards the general framework of pre-training and fine-tuning. In this framework, a transformer network is first pre-trained on the task of either masked language modelling (e.g., [28, 89, 101], inter alia) or using a causal (auto-regressive) language modelling objective (e.g., [17, 132, 133], inter alia). During pre-training, the models aim to learn the general structure of language by either predicting left-out (masked) tokens or the next token of a sequence, given the left-side context. PLMs can be further classified into encoder-only (e.g., BERT [28]), decoder-only (e.g., GPT-X [17, 133]) and encoder-decoder (e.g., BART [89]) architectures.

Given the promising results of PLMs on a large variety of downstream tasks (e.g., sentiment analysis [178], summarization [184] and text classification [28]), some notion of implicit semantic and pragmatic knowledge is potentially stored in the dense representations of the model (previously investigated in [85, 124, 169, 187]). Using this intuition, recent work started to explore the alignment of PLMs with discourse information. Along those lines, Wu et al. [169] present a parameter-free probing task for both, syntax and discourse. Further, Zhu et al. [187] use 24 hand-crafted rhetorical features to execute three different supervised probing tasks, showing promising performance of the BERT model. Similarly, Pandia et al. [124] aim to infer pragmatics through the prediction of discourse connectives by analyzing the model inputs and outputs and Koto et al. [85] explore the discourse information in PLMs through seven supervised probing tasks, finding that BERT and BART contain the most information related to discourse.

As a result, recent approaches in discourse parsing are also shifting towards successfully incorporating a variety of PLMs into the process of discourse prediction, such as ELMo embeddings [82], XLNet [120], BERT [86], RoBERTa [51] and SpanBERT [49]. Despite these early works showing the usefulness of PLMs for discourse parsing, they cast the task into a “local”
problem, using only partial information through the shift-reduce framework (see Figure 2.3 for a small-scale example) [49, 51], natural document breaks (e.g. paragraphs [83]) or by framing the task as an inter-EDU sequence labelling problem on partial documents [86]. However, we believe that the true benefit of discourse information emerges when complete documents are considered, leading us to propose a new approach in Chapter 6, connecting PLMs and discourse structures in a “global” manner, superseding local proxy-tasks.

**Autoencoder Language Modelling**

Within the last decade, general autoencoder frameworks have been frequently used to compress data [147]. More recently, sequential autoencoders have been applied in the field of NLP [90], with many popular approaches (e.g., sequence-to-sequence models [150]) having strong ties to this class of methodologies. Based on the promising results of sequential autoencoders, researchers started to compress and reconstruct more general structures in tree-style models, such as Chen et al. [20], showing that with available gold-standard trees, the programming-language translation task from CoffeeScript to JavaScript can be learned with a tree-to-tree style neural autoencoder network. Furthermore, variational autoencoders have been shown effective for the difficult task of grammar induction by Kusner et al. [88].

While both previously mentioned applications for tree-style autoencoder models require readily available tree structures to guide the aggregation process, another line of work by Socher et al. [144] overcomes this requirement by using the reconstruction error of an autoencoder applied to every two adjacent text spans as an indicator for syntactic correctness within a sentence. In their model, Socher et al. [144] combine the tree-inference objective with the autoencoder topology, training a self-supervised tree-structured model, which is subsequently fine-tuned on a small-scale supervised dataset. Fully self-supervised and more holistically inspired, the DIORA approach proposed by Drozdov et al. [31] employs a neural version of the inside-outside algorithm [9]. The authors thereby encode and decode sentences using a tree-style
autoencoder to infer syntactic structures by joining possible sub-trees using a soft attention mechanism.

As indicated above, the general task of autoencoder-style tree inference has been mostly explored on sentence level, for instance in Socher et al. [144] using the reconstruction error, in Drozdov et al. [31] using the inside-outside algorithm, by applying reinforcement-style learning [179], or using the more traditional CKY methodology [106] for syntactic parsing. In comparison to these approaches, we propose a novel and fully differentiable methodology to generate discourse structures following the intuition from previous work in syntactic parsing, exploiting a tree-style autoencoder framework on document-level in Chapter 7.
Part I

Discourse Inference from Distant Supervision
In Part I, we explore novel approaches to generate discourse trees by substituting the human annotation signal, commonly used for computational approaches to discourse analysis, with naturally annotated data from auxiliary tasks. Naturally annotated data, with supervision signals provided by the original author, present the most reliable and strongest possible supervision signal. Here, we aim to exploit this available, high-quality data through distant supervision (i.e., supervision signals from tasks other than discourse itself), to generate large-scale, “data-driven” discourse structures for arbitrary text. Compared to “after-the-fact” third party annotations (e.g., discourse trees), natural supervision signals capture the authors true intention, directly reflected in the annotation itself. For example, a product or movie review alongside a star rating, both generated by a verified customer, are indisputably aligned and give reliable insights into the authors true opinion underlying the textual review. On the other hand, the retrospective annotation of RST-style discourse structures by third-party linguists generally provides lower-quality supervision signals. For example, regarding the binary nuclearity attribute, 16% of all assignments in RST-DT [18] are inconsistent [64]. Using this intuition, along with a number of previous works showing the synergistic relationship between sentiment analysis [14, 58, 119], topic segmentation [70], and summarization [43] with their underlying discourse structures, we extract discourse from auxiliary supervision signals through distant supervision.

Following the popular book “Speech and Language Processing” (3rd Edition) by Jurafsky and Martin [74], a text created by concatenating random sentences from varying sources will most likely not lead to a coherent discourse, since a proper discourse is locally and globally coherent. Here, we describe two distantly supervised tasks focusing on the local and global coherence of discourse structures (also called discourse spans), respectively. Most aligned, however not limited to, short- and mid-range discourse structures (presented in purple in Figure 2.1), we explore the task of sentiment analysis as an auxiliary task to infer the local coherence structure following the RST discourse theory in Chapter 3. In contrast to that, we primarily aim at high-level structures of the complete discourse tree using the auxiliary task of topic segmentation in Chapter 4. Next, moving beyond the focus on
plain discourse structures, we explore the auxiliary task of summarization, previously shown to be positively influenced by the nuclearity attribute [109], in Chapter 5.
Chapter 3

Discourse Inference from Sentiment Annotations

A version of this chapter has been published at EMNLP 2019 [59] and EMNLP 2020 [61]. In both works, I was the lead investigator, responsible for all major areas of concept formation, statement of research questions, data collection, implementation as well as paper composition. Giuseppe Carenini was the supervisory author on both projects, involved in concept formation, discussions, and paper composition.

3.1 Motivation

As described in Chapter 2, discourse parsing is an important Natural Language Processing (NLP) task, aiming to uncover the hidden structure underlying coherent documents. In line with Section 2.1, we follow the Rhetorical Structure Theory (RST) [107] in this chapter, shown to enhance a variety of key NLP downstream applications, including the task of sentiment analysis [14, 58, 119]. Consequently, to overcome the prevalent data sparsity issue described in Section 2.2, we propose a novel approach using distant supervision from the auxiliary task of sentiment classification. This way, we generate abundant RST-style discourse structures, subsequently evaluated against gold-standard discourse trees.
Figure 3.1: Example of a negative review in the Yelp’13 corpus:
Panera bread wannabes. [Food was okay and coffee] [was eh.] [Not large portions for the price.] [The free chocolate chip cookie was a nice touch] [and the orange scone was good.] [Broccoli cheddar soup was pretty good.] [I would not come back.]

The synergy from RST-style discourse parsing to sentiment analysis established in prior work [14, 58, 119] are thereby intuitive to grasp, given that a discourse describes the organizational structure of the document, an important cue to determine the author’s sentiment. Anecdotal evidence for this synergy is presented in Figures 2.4 and 3.1, showing the RST discourse annotation of two real-world examples, in which the overall amount of negative sentiment does not outweigh the number of positive words. However, looking at the RST-style discourse tree, the true sentiment of the overall documents becomes apparent.

Given this intuition on the role of discourse annotations for the task of sentiment analysis, the question regarding the directionality of the observed synergies naturally arises. Since the textual organization of a document can help sentiment prediction, can sentiment-related information also be used
to infer discourse annotations? To answer this question, we investigate the synergistic effects in the opposite direction, using sentiment analysis to create discourse structures.

Specifically, we combine a neural variant of the multiple-instance learning framework (MILNet) [4], with an optimal CKY-style tree generation algorithm [74]. In a first step, MILNet computes fine-grained sentiment values and un-normalized attention scores [69] on EDU-level, by solely relying on distant supervision signals from document-level annotations, abundantly available from several published open-source datasets (e.g., Yelp’13 [154], IMDB [29], Amazon [185]). Then, the sentiment values and attention scores are aggregated to guide the discourse tree construction, optimized on the document gold-label sentiment, using optimal CKY-style parsing. This way, we generate a new “silver-standard” discourse dataset based on the (distant) sentiment supervision signal. We call the base version of the resulting treebank MEGA-DT\textsubscript{base}, containing $\approx$100,000 documents with $\leq$20 EDUs.

Based on the MEGA-DT\textsubscript{base} architecture, we further propose two extensions to (i) make the approach more scalable, allowing for documents of arbitrary length and (ii) incorporate the nuclearity attribute, oftentimes critical in informing downstream tasks [69, 110, 142]. Both extensions are inspired by the recent success of heuristic search in NLP tasks involving trees (e.g., Fried et al. [41], Mabona et al. [104]), leading us to develop a beam-search strategy employing an exploration-exploitation trade-off, as commonly used in reinforcement-learning (RL) [129].

Remarkably, by following our approach presented in this chapter, any large sentiment annotated corpus can be turned into a “silver-standard” discourse treebank to train domain/genre-specific discourse parsers. As a case study for this process, we annotate, evaluate and publicly release a new discourse-augmented Yelp’13 corpus [154] called MEGA-DT\textsuperscript{1}, comprising of $\approx$250,000 documents, over two orders of magnitude more than any existing gold-standard discourse treebank, by solely leveraging the corpus’ document-level sentiment annotation.

\textsuperscript{1}Our new Discourse Treebank and the code to generate further “silver-standard” discourse treebanks can be found at: https://nlp.cs.ubc.ca/mega-dt
To evaluate the quality of our final MEGA-DT corpus on the task of discourse parsing, we train the top-performing parser by Wang et al. [160] on our MEGA-DT dataset and compare its performance with the same parser trained on previously proposed treebanks. Specifically, we compare our final MEGA-DT dataset against the MEGA-DT\textsubscript{base} version with \(\approx 100,000\) documents of \(\leq 20\) EDUs, as well as two standard human-annotated corpora in the news domain (RST-DT) [18] and the instructional domain [149].

Our results indicate that while training a parser on MEGA-DT does not yet match the performance of fully supervised training and testing on the same treebank (intra-domain), it does push the boundaries of what is possible with distant supervision. In most cases, training on MEGA-DT delivers statistically significant improvements on the arguably more difficult and useful task of inter-domain discourse prediction, where a parser is trained on one domain and tested/applied to another.

### 3.2 Related Work

The most closely related line of work is RST-style discourse parsing, with the goal to obtain a complete discourse tree, including structure, nuclearity and relations\(^2\). Along this line, a large number of highly diverse discourse parsers have been published in previous work. In this chapter, we follow the parser proposed by Wang et al. [160] due to its top performance and clear separation of the structure/nuclearity and relation classification modules, well-aligned with our experiments excluding the relation attribute.

The second line of related work is regarding the auxiliary task used for distant supervision: sentiment analysis. Previous studies, e.g., Bhatia et al. [14], Ji and Smith [69] have shown that sentiment prediction can be enhanced by leveraging discourse information, as the tree structure influences the significance of clauses in the document (see Figures 2.4 and 3.1\(^3\)). Here, we exploit the relation between sentiment analysis and discourse parsing

\(^2\)For more information on the RST discourse theory and top-performing discourse parsers, please refer to Sections 2.1 and 2.3 respectively.

\(^3\)Further related work on the relationship between discourse parsing and the downstream task of sentiment analysis can be found in Section 2.4.1.
in the opposite direction, using sentiment annotations to create discourse structures.

The third line of related work infers fine-grained information from coarse-grained supervision signals using machine learning. Due to the lack of annotated data in many domains and for many real-world tasks, methods to automatically generate reliable, fine-grained data labels have been explored for many years. One promising approach in this area is Multiple Instance Learning (MIL) [76]. The general task of MIL is to retrieve fine-grained information (called instance-labels) from high-level supervision (called bag-labels), using correlations of discriminative features within and between bags. With the recent rise of deep learning, neural MIL approaches have also been proposed [4], showing promising results in capturing EDU-level sentiment information as well as the relative importance of EDUs. In this chapter, we adapt the MILNet model [4] to generate fine-grained information, allowing us to derive discourse trees from corpora with document-level sentiment annotations.

The last stream of related work is on leveraging heuristic search algorithms for trees. For syntactic parsing, Vinyals et al. [158] and Fried et al. [41] show that a static, small beam size (e.g. 10) achieves good performance on the tree generation task, while Dyer et al. [34] even deliver promising results by using a greedy decoding strategy. As a recent example in the area of discourse parsing, Mabona et al. [104] successfully combine standard beam-search with shift-reduce parsing using two parallel beams for shift and reduce actions. Overall, recent work shows that beam-search approaches and their possible extensions can effectively address scalability issues in multiple parsing scenarios. In this chapter, we extend the standard beam-search approach with a stochastic exploration-exploitation trade-off, as used in Reinforcement Learning, where signals also tend to be sparse and noisy.
3.3 General Discourse Tree Inference from Sentiment

Drawing intuition from previous work using discourse parsing to enhance sentiment analysis, the central hypothesis of this thesis states that such synergies hold bidirectionally. The anecdotal evidence shown in Figures 2.4 and 3.1 supports this assumption, showing that EDUs which are essential to the overall communicative goal of the author (here with sentiment used as a proxy) are likely to be close to the tree root.

To generate a large number of discourse structures through distant supervision from sentiment, we propose a four-step approach, shown in Figure 3.2. On the left, we illustrate the first stage, where for each document in the dataset, (a) the document is segmented into EDUs and (b) our adaptation of MILNet is trained on the document-level sentiment. Next, shown on the right of Figure 3.2, we again (a) segment the document into EDUs and use (b) the MIL network to generate fine-grain sentiment and importance scores. Then in (c), we prepare those scores to be used in (d), a CKY-like parser, which generates an optimal RST discourse tree for the document, based on the EDU-level scores and the gold-label document sentiment.
3.3.1 Segmentation and Preprocessing (a)

We initially separate the sentiment documents into a disjoint sequence of EDUs. The segmentation is obtained using the discourse segmenter by Feng and Hirst [38] as generated and published by Angelidis and Lapata [4]. We preprocess the EDUs by removing infrequent- and stop-words and subsequently apply lemmatization.

3.3.2 Multiple-Instance Learning (MIL) (b)

Our MIL model is closely related to the methodology described in Angelidis and Lapata [4], as well as the approaches by Yang et al. [177] and Ji and Smith [69]. The computation is based on the initial segmentation described in Section 3.3.1 and is shown in further detail in Figure 3.3.

Our model consists of two levels of Recurrent Neural Networks (RNN) inspired by Yang et al. [177] as well as a sentiment and attention module. The computational flow in the model is defined from bottom to top in Figure 3.3. In a first step, the sparse one-hot word representations are transformed into dense vector representations \( w_i \) using a pre-trained GloVe word embedding matrix [128]. The dense word representations of a single EDU \( E_i = (w_j, ..., w_k) \) are used as the sequential input for the EDU-level RNN, implemented as a bi-directional GRU module with a standard attention mechanism [8]. The attention-weighted hidden-states \( R_{w_i} = H_{w_i} \ast A_{w_i} \) are concatenated along the time axis to represent \( R_{E_i} \). The second RNN on document-level subsequently uses the distributed EDU representations \( (R_{E_1}, ..., R_{E_s}) \) as inputs for the neural network. Based on the sequence of computed hidden-states \( (H_{E_1}, ..., H_{E_s}) \) in the bi-directional GRU network, two parallel model components are executed, as follows:

The non-competitive attention score module was proposed by Ji and Smith [69] to leverage discourse structure for sentiment prediction. By following the same intuition, we replace the softmax activation on the attention weights by a sigmoid function. This way, each attention weight \( A_{E_i} \) is still limited within the range \((0, 1)\), but the sum of all attention scores
Figure 3.3: MIL Network Topology. We omit the second subscript of the hidden representations $H_{wi}$ and $H_{Ei}$ for readability.

is not necessarily bound by 1. We use the attention weight $A_{Ei}$ as the importance scores of EDU$_i$.

The sentiment score module is also executed directly on the hidden-states ($H_{E1},...,H_{Es}$) generated by the document-level RNN. To be able to interpret the dense hidden representations as sentiment predictors, we use a single feed-forward neural network layer $S$ with $|C|$ neurons, representing the disjoint sentiment classes ($C_1,...,C_m$) in the dataset. We add a sigmoid activation $\text{sigm}$ after the feed-forward layer to obtain the final, internal EDU sentiment prediction:

$$S_{Ei} = \text{sigm}(S(H_{Ei}))$$

(3.1)

The output of the two parallel modules is multiplied EDU-wise and summed up along the time axis to calculate the final sentiment prediction of

---

4In the Yelp’13 dataset, the feed-forward operation $S$ results in 5 real-valued outputs.
our MILNet model as:

\[ O_D = \sum_{E_i \in D} S_{E_i} \ast A_{E_i} \]  \hfill (3.2)

To train our MILNet model, we use the cross-entropy loss function to compare \( O_D \) with the gold document-level sentiment label of the review and train using the Adadelta optimizer [182]. By separating the sentiment and attention components and directly computing the final output based solely on these two values, the neural network implicitly learns the sentiment and attention scores on EDU level as a by-product of the document-level prediction. For more information on this technique, we refer interested readers to Angelidis and Lapata [4].

The hyper-parameter setting of our model follows the implementation of previous work whenever possible. We use a batch size of 200 documents and train the model for 25 epochs. The bidirectional GRU layers contain 100 neurons and the model inputs are preprocessed by limiting the number of EDUs in a document to 150 and defining the maximum length of an EDU to be 20 words. With these settings, we capture over 90% of the data by significantly decreasing the training efforts. We apply 20% dropout on the internal sentiment layer.

3.3.3 Information Extraction and Transformation (c)

Once our MILNet adoption is fully trained, we extract the attention \( A_{E_i} \) and sentiment score \( S_{E_i} \) for each EDU \( E_i \) in a document. However, while each \( A_{E_i} \) is already a scalar, \( S_{E_i} \) is a vector with \( |C| \) elements, one for each sentiment class \( C_i \). In order to effectively combine the attention and sentiment scores for further processing, we transform \( S_{E_i} \) into a scalar polarity score \( pol \), centred around 0 and uniformly distribute the \( |C| \) sentiment classes within the interval of \([-1, 1]\). For instance, if \(|C| = 5\), this would result in the five classes \( pol_{coeff} = [-1, -0.5, 0, 0.5, 1] \). The polarity \( pol_{E_i} \) of EDU \( E_i \) is computed by calculating the element-wise product of the sentiment score...
and the uniform distribution $\text{pol}_{\text{coeff}}$.

$$\text{pol}_{E_i} = \sum_{c \in C} S_{E_i}^{(c)} \text{pol}^{(c)}_{\text{coeff}}$$

(3.3)

We transform the gold labels equally, to keep the representations consistent. With the polarity scores $\text{pol}_{E_i}$ replacing the original sentiment scores $S_{E_i}$, a neutral sentiment document now receives a sentiment polarity of 0, while heavily positive or negative EDUs are mapped onto the scores +1 and −1 respectively. This way, the obtained attention scores $A_{E_i}$ and the calculated polarities $\text{pol}_{E_i}$ can be combined to create a weighted sentiment score with high attention values resulting in stronger polarities.

### 3.3.4 CKY Tree Generation (d)

The final step in our approach takes the tuples of EDU-level attention scores and the generated polarities from the MILNet model to create a set of possible discourse trees. We then select the discourse tree that most precisely computes the overall sentiment of the document. To find the globally best tree, we compute all possible tree structures (with some constraints) using a dynamic programming approach closely related to the widely used CKY algorithm [74].

To create discourse trees bottom-up using CKY, we define the necessary aggregation rules for local trees. For each binary sub-tree, we need to define a function $p(c_l, c_r)$ on how to aggregate the information of the two children $c_l$ and $c_r$ to represent the parent node $p$. For sentiment, we use the intuitive attention-weighted average of the children’s sentiments, defined by:

$$p_s(c_l, c_r) = \frac{c_{l_s} * c_{l_a} + c_{r_s} * c_{r_a}}{c_{l_a} + c_{r_a}}$$

(3.4)

This way, the parent sentiment does not only depend on the sentiment of its children but also on their relative importance. For the attention computation, we consider three different aggregation functions:

(i) The sum of the children’s attention (Equation 3.5). This way, the combined importance of the children is inherited by the parent node, making the node
as important as the combination of all sub-nodes.

\[ p_{asum}(c_l, c_r) = (c_{la} + c_{ra}) \times (1 - \lambda) \]  

(3.5)

\( \lambda \) thereby represents a damping factor to penalize lower sub-trees, empirically chosen to be 1% using grid-search.

(ii) The maximum of the children’s attention.

\[ p_{amax}(c_l, c_r) = \max(c_{la}, c_{ra}) \]  

(3.6)

As shown in Equation 3.6, the attention of the parent node is calculated as the maximum attention value of the two children. This aggregation function follows the intuition that the parent node is only as relevant as the most important child node.

(iii) The average of the children’s attention.

\[ p_{aavg}(c_l, c_r) = \frac{c_{la} + c_{ra}}{2} \]  

(3.7)

This aggregation function (Equation 3.7) assigns the average importance of the two children to their parent.

To create RST-style discourse documents, which can be used by existing parsers, we have to provide additional nuclearity and relation labels for every tree node. At this point, we assign the nuclearity attribute solely depending on the attention value of the children nodes, making the following binary decision:

\[
cl_n = \begin{cases} 
"Nucleus", & \text{if } c_{la} \geq c_{ra} \\
"Satellite", & \text{otherwise}
\end{cases}
\]  

(3.8)

Please note that this initial approach cannot assign “Nucleus-Nucleus” attributes, a significant weakness, which we explicitly address in Section 3.4.3.

Finally, for the necessary rhetorical relation attribute, we assign a placeholder “span” relation to every node.

Due to the high complexity of the optimal CKY algorithm following the
Catalan number\(^5\), we introduce two constraints on the generated discourse trees:

- We prohibit inter-sentence relations unless the complete sentence is represented by a single node, as shown to capture the vast majority of discourse relations by Joty et al. [72].

- We only process documents with \(\leq 20\) EDUs per document\(^6\).

With the aggregation functions and restrictions described above, we run the CKY-style dynamic programming approach and compare the sentiment at the root node of each of the complete discourse trees with the dataset gold label for the document. The discourse tree with the smallest distance from the gold label is selected as the discourse representation of the document (see Figure 3.2, right) and saved in a serializable RST-DT format. This way, we generate a new “silver-standard” discourse dataset containing 100k discourse trees called MEGA-DT\(_{\text{base}}\).

### 3.4 Predicting Discourse Structure and Nuclearity for Arbitrary Documents

The previously described process of selecting the best discourse tree manifests the most complete solution to this problem. While this approach is, in general, useful, the computational complexity quickly exceeds modern hardware, requiring the implementation of unwanted constraints to be computationally feasible, as shown in Section 3.3.4. A small-scale example visualizing the explosion of the computational complexity in the full CKY setting is shown in Figure 3.4. As a result, the previously presented approach, generating MEGA-DT\(_{\text{base}}\), needs to be improved in two fundamental ways: (i) Making the solution more scalable. Since its space complexity grows with the Catalan number \(C_n = \frac{1}{n+1} \binom{2n}{n}\) for trees with \(n + 1\) EDUs, it can otherwise only be applied to short (\(\approx \leq 20\) EDUs) documents (see the bottom row in Table 3.1), making it impractical for many real-world datasets containing longer

\(^5\) All experiments are executed on an Intel Core i9-9820X, RTX 2080 Ti, 128 GB RAM

\(^6\) We lift this restriction in Section 3.4
documents, such as the Yelp ‘13 [154], IMDB [29] or Amazon Review dataset [185]. (ii) Expand the approach to additional RST properties. Due to the high computational complexity of the structure prediction itself, the inference of further RST-tree properties, such as nuclearity and relations, often critical for downstream tasks, are not feasible with the unconstrained CKY approach.

We are now presenting a set of novel methods to overcome these two major restrictions, inspired by the recent success of beam-search methods for multiple NLP parsing tasks [34, 41, 104, 158].

3.4.1 Stochastic Beam-Search

In essence, the general CKY dynamic programming algorithm creates all possible binary trees covering the \( n \) EDUs by internally filling an \( (n \times n) \) matrix, where each cell \((i, j)\) contains the information on all sub-trees covering the text spans from \( EDU_i \) to \( EDU_j \). Our heuristic beam-search solution limits the computational complexity of this process by reducing the number of
sub-trees stored in each cell to a constant beam size $B$. This naturally raises
the question of how to select the $B$ sub-trees to preserve in each cell. We
follow the intuitive assumption that sub-trees for which the sentiment diverges
most from the overall document sentiment (the only available supervision
for this task) can be safely discarded. Out of the set of possible sub-trees $T$
for a given cell, only the subset $T'$ with $|T'| = B$ is kept, containing the $B$
sub-trees with the closest sentiment polarity $p_{t_i}$ to the gold-label sentiment
$gl$ of the document. Formally:

$$T' = \arg\min_{t_i \in T, |T'|=B} |p_{t_i} - gl|$$  \hspace{1cm} (3.9)

One limitation of this heuristic rule is that it strictly prefers sub-trees
with sentiment closer to the overall document sentiment, independent of their
distance from the root node. This can be problematic when applied in early
stages of the tree-generation process, where only a few EDUs are combined.
For instance, a mostly positive document might still contain certain negative
sub-trees at its lowest levels, which also need to be aggregated appropriately.

Ideally, we would like to support a high degree of exploration on low levels
of the tree, only loosely forcing the sentiment of sub-trees in the beam to
align with the overall document gold-label sentiment; while on higher levels
of the tree, the requirement of closely reflecting the document’s gold-standard
sentiment should be strictly enforced (i.e., exploiting the distant supervision).

We implement this strategy through a stochastic beam-search approach,
which relies on a softmax selection using the Boltzmann–Gibbs distribution

[129]. The temperature coefficient $\tau$ thereby modulates the exploration-
exploitation trade-off (similar to previous work in RL), by influencing the
divergence of the softmax outputs. We then sample from the resulting,
categorical probability distribution $P = (\text{Prob}(t_1), ..., \text{Prob}(t_n))$, computed
for every local sub-tree $t_i \in T$ to obtain a subset $T'$ of size $B$ (as shown in
Equation [3.10]).
Figure 3.5: Standard beam-search approach (left) picking the top \( B = 2 \) tree candidates with the smallest distance \( |p_{t_i} - gl| \) in every CKY cell. The stochastic beam-search approach (right) calculates the Boltzmann–Gibbs distribution with the tree-coverage dependent temperature \( \tau \), modulating the sub-tree sampling process of the tree candidates. (For readability, we only show a maximum of 4 sub-trees per CKY cell)

\[
\text{Prob}(t_i) = \frac{e^{-|p_{t_i} - gl|/\tau}}{\sum_{t_j \in T} e^{-|p_{t_j} - gl|/\tau}} \quad (3.10)
\]

\[
\tau = f(n, c) = (n - c) + 1 \quad (3.11)
\]

In this work, the parameter \( \tau \) is defined as a linear function \( f(n, c) \) parameterized by the number of EDUs \( c \) covered under the sub-tree \( t_i \) as well as the total number of EDUs \( n \) (see Equation 3.11). This way, \( \tau \) influences Equation 3.10 such that for larger values of \( c \) (at the top of the tree), \( \tau \) gets close to 1 and the sampling is likely to select sub-trees with low distance \( |p_{t_i} - gl| \). For sub-trees with a small coverage \( c \) (at the bottom of the tree), \( \tau \) becomes \( >> 1 \) and \( \text{Prob}(t_i) \) resembles the uniform distribution, allowing
for a high degree of exploration. For illustration, Figure 3.5 highlights the
differences between the standard and the stochastic beam-search approach.

3.4.2 Analysis of Spacial Complexity
The beam-search extension described in Section 3.4.1 significantly reduces
the spatial complexity, independent of whether a stochastic component is
used or not. In this section, we now quantify the theoretical upper bounds
for the space consumption of the unrestricted CKY approach (Equation 3.12)
against the upper bounds for the heuristically constrained CKY method
(Equation 3.13).

\[
\sum_{i=1}^{n-1} \frac{4(n-i)}{i} \binom{2i-2}{i-1} 
\]

(3.12)

\[4n^2B + 4(n-1)B^2\]

(3.13)

In both equations, \( n \) represents the number of leaf nodes (EDUs) in the
discourse tree. In Equation 3.12 the number of generated trees at every
level of the tree is bound by the Catalan number, while in Equation 3.13 the
bound has a quadratic dependency on the input size and the beam size. For
the equations shown, we assume that on every level of the tree, each of the
possible sub-trees is represented by 2 pointers to the child nodes as well as a
sentiment and attention value for the sub-tree itself. Table 3.1 compares the
space capacities required with increasing document length, indicating that
with a proper beam size, our heuristic strategy can deal with tree structures
of very long documents.

3.4.3 Integration of Nuclearity
With the scalable solution shown to significantly reduce the spatial complexity
compared to the unconstrained CKY approach, we can now take additional
properties, like the nuclearity attribute, into account. The advantage of
generating nuclearity-attributed discourse trees thereby becomes obvious
when revisiting the definition in RST [107], where the nuclearity-attribute
Table 3.1: Upper bound of spatial complexity for different beam sizes and unconstrained CKY (∞), assuming 1Byte per unit in memory.

<table>
<thead>
<tr>
<th>Beam</th>
<th>20 EDUs</th>
<th>30 EDUs</th>
<th>100 EDUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.6KB</td>
<td>3.7KB</td>
<td>40KB</td>
</tr>
<tr>
<td>10</td>
<td>24KB</td>
<td>48KB</td>
<td>440KB</td>
</tr>
<tr>
<td>100</td>
<td>920KB</td>
<td>1.5MB</td>
<td>7.9MB</td>
</tr>
<tr>
<td>∞</td>
<td>3.6GB</td>
<td>1.9PB</td>
<td>400SB</td>
</tr>
</tbody>
</table>

KB = 10^3, MB = 10^6, GB = 10^9, PB = 10^{15}, SB = 10^{54}

Encodes a notion of “importance” in the local context, with Nucleus-Satellite (N-S) and Satellite-Nucleus (S-N) attributions defining the directionality between two nodes, while the Nucleus-Nucleus (N-N) attribution implies equal importance. Expressing this notion of importance, it is not surprising that nuclearity-attribute is frequently critical in informing many downstream tasks like summarization and text categorization.

We integrate the nuclearity attribute into the tree-generation process by assigning each sub-tree one of the three nuclearity classes N-S, S-N or N-N, following the assumption that the attention values \( a_{c_l}, a_{c_r} \) capture the nodes’ relative importance in the tree. Starting from the leaf node attention, extracted from MILNet, we propagate the attention values through the tree structure according to Equations 3.4 and 3.7. For a sub-tree where the attention value \( a_{c_l} \) is greater than the attention \( a_{c_r} \), we will assign the N-S label, while S-N is assigned if the opposite is true. In this way, only two of the three possible nuclearity classes can be represented (namely N-S and S-N), since the real-valued attention scores are distinct. To further account for the third class of N-N, we include an additional sub-tree at every merge in the CKY procedure, which averages not only the two attention values \( a_{c_l}, a_{c_r} \) (as shown in Equations 3.4 and 3.7) but also the child polarity scores \( p_{c_l}, p_{c_r} \). This reflects the definition of the N-N nuclearity class according to RST, where uniform importance for all child nodes is assumed in the multi-nucleus case. This additional complexity introduced in each cell by the N-N class is only manageable due to the use of our heuristic approach, allowing the generation of more complete discourse trees.
### Table 3.2: Size, length and vocabulary diversity of the datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Documents</th>
<th>#EDU/Doc</th>
<th>Vocab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp'13[154]</td>
<td>335,018</td>
<td>19.1</td>
<td>183,614</td>
</tr>
<tr>
<td>RST-DT[18]</td>
<td>385</td>
<td>56.0</td>
<td>15,503</td>
</tr>
<tr>
<td>Instr-DT[149]</td>
<td>176</td>
<td>32.6</td>
<td>3,453</td>
</tr>
</tbody>
</table>

#### 3.5 Evaluation

We now describe the evaluation of our new MEGA-DT discourse dataset. We start with the datasets and discourse parsers used to train and test our approach. Next, we take a look at the vocabulary overlap between train and test data and describe the evaluation metrics. We finish with preliminary experiments and our final results.

#### 3.5.1 Datasets

We use three datasets to train and evaluate our approach, resulting in two new, “silver-standard” discourse treebanks. Table 3.2 summarizes the most important dataset dimensions of the train and evaluation corpora. Besides RST-DT and the Instructional Treebank (short: Instr-DT) described in Section 2.2, we use the Yelp’13 review dataset collected for the Yelp Dataset Challenge in 2013 [154] to train our models. Every data point in the Yelp’13 corpus thereby consists of a review along with a star rating on a 5-point scale. We use the discourse segmented version of the corpus by Angelidis and Lapata [4] to train our system on the auxiliary sentiment prediction task. With the three training/evaluation datasets mentioned above, we generate:

**MEGA-DT$_{base}$**, using our unconstrained CKY approach described in Section 3.3 and Huber and Carenini [59]. We use the pre-segmented version of the Yelp’13 customer review dataset by Angelidis and Lapata [4], separated into EDUs by applying the publicly available discourse segmenter proposed in Feng and Hirst [39]. MEGA-DT$_{base}$ contains short documents with $\leq 20$ EDUs, considering two nuclearity classes (namely N-S and S-N).
Table 3.3: Vocabulary overlap between datasets

<table>
<thead>
<tr>
<th>Dataset Pair</th>
<th>Jaccard Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp’13 ↔ RST-DT</td>
<td>6.28%</td>
</tr>
<tr>
<td>Yelp’13 ↔ Instr-DT</td>
<td>1.73%</td>
</tr>
<tr>
<td>RST-DT ↔ Instr-DT</td>
<td>11.65%</td>
</tr>
</tbody>
</table>

MEGA-DT, our novel treebank described in Section 3.4 and Huber and Carenini [61], is also generated from the original Yelp’13 corpus, akin to MEGA-DT\textsubscript{base}. However, due to the additionally proposed, scalable extensions, MEGA-DT is much larger and more comprehensive, containing \( \approx 250k \) documents and integrating all three nuclearity classes\(^7\).

### 3.5.2 Vocabulary Overlap

We measure the vocabulary overlap of the training and evaluation datasets using the Jaccard similarity index. We show the absolute vocabulary sizes of the datasets in Table 3.2 and visualize the overlap in Table 3.3. The vocabulary overlap between the Yelp’13 corpus (containing reviews), the RST-DT dataset (on news articles) and the Instr-DT corpus (containing home-repair instructions) is predictably low, given the different domains of the datasets. While this would likely be a problem for models solely basing their prediction on raw input words, we use pre-trained word embeddings and a combination of syntactical and lexical features in the discourse parsing component.

### 3.5.3 Discourse Parsers

In our experiments, we apply five simple baselines and six competitive discourse parsers, often used in previous work for comparison studies.

\(^7\)Please note that the MEGA-DT dataset is smaller than the Yelp’13 corpus due to the removal of short reviews with less than 3 EDUs.
**Right/Left Branching Baselines:** predict a binary, fully right- or left-branching tree for every document in the dataset.

**Hierarchical Right/Left Branching Baselines:** predict a binary, fully right- or left-branching tree on sentence level and combine the sentence-level trees in right- or left-branching manner for every document in the dataset.

**Majority Class:** As a baseline for the nuclearity prediction task, we compute the majority class on the training corpora. It is important to note that while the baselines for structure do not require available training data, the majority class measure depends on access to an annotated corpus in the target domain.

**DPLP:** an SVM-based shift-reduce parser build on linear projections of lexical features transformed into discourse-driven latent representations [68].

**gCRF:** proposed by Feng and Hirst [39] implements a greedy bottom-up model using two linear-chain CRFs based on local classifiers. The approach stands out compared to commonly used methodologies through its fast inference time complexity, being linear in regard to the number of sentences.

**CODRA:** a CKY-based chart parser combined with Dynamic Conditional Random Fields, separating the computation on sentence- and document-level [72].

**Two-stage Parser:** by Wang et al. [160] employs two separate SVM classifiers for structure/nuclearity and relations. We rely on this parser in our experiments due to its performance advantage compared to other discourse parsers and its separate computation of the structure/nuclearity and the discourse relation. We use the publicly available code provided by Wang et al. [160] and remove the relation classification module.
Yu et al. [180]: propose a transition-based neural model using implicit syntax features from a fully trained syntax parser with an additional dynamic oracle to predict the discourse of complete documents.

Li et al. [92]: an attention-based neural model targeting the task of discourse parsing by adding tensor-based transformations to better model feature interactions.

3.5.4 Metrics
Consistent with previous work, e.g., Joty et al. [72], Wang et al. [160] and following the recent analysis by Morey et al. [115], our key metric is the average micro precision on span level, computed as the global overlap of the discourse structure prediction and the gold structure. We traverse both discourse trees $\text{tree}_{\text{pred}}$ and $\text{tree}_{\text{gold}}$ of each document $i$ in post-order and compute:

$$\text{precision} = \frac{\sum_i |\text{tree}_{\text{pred}} \cap \text{tree}_{\text{gold}}|}{\sum_i |\text{tree}_{\text{gold}}|}$$

(3.14)

Notice that the choice of precision over recall and F-score has no impact on the results when using manual segmentation, as shown in previous work [72, 160].

3.5.5 Preliminary Evaluation
Prior to our final evaluation, we run a set of preliminary experiments to evaluate different attention aggregation functions ($\text{avg, max, sum}$) as discussed in Section 3.3.4. We find that tested on the inter-domain discourse parsing task, the $\text{avg}$ attention-aggregation function achieves the most consistent performance. In a set of preliminary evaluations on the heuristic and stochastic extensions described in Section 3.4, we find that: (i) A beam size of 10 delivers the best trade-off between computational complexity and performance (out of $\{1, 5, 10, 50, 100\}$), when tested according to the distance between gold-label sentiment and model prediction. (ii) A sentence-first aggregation strategy, using sentence boundary predictions from the NLTK
<table>
<thead>
<tr>
<th>Approach</th>
<th>RST-DT Par.</th>
<th>RST-DT R-Par.</th>
<th>Instr-DT Par.</th>
<th>Instr-DT R-Par.</th>
<th>RST-DT Par.</th>
<th>RST-DT R-Par.</th>
<th>Instr-DT Par.</th>
<th>Instr-DT R-Par.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Branch</td>
<td>9.3</td>
<td>54.6</td>
<td>25.5</td>
<td>62.7</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Left Branch</td>
<td>7.5</td>
<td>53.7</td>
<td>4.3</td>
<td>52.2</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Hier. Right Branch</td>
<td>48.7</td>
<td>74.4</td>
<td>50.7</td>
<td>75.3</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Hier. Left Branch</td>
<td>41.2</td>
<td>70.6</td>
<td>27.5</td>
<td>63.8</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Majority Class</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>N</td>
<td>61.3</td>
<td>N</td>
<td>52.3</td>
</tr>
</tbody>
</table>

**Intra-Domain Evaluation**

<table>
<thead>
<tr>
<th>Approach</th>
<th>RST-DT Par.</th>
<th>RST-DT R-Par.</th>
<th>Instr-DT Par.</th>
<th>Instr-DT R-Par.</th>
<th>RST-DT Par.</th>
<th>RST-DT R-Par.</th>
<th>Instr-DT Par.</th>
<th>Instr-DT R-Par.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPLP*</td>
<td>64.1</td>
<td>82.0</td>
<td>–</td>
<td>–</td>
<td>54.2</td>
<td>68.2</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>gCRF*</td>
<td>68.6</td>
<td>84.3</td>
<td>–</td>
<td>–</td>
<td>55.9</td>
<td>69.4</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>CODRA*</td>
<td>65.1</td>
<td>82.6</td>
<td>–</td>
<td>82.9</td>
<td>55.5</td>
<td>68.3</td>
<td>–</td>
<td>64.1</td>
</tr>
<tr>
<td>Li et al.*</td>
<td>64.5</td>
<td>82.2</td>
<td>–</td>
<td>–</td>
<td>54.0</td>
<td>66.5</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>TS</td>
<td>71.0</td>
<td>86.0</td>
<td>58.9</td>
<td>79.4</td>
<td>58.0</td>
<td>72.4</td>
<td>40.0</td>
<td>62.4</td>
</tr>
<tr>
<td>Yu et al.</td>
<td>–</td>
<td>85.5</td>
<td>–</td>
<td>–</td>
<td>73.1</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

**Inter-Domain Evaluation**

<table>
<thead>
<tr>
<th>Approach</th>
<th>RST-DT Par.</th>
<th>RST-DT R-Par.</th>
<th>Instr-DT Par.</th>
<th>Instr-DT R-Par.</th>
<th>RST-DT Par.</th>
<th>RST-DT R-Par.</th>
<th>Instr-DT Par.</th>
<th>Instr-DT R-Par.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS&lt;sub&gt;RST-DT&lt;/sub&gt;</td>
<td>×</td>
<td>×</td>
<td>46.0</td>
<td>73.6</td>
<td>×</td>
<td>×</td>
<td>27.2</td>
<td>49.8</td>
</tr>
<tr>
<td>TS&lt;sub&gt;Instr-DT&lt;/sub&gt;</td>
<td>46.0</td>
<td>74.3</td>
<td>×</td>
<td>×</td>
<td>22.2</td>
<td>44.7</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS&lt;sub&gt;Mega-DT-base&lt;/sub&gt;</td>
<td>53.0</td>
<td>76.4</td>
<td>46.6</td>
<td>74.1</td>
<td>15.5</td>
<td>35.7</td>
<td>7.3</td>
<td>33.4</td>
</tr>
<tr>
<td>TS&lt;sub&gt;Mega-DT&lt;/sub&gt;</td>
<td>†55.8</td>
<td>†77.8</td>
<td>†50.2</td>
<td>†75.2</td>
<td>15.9</td>
<td>44.9</td>
<td>20.3</td>
<td>†54.9</td>
</tr>
<tr>
<td>Human [115]</td>
<td>78.7</td>
<td>88.3</td>
<td>–</td>
<td>–</td>
<td>66.8</td>
<td>77.3</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

**Table 3.4:** Results of the micro-averaged precision measure using the original Parseval method (Par.) and RST Parseval (R-Par.). Inter-domain subscripts identify the training set. Inter-domain results averaged over 10 independent runs. Models with stochastic components are averaged over 3 distinct generation processes. Best performance per sub-table is **bold.** *Results taken from Morey et al. [115], †statistically significant with p-value ≤ .05 to the best inter-domain baseline (Bonferroni adjusted), – non-published values, × not feasible combinations, TS=Two-Stage, Branch=Branching.

toolkit[8] generally improves performance, consistent with previous findings showing that sentence boundaries are key signals for tree aggregations [72].

---

[8]: www.nltk.org/api/nltk.tokenize.html
3.5.6 Final Evaluation

Main Results
To verify the ability of our newly generated discourse treebanks (MEGA-DT$_{base}$ and MEGA-DT) to support domain-independent discourse parsing, we evaluate their performance on the inter-domain discourse parsing task. Therefore, we train the Two-Stage discourse parser [160] on one domain (e.g., Yelp user reviews in MEGA-DT) and evaluate it on documents in a different domain (e.g., news articles in RST-DT). We compare the obtained performances against the classic and arguably easier intra-domain measure (training and testing on documents within the same domain).

The results of the final evaluation are summarized and aggregated in three sets of experiments in Table 3.4. In the first set (top of Table 3.4), we show the micro-averaged original Parseval performance (Par.) [115] as well as the RST-Parseval measures (R-Par.) of standard linguistic baselines for the structure- and nuclearity-prediction task. Regarding the structure prediction (left), we compare the performance when applying a strictly right- or left-branching tree to the data, as well as hierarchical versions of those (right-/left-branching trees on sentence-level combined by right-/left-branching trees on document-level). The results indicate that the hierarchical right-branching tree resembles the original tree structure the closest on both metrics and either evaluation treebank$^9$. As a baseline for the nuclearity prediction task, we compute the majority class on the training corpora. It is important to note that while the linguistic baselines for structure do not require available training data, the majority class measure depends on access to an annotated corpus in the target domain.

The second set shows the intra-domain results of top-performing discourse parsers, frequently evaluated in the past. While all parsers except CODRA [72] have been only evaluated on RST-DT, we additionally train and evaluate the Two-Stage parser on the Instr-DT corpus. When comparing the intra-

$^9$Note that the performance of the Hierarchical Right-Branching baseline is higher than reported in Huber and Carenini [59], due to additional clean-up step during the data preprocessing.
domain discourse parsing performance, the Two-Stage parser reaches the consistently best performance on RST-DT structure prediction, while Yu et al. [180] achieves the best results on the RST-DT nuclearity prediction. CODRA reaches the best performance on the Instr-DT corpus.

The main contribution of this work is placed in the third set of results, where the Two-Stage discourse parser is trained and tested on different, non-overlapping domains (i.e., inter-domain). This task is arguably more useful and significantly more difficult than the task evaluated in the second set, which is reflected in the performance decrease for structure and nuclearity in the first two rows of the sub-table, confirming that the transfer of discourse structures and nuclearity between domains is a challenging task. The results presented in the third row of the sub-table show the performance of the Two-Stage parser when trained on MEGA-DT\textsubscript{base}, containing short documents with limited nuclearity annotations. The approach achieves consistently better performance compared to the first two rows on the inter-domain structure prediction task (For both, original Parseval and RST-Paraseval). However, only considering two out of three nuclearity classes (N-S and S-N), the system performs rather poorly on the nuclearity classification task. The bottom row of the third sub-table displays the performance of the Two-Stage discourse parser when trained on our new MEGA-DT corpus. Training
Figure 3.6: Performance trend over increasingly large subsets tested on RST-DT (left) and Instr-DT (right). Each sample is generated as the average performance over 10 random subsets, drawn from 3 independently created treebanks.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>N-N</th>
<th>N-S</th>
<th>S-N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-N</td>
<td>243</td>
<td>11</td>
<td>40</td>
</tr>
<tr>
<td>N-S</td>
<td>570</td>
<td>28</td>
<td>96</td>
</tr>
<tr>
<td>S-N</td>
<td>244</td>
<td>5</td>
<td>47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted</th>
<th>N-N</th>
<th>N-S</th>
<th>S-N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-N</td>
<td>85</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>N-S</td>
<td>77</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>S-N</td>
<td>23</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3.6: Confusion Matrices for our MEGA-DT-trained model evaluated on RST-DT (left) and Instr-DT (right).

on MEGA-DT delivers statistically significant improvements over the best inter-domain baseline in all structure prediction tasks. Furthermore, our new system also achieves statistically significant gains on the Instr-DT nuclearity prediction, when evaluated according to the RST-Parseval metric. The nuclearity measure on RST-DT using RST-Parseval is statistically equivalent to the best baseline system. Overall, our MEGA-DT corpus appears to outperform previously published treebanks for inter-domain discourse parsing on every sub-task on at least one competitive metric.

Ablations and Additional Experiments

In order to gain deeper insights into the effectiveness of our proposed treebank generation approach, we run a set of four additional evaluations. First, we
evaluate the individual components of our system by showing an ablation study in Table 3.5 starting with the performance of the discourse parser trained with MEGA-DT-beam, generated with the standard beam-search approach, without integrating the stochastic component and nuclearity. Adding each feature separately (+stoch, +nuc), we observe improvements on at least one of the sub-tasks; however, the combination of the two components produces the best-performing MEGA-DT corpus. Second, we show the performance trend over increasingly large subsets of MEGA-DT in Figure 3.6 tested on RST-DT (left) and Instr-DT (right). The two trends highlight consistent improvements with increasingly large dataset sizes, suggesting further possible gains with even larger treebanks. Third, we further analyze the nuclearity classification performance in Table 3.6, which presents four confusion matrices for the discourse parsing output of our MEGA-DT treebank, evaluated according to the original Parseval metric on RST-DT and Instr-DT. The matrices show a potential explanation for the performance gap between the original Parseval and the RST-Parseval metrics, identifying the over-prediction of the N-N class, especially for gold-label N-S nuclearities. Further, we frequently misclassify the gold-label N-S nuclearity class as S-N. Lastly, we present an additional qualitative analysis in Figure 3.7 and Appendix A.1 to investigate the strength and potential weaknesses of trees in MEGA-DT. In general, the qualitative analysis shows that trees in MEGA-DT are non-trivial, reasonably balanced, strongly linked to the EDU-level sentiment and mostly well-aligned with meaningful discourse structures.

3.5.7 Contributions

In this chapter, we address a key limitation to further progress in discourse parsing: the lack of annotated datasets. We show promising results to overcome this limitation by creating a large-scale discourse dataset for documents of arbitrary length, solely using distantly supervised signals from document-level sentiment information. To deal with the increasing spatial complexity, we apply and compare heuristic beam-search strategies, including a stochas-
Figure 3.7: Teaser for a tree analyzed in Appendix A.1 containing 28 EDUs and neutral document-level sentiment.

Heuristic variant inspired by RL techniques. Experiments indicate that a parser trained on our new corpus outperforms parsers trained on human-annotated datasets on the challenging and very useful task of inter-domain discourse structure prediction. Furthermore, our results suggest that the heuristic approach (i) enhances the structure prediction task through more diversity in the early-stage tree selection, (ii) allows us to reasonably predict nuclearity and (iii) helps to significantly reduce the complexity of the unrestricted CKY approach to scale for arbitrarily long documents.

In conclusion, our new approach allows the NLP community to augment any existing sentiment-annotated dataset with discourse trees, enabling the automated generation of large-scale domain/genre-specific discourse treebanks. As a case study for the effectiveness of the approach, we annotate and publish our MEGA-DT corpus as a high-quality RST-style discourse treebank. Our results shown in this chapter suggest that discourse parsers trained on our MEGA-DT corpus (or further domain-specific treebanks generated according to our approach) should be preferred over supervised discourse parsers when aiming to derive discourse trees in domains where no gold-labelled data is available.

Looking back at the research questions posed in Chapter 1, this line of
work answers research question (RQ1), describing a promising approach to generate large-scale discourse structures, which outperform human-annotated treebanks on the inter-domain discourse parsing task. As such, this chapter partially confirms our hypothesis, presenting evidence that to successfully model the task of sentiment analysis, some notion of discourse is internally encoded. Since the publication of the code and treebank in November 2020, the corpus has been downloaded over 30 times by well-known institutions and companies, including, but not limited to: UC Berkeley, New York University, University of Pittsburgh, Nanyang Technological University, Michigan State University, University of Maryland, Disney Tech, Microsoft Research.

Despite the promising performance of our newly proposed MEGA-DT corpus on the inter-domain discourse parsing task, three major open questions remain: (i) We show the effectiveness of the MEGA-DT treebank to predict inter-domain discourse structures, aligned with human-annotated trees. While this is likely the most crucial evaluation for a new discourse dataset, to better understand its usefulness in real-world scenarios, additional experiments regarding performance improvements on NLP downstream tasks are highly valuable. Hence, we explore the potential of MEGA-DT to aid the prediction of sentiment labels for long documents in Chapter 9. (ii) The auxiliary task of sentiment analysis is well aligned with low- and mid-level discourse structures. However, looking at the high-level, topical structure of texts, sentiment likely plays only a minor role in the decision on how to combine sub-trees. To this end, we explore the auxiliary task of topic segmentation in Chapter 4, specifically aiming to predict high-level structures in constituency discourse trees. (iii) Even though we are able to infer nuclearity attributes as part of the CKY decoding process, our experiments executed in Section 3.5.6 show clear weaknesses of the nuclearity attribution using the auxiliary task of sentiment analysis. For example, the N-N class is severely over-predicted. To improve the inter-domain discourse parsing performance along this line, we explore the auxiliary task of summarization in Chapter 5, which has previously shown to be well aligned with the nuclearity attribute of constituency discourse trees.
Chapter 4

Discourse Inference from Topic Segmentation Annotations

This chapter was published and presented at AAAI 2022 [65]. I was the lead investigator, working with fellow Ph.D. student Linzi Xing. In this project, I was partially responsible for the concept formation, formulation of research questions, data collection, and implementation of discourse-related components and evaluations. I further guided the paper composition and wrote substantial parts of the publication. Giuseppe Carenini supervised and supported the project.

4.1 Motivation

Expanding upon the work described in Chapter 3, focusing on low- and mid-level discourse structures and their alignment with the auxiliary task of sentiment analysis, we now take a detailed look at the important high-level discourse structures. Given the definition of complete discourse trees in the Rhetorical Structure Theory (RST), structures inevitably become deeper as documents grow longer\(^1\). Furthermore, the defining factors for

\(^1\)For more information on the RST discourse theory, please refer to Section 2.1
the tree aggregation on higher levels differ considerably from the ones on lower levels (e.g., aggregating multiple paragraphs vs. combining EDUs) [70]. For example, suitable features for EDU level tree aggregations (i.e., low levels) are mostly influenced by local syntactic and semantic signals, while the tree aggregation on paragraph-level (i.e., high levels) is likely to follow more global features, such as major topic shifts planned by the author for possibly complex communicative goals [148].

Researchers working on RST-style discourse parsing take these considerations into account by either (i) proposing hard constraints to construct discourse trees on distinct textual levels by using varying feature sets [68, 72] or (ii) as done in recent work by Wang et al. [160] and Guz and Carenini [49], by proposing soft constraints, encoding sentence- and paragraph-breaks as input features.

Ideally, an RST-style discourse parser should achieve good performance on all levels shown in Figure 4.1. However, as argued in Kobayashi et al. [83], the performance on high-level constituency tree structures is especially important (green arrow in Figure 4.1) when converting these constituency trees into dependency structures, as typically done for key downstream
applications [60, 69, 109, 110, 142, 173].

Unfortunately, training samples for these critical high-level discourse structures are extremely limited (see red arrow in Figure 4.1). Not only does the largest available human-annotated treebank in English solely contains 385 documents, but for each document, vastly more training samples are available for structures within sentences than for structures connecting sentences and paragraphs, since the number of nodes in a binary tree decreases exponentially from the leaves towards the root.

To tackle the data sparsity issue for discourse parsing, previous work has proposed to leverage distant supervision from tasks with naturally annotated and abundantly available training data, such as sentiment analysis [61] and summarization [173]. While we believe that both these auxiliary tasks capture some structural information, they are plausibly even more aligned with other aspects of discourse, such as nuclearity for summarization and discourse relations (e.g., evidence, concession) for sentiment. In contrast, this chapter focuses exclusively on high-level tree structure generation, by exploiting signals from the auxiliary task of topic segmentation. Training on naturally occurring and abundantly available topic segmentation annotations (sections/paragraphs), we believe that valuable information on major topic shifts, an effective signal indicative of high-level discourse tree structures, can be learned [148].

More specifically, we train the top-performing neural topic segmentation model proposed in Xing et al. [174] and use the trained model to generate discourse structures, which we evaluate on two popular discourse parsing treebanks. Based on the sequential output of the topic segmentation model, we explore the generation of discourse trees using (i) a greedy top-down algorithm and (ii) an optimal bottom-up CKY dynamic programming approach [74], predicting RST-style tree structures on/above sentence-level.

To better understand and properly compare the performance of our discourse tree generation algorithm with previously published models, as well as a set of baselines, we evaluate all approaches on three partially overlapping discourse tree subsets from sentence-to-paragraph (S-P), paragraph-to-document (P-D) and sentence-to-document (S-D). In our evaluation, we find
that distant supervision from topic segmentation achieves promising results on the high-level tree structure generation task, consistently outperforming previous methods with distant supervision on sentence-to-document-level and in some cases reaching superior performance compared to supervised models.

4.2 Related Work

As motivated in Section 2.1, the Rhetorical Structure Theory (RST) \cite{107} is one of the main guiding theories for discourse parsing, postulating complete constituency discourse trees consisting of a structure, nuclearity and relation attribute. In this chapter, we follow the general RST paradigm, however, exclusively generate above-sentence, “plain” constituency discourse trees (without nuclearity and relation attributes) through distant supervision from topic segmentation. We thereby draw inspiration from the findings in Jiang et al. \cite{70}, showing that a pre-trained topic segmenter can benefit supervised discourse parsers, especially on high levels. However, instead of augmenting a supervised discourse parser with information obtained from a topic segmentation model, we explore the potential of directly inferring high-level discourse structures from the output of topic segmentation, bypassing the data sparsity issue of supervised models through the use of distant supervision\textsuperscript{2}.

The auxiliary task in this chapter, topic segmentation, aims to uncover the latent topical structure of a document by splitting it into a sequence of thematically coherent segments using large-scale corpora sampled from Wikipedia \cite{5, 84}. Specifically, we employ the top-performing neural topic segmenter proposed in Xing et al. \cite{174} as our base model for the discourse tree inference\textsuperscript{3}.

\textsuperscript{2}For further related work on discourse parsing, we refer readers to Section 2.3.

\textsuperscript{3}More details on the auxiliary task of topic segmentation can be found in Section 2.4.2.
4.3 Methodology

In this section, we describe our new approach to generate high-level discourse structures for arbitrary documents using distant supervision from topic segmentation. An overview of our methodology is presented in Figure 4.2. At the top, we visualize a naturally organized document, where words are aggregated into clauses (or EDUs), which are in turn concatenated into sentences. Sentences are then further combined into paragraphs\(^4\), forming the complete document. Here, we aggregate sentences as the atomic units to obtain high-level discourse trees (bottom of Figure 4.2). We do not consider the intra-sentence task (EDU-to-sentence), as individually tackled by Lin et al. [94] with great success.

\[^4\text{We overload the term “paragraph”, also referring to what in other places is called “sections”.}\]
4.3.1 The Topic Segmentation Model

The topic segmentation task is commonly interpreted as a sequence labelling problem. Formally, given a document \( d \) in the form of a sequence of sentences \( \{s_1, s_2, ..., s_k\} \), a topic segmentation model assigns a probability score to each sentence \( s_i \) for \( i \in [1, ..., k-1] \), generating a sequence of probabilities \( P = \{p(s_1), p(s_2), ..., p(s_{k-1})\} \) with \( p(s_i) \in [0, 1] \) indicating how likely sentence \( s_i \) is the end of a segment\(^5\).

Based on the set of probability scores \( P \) and an additional threshold \( \tau \), determined on a held-out development set, topic segmentation models make binary predictions, labelling sentences with a probability larger than \( \tau \) as the “end of segment”. In contrast to this common approach for topic segmentation, relying on the threshold parameter \( \tau \) to convert the probabilities into discrete sentence boundary labels, we directly utilize the real-valued probability scores as our supervision signal to infer RST-style discourse trees here.

The high-level architecture of the chosen neural topic segmentation model [174] is shown in Figure 4.3. Architecturally based on the proposal in Koshorek et al. [84], the approach extends a hierarchical BiLSTM network by adding a BERT encoder, an auxiliary coherence prediction module and restricted self-attention, shown to improve the model’s effectiveness for

\(^{5}\)The last sentence \( s_k \) does not need to be scored, since it is by definition the end of the last segment.
Table 4.1: Average topic segmentation performance ± standard deviation over 5 runs, using the $P_k$ error score [11] (lower values indicate better performance). Best performance per column in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>Wiki</th>
<th>RST-DT</th>
<th>GUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextTiling [53]</td>
<td>62.5 ± 0.00</td>
<td>44.3 ± 0.00</td>
<td>51.6 ± 0.00</td>
</tr>
<tr>
<td>BayesSeg [36]</td>
<td>38.6 ± 0.00</td>
<td>37.5 ± 0.00</td>
<td>49.8 ± 0.00</td>
</tr>
<tr>
<td>GraphSeg [46]</td>
<td>43.2 ± 0.00</td>
<td>58.7 ± 0.00</td>
<td>53.9 ± 0.00</td>
</tr>
<tr>
<td>Koshorek et al. [84]</td>
<td>27.9 ± 0.12</td>
<td>26.9 ± 0.49</td>
<td>48.0 ± 1.37</td>
</tr>
<tr>
<td>Xing et al. [174]</td>
<td>25.1 ± 0.07</td>
<td>25.4 ± 0.33</td>
<td>40.8 ± 1.64</td>
</tr>
</tbody>
</table>

context modelling. Since the topic segmentation task is generally applied on sentence level, any discourse tree generated based on its output can by definition only cover structures on and above sentence level (i.e., where leaf nodes represent sentences). The effectiveness of the selected topic segmentation model is demonstrated in Table 4.1, showing the approach by Xing et al. [174] to yield superior performance over its direct competitors on the task of topic segmentation itself.

4.3.2 Tree Generation

To convert the sequential outputs of the topic segmentation model into tree structures, we follow the intuition that the value at position $i$ in the output of the topic segmentation model, namely the likelihood of $s_i$ being the end of a thematically coherent segment, can be interpreted as the topical distance between $s_i$ and $s_{i+1}$. A natural way to exploit the output of the topic segmentation model is to create a binary discourse tree by applying a greedy top-down algorithm, where text spans (i.e., sub-trees) are determined recursively by splitting the two sentences with the largest topical distance. A small-scale example of this approach is shown in Figure 4.4. Here, we first search for the sentence

$$s_{max} = \arg \max_{s_i \in d} p(s_i)$$ (4.1)
with the maximum probability in $P$, making it our segmentation point. We then segment the sequence $P$ into two sub-sequences:

$$P_l = \{p(s_1), p(s_2), ..., p(s_{\text{max}})\}$$

$$P_r = \{p(s_{\text{max}+1}), p(s_{\text{max}+2}), ..., p(s_k)\}$$

Finally, we save and mark $s_{\text{max}}$ as a selected segmentation point by setting $p(s_{\text{max}}) = 0$. We recursively repeat this process for the two sub-sequences $P_l$ and $P_r$ in a divide-and-conquer style, until all sentence probabilities are set to 0. Noticeably, the bottom-up greedy strategy is equivalent to the top-down approach in this case.

Besides the greedy approach described above, a commonly used tree-aggregation technique to convert a real-valued sequence into a binary tree structure is the optimal CKY dynamic programming algorithm, as used in Chapters 3 and 5. However, applying the CKY algorithm to the real-valued topic break probabilities is problematic, since the output of any topic segmentation model only contains a single sequence. The intuitive way to fill the CKY matrix is to start merging any two consecutive units $s_i$ and $s_{i+1}$ with the score $p(s_i)$ of the former unit $s_i$ and assigning the new (merged) span the aggregation score of the latter, here $p(s_{i+1})$. Applying this strategy using the dynamic programming approach, every tree candidate receives the same likelihood to represent the document at hand, due to the commutativity
Figure 4.5: Problematic CKY aggregation without the use of discounting. All trees receive the same final score.

property of the sub-tree aggregation (see Figure 4.5 for an example). In order to address this issue, additional hyper-parameters can be introduced, such as an attribute to quantify the benefit of merging low-probability sentences early on and delaying likely topic breaks. We explore this extended version of the CKY approach using a set of fixed discount factors. In preliminary experiments, we found that this adaption of the CKY approach, despite being theoretically superior, is in practice inferior to the greedy approach, showing unstable performance and resulting in heavily balanced/imbalanced trees for large/small discount factors, respectively. As a result, we leave the task of finding a more effective discounting function to future work and focus on the superior greedy top-down approach (as illustrated in Figure 4.4) from here on.

4.4 Evaluation
# Datasets

We use three datasets to train the topic segmentation models: (i) We randomly sample a subset of Wikipedia articles from the Wikipedia dump\(^6\), strictly following the sampling scheme in Koshorek et al. [84] (from here on called Wiki). Our Wiki corpus consists of 20,000 documents, consistent in size with previously proposed topic segmentation training corpora, such as the Wiki-Section dataset [5], originally used in the top-performing model by Xing et al. [174]. In contrast to the popular Wiki-Section dataset, which extracts a strictly domain-limited subset of Wikipedia, exclusively covering articles on cities and diseases, we lift this topic restriction by uniformly sampling from all of Wikipedia. We split our Wiki dataset into training, validation and test sets using the default 80%, 10%, 10% data split. We further train the topic segmentation model on (ii) the RST-DT dataset [18] and (iii) the GUM [183] corpus, both described in Section 2.2. Since both discourse corpora do not have any explicit human-annotated topic segment boundaries, we use the (arguably more fine-grained) paragraph breaks contained in the textual representation of the data as topic-shift indicators. The key statistics of all three topic segmentation training datasets are presented in Table 4.2. Please note that we do not use any human-annotated discourse structures during the training procedure of the topic segmentation model.

\(^6\)https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Wiki</th>
<th>RST-DT</th>
<th>GUM</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Docs.</td>
<td>20,000</td>
<td>385</td>
<td>150</td>
</tr>
<tr>
<td># of Para./Doc.</td>
<td>31.1</td>
<td>9.99</td>
<td>12.3</td>
</tr>
<tr>
<td># of Sents./Doc.</td>
<td>144.9</td>
<td>22.5</td>
<td>49.3</td>
</tr>
<tr>
<td># of EDUs/Doc.</td>
<td>(\times)</td>
<td>56.6</td>
<td>114.2</td>
</tr>
<tr>
<td># of EDUs/Para.</td>
<td>(\times)</td>
<td>5.67</td>
<td>9.29</td>
</tr>
<tr>
<td># of EDUs/Sent.</td>
<td>(\times)</td>
<td>2.51</td>
<td>2.32</td>
</tr>
</tbody>
</table>

Table 4.2: Statistics of the three training/testing datasets.
4.4.2 Baselines

We compare our distantly supervised model against three sets of baselines: (i) Simple, but oftentimes competitive structural and random baselines, including Right-Branching, Left-Branching and Random Trees. (ii) Completely supervised models, including the traditional Two-Stage model [160] trained within- and cross-domain, as well as the recently proposed neural SpanBERT approach [49]. (iii) Distantly supervised models, exploiting signals from auxiliary tasks with available, large-scale training data. We compare our results against our model presented in Chapter 3, training the Two-Stage parser on the MEGA-DT discourse treebank inferred from sentiment analysis, and our model described in Chapter 5, directly computing discourse structures from an extractive summarization model trained on the CNN-DM dataset.

For topic segmentation, we compare the model proposed in Xing et al. [174] against three widely applied unsupervised baselines: TextTiling [53], BayesSeg [36] and GraphSeg [46], and a competitive supervised model by Koshorek et al. [84], as previously shown in Table 4.1.

4.4.3 Experimental Design

Given that our newly proposed RST-style tree generation strategy can only be applied on and above sentence level, we can not directly compare our model with results provided in the literature, generally evaluating complete discourse trees from EDU-to-document level. In order to fairly compare against baselines and previously proposed models, and to provide a better understanding of how our proposal performs on different textual levels of the document (as depicted in Figures 4.1 and 4.2), we design a set of comparisons extending the regular evaluation technique for complete discourse trees. More specifically, instead of simply computing the micro-average performance for a complete document (i.e., EDU-to-document), we aim to investigate the structural performance on sentence-to-document (S-D), sentence-to-paragraph (S-P) and paragraph-to-document (P-D) level, allowing us to observe trends on different levels of the document. To obtain partial discourse trees for different textual levels from the complete (EDU-
to-document) gold structures, we need to trim the complete tree to (i) only contain nodes covering a complete unit of the lower bound of measure (e.g., sentences in S-D/S-P) and (ii) do not contain any nodes covering multiple units of the upper bound of measure (e.g., paragraphs in S-P). Hence, we propose the following two strategies:

1. Trimming Trees to the Lower Bound: Assuming we want to trim the discourse tree up to the sentence level (however, similarly applicable for paragraphs), we aim to remove all intra-sentence sub-trees that do not cover at least a complete sentence. In other words, assuming we have a sequence of EDUs: \( E = \{e_1, e_2, \ldots e_m\} \), we keep a node \( n \) spanning \( e_i \) to \( e_j \) in the discourse tree iff \( \text{sent}(e_i) \neq \text{sent}(e_j) \) or \( e_i.\text{is.beginning}(\text{sent}_k) \land e_j.\text{is.end}(\text{sent}_k) \), with \( \text{sent}() \) returning the sentence assignment of an EDU. Subsequently, we update the node \( n = (e_i, e_j) \) to \( n = (\text{sent}(e_i), \text{sent}(e_j)) \). While this approach works for complete sentence sub-trees, we follow the additional rules presented in Sporleder and Lascarides [146] for potentially leaky sentence-level trees (about 5% of gold-standard trees in RST-DT [72]).

2. Restricting Trees to the Upper Bound: Regarding the restriction of trees to selected upper bounds (here exemplified on paragraph-level), we remove any node \( n \) covering sentences \( s_i \) to \( s_j \) iff \( \text{para}(s_i) \neq \text{para}(s_j) \) with \( \text{para}() \) returning the paragraph assignment of a sentence.

### 4.4.4 Experiments and Results

We show the discourse structure parsing performance of each model on the sentence-to-paragraph (S-P), paragraph-to-document (P-D) and finally sentence-to-document (S-D) level in Table 4.3. Our results are subdivided into performances on RST-DT (left) and GUM (right). We further show separate sub-tables aligned with the type of supervision used. The top sub-table contains unsupervised baselines generating either random trees

---

7 Additional experiments and results using the original parseval score and EDU-level performances can be found in Appendix [A.2]
or completely right-/left-branching structures. The second set of results (in the center sub-table) contains supervised discourse parsers, and the bottom sub-table shows distantly supervised models, including our results. To evaluate the ability of our proposal to generate domain-independent discourse structures, we compare the Topic Segmenter (TS) trained on in-domain data (TS_{RST-DT/GUM})\(^8\), the out-of-domain Wiki corpus (TS_{Wiki}) as well as a “fine-tuned” approach, first trained on the Wiki corpus, and subsequently fine-tuned on RST-DT or GUM (TS_{Wiki+RST-DT/GUM}). Finally, to further assess the role of the context modelling component (coherence module and restricted self-attention) in regard to the performance of topic segmentation as the distantly supervised task for discourse tree generation, we ablate the context modelling component (red/green parts in Figure 4.3) as “Ablation – TS_{Wiki}” in the last row of Table 4.3.

Not surprisingly, when evaluating discourse structures on the RST-DT dataset (Table 4.3, left), supervised models generally outperform unsupervised baselines and distantly supervised models. Assessing the bottom sub-table in more detail, it becomes clear that while the Two-Stage parser trained on MEGA-DT achieves the best performance on the sentence-to-paragraph-level, our model solely trained on the Wiki dataset performs best on the paragraph-to-document and sentence-to-document-level, showing that (i) obtaining discourse structures from topic segmentation effectively supports high-level discourse parsing and (ii) general out-of-domain training on large-scale data (TS_{Wiki}) performs better than models trained or fine-tuned on in-domain data (TS_{RST-DT} and TS_{Wiki+RST-DT} respectively). We believe that a possible explanation for the surprising under-performance of the in-domain training described above could originate from the limited size of the RST-DT dataset, as well as the mismatch between the “granularity” of segment information in Wikipedia and RST-DT (see Table 4.2), where the average number of sentences per segment is about twice as large in Wikipedia than RST-DT. Plausibly, the larger segments in Wikipedia better support higher-level discourse structures (which cover larger text spans), leading to

\(^8\)Please note that we only use the textual data from the RST-DT/GUM corpora, not using any available discourse annotations.
Model | RST-DT | GUM
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S-P</td>
<td>P-D</td>
</tr>
</tbody>
</table>
| Baselines
| Random* | 77.11 | 63.90 | 60.20 | 67.53 | 60.96 | 57.99 |
| Right-Branching | 73.57 | 65.50 | 59.46 | 64.15 | 72.71 | 59.39 |
| Left-Branching | 72.41 | 64.07 | 58.07 | 62.07 | 54.35 | 51.56 |
| Supervised RST-style Parsers
| Two-StageGUM | 88.82 | 65.63 | 69.58 | 76.70 | **72.94** | **68.38** |
| Two-StageRST-DT | 90.64 | 68.09 | 72.11 | 74.20 | 63.29 | 63.65 |
| SpanBERTRST-DT | **90.75** | **76.03** | **77.19** | – | – | – |
| Distantly Supervised RST-style Parsers
| Xiao et al. [173]CNN/DM | 74.23 | 66.15 | 59.10 | 67.89 | 57.80 | 53.82 |
| Two-StageMEGA-DT | 85.00 | 65.50 | 66.99 | 73.37 | **69.88** | 64.69 |
| TS_{RST-DT} | 84.34 | 62.52 | 65.96 | – | – | – |
| TS_{GUM} | – | – | – | 72.54 | 67.60 | 62.79 |
| TS_{Wiki} | 83.43 | **69.78** | **68.13** | **76.98** | 63.53 | 65.84 |
| TS_{Wiki+RST-DT} | 83.84 | 66.54 | 65.84 | – | – | – |
| TS_{Wiki+GUM} | – | – | – | 74.48 | 67.29 | 64.69 |
| Ablation – TS_{Wiki} | 83.51 | 68.61 | 67.47 | 75.94 | 64.71 | 65.38 |

Table 4.3: Evaluation results using the RST Parseval micro-average precision measure. Subscripts indicate the training dataset. TS = Topic Segmentation Model. * = Average performance over 10 runs. Best performance per sub-table underlined, best performance per column **bold**.

superior performance, with the RST-DT fine-tuning step (at finer granularity) introducing a mismatch, and therefore not delivering the expected benefits.

Our evaluation results on the GUM dataset (Table 4.3 right) show similar trends to the evaluation on RST-DT. The in-domain trained and fine-tuned models in the bottom sub-table do not achieve improved performances compared to the out-of-domain TS_{Wiki} model. We believe an additional factor for this behaviour is the mix of domains within the small GUM training portion, resulting in the fine-tuning step mixing noisy signals from different genres, hence not providing consistent improvements over Wikipedia.
Interestingly, however, there are some observations which differ from RST-DT: (i) Unlike for RST-DT, our distantly supervised model trained on Wiki even outperforms supervised approaches on sentence-to-document-level. (ii) Right-branching trees outperform random trees, which hasn’t been the case on the RST-DT dataset and (iii) the paragraph-to-document (P-D) level differs from previous results, with the right-branching baseline reaching a performance close to the best supervised model (Two-Stage GUM), outperforming the out-of-domain supervised model (Two-Stage RST-DT) and all distantly supervised approaches.

Regarding the ablation studies, our results shown in the last row of Table 4.3 imply that the context modelling component, shown to boost the topic segmentation performance, can also consistently benefit the high-level discourse structure inference on the S-D level.

To further investigate the performance on paragraph-to-document-level for the GUM corpus, we show a comparison by genre for the surprisingly high-performing right-branching baseline, the two supervised models and our methods based on Wiki and GUM in Table 4.4. Right-branching trees
thereby achieving the best performance in 3 out of the 9 genres, including textbooks and fiction. Supervised methods perform best on 5 out of the 9 genres, including highly structured domains such as academic writing and interviews. Our distantly supervised model trained on Wikipedia reaches the best performance on biographies and the topic segmentation model trained on GUM achieves the highest score on how-to guides and news articles. Furthermore, as expected, the supervised parser trained on RST-DT (i.e., the news domain) performs well in the news genre. Overall, while these mixed results appear to align well with our intuition on the prevalent structures in certain genres, further research is required to better understand the relationship between modelling decisions and their impact on different discourse structures across genres.

To complement our quantitative evaluation, we show a set of predicted
trees and their respective gold-label structures for well captured above-sentence discourse (Figure 4.6), randomly selected documents (Figure 4.7) and poorly captured samples (Figure 4.8). Additional qualitative tree examples are shown in Appendix A.3.

Overall, inspecting a large number of tree structures along with their gold labels, we recognize that the generated trees are slightly more balanced than the respective gold-label trees (see for example Figures 4.7 and 4.8), however generally represent non-trivial structures, oftentimes well-aligned with major topic shifts and high-level discourse.

4.5 Contributions

In this chapter, we show that topic segmentation can provide useful signals for high-level discourse constituency tree generation. Comparing multiple aggregation approaches, our proposal using a greedy top-down algorithm performs well when applied on two popular gold-standard discourse treebanks, namely RST-DT and GUM. We provide a detailed evaluation based on textual levels in documents, giving insights into the strength and weaknesses of simple baselines, previously proposed models and our new, distantly supervised approach using topic segmentation, on sentence-, paragraph- and document-level. We compare a variety of topic segmentation training datasets within domain (i.e., on RST-DT and GUM) and outside of the evaluation domain.
(i.e., Wikipedia) in combination with the state-of-the-art topic segmentation model by Xing et al. [174]. In our experiments, we find that large-scale out-of-domain datasets with longer average sections provide better supervision signals for high-level discourse structures than small-scale, short, in-domain paragraphs. Interestingly, we show that even fine-tuning on in-domain data does not help for both, RST-DT and GUM, likely caused by the mismatch in paragraph/section length. Besides the main results, we further show additional insights into our modelling approach through an ablation study, per-genre evaluations and qualitative tree generation samples.

In conclusion, the line of research presented in this chapter effectively alleviates one of the weaknesses shown in our previous work using distant supervision from sentiment signals (Chapter 3): The discourse structure prediction on high levels of long and diverse documents. With topic segmentation signals reasonably showing performance improvements for between-paragraph discourse structures, the inferred trees can be used to complement low- and mid-level sentiment inferred discourse structures, making the approach a valuable extension of our previous work.

Putting this line of research into the context of our research questions presented in Chapter 1, we answer research question (RQ1) by proposing another effective approach to infer discourse structures, which improves our previously obtained results for high-level documents. This chapter further validates our initial hypothesis on the existence of discourse information in auxiliary-task models by showing that the SOTA topic segmenter encodes some notion of high-level discourse.

While topic segmentation has presented itself as a valid auxiliary task for distant supervision of high-level constituency discourse structures, the results in this chapter are limited to purely structural information, leaving out other important RST-style discourse properties, namely the nuclearity and relation annotation. Extending our work on distantly supervised approaches to infer discourse structures, we explicitly tackle the missing nuclearity attribute in Chapter 5 by using the promising auxiliary task of extractive summarization.
Chapter 5

Discourse Inference from Summarization Annotations

This chapter has been published at NAACL 2021 [171]. I have been the second author, with Wen Xiao as the lead investigator. Throughout the project, I contributed to the process of defining project goals and key research questions, supported the implementation, and took stakes in the design of the experiments. I further contributed to the paper itself. Giuseppe Carenini was the supervisory author of the project.

5.1 Motivation

Similar to the previously described synergistic connection of sentiment annotations and topic segmentation labels with the task of discourse parsing, we are now exploring the relationship between (extractive) summarization and discourse, extending the previously described approaches in Chapters 3 and 4 by focusing on discourse nuclearity, either shown to not be sufficiently captured or not taken into account before.

The auxiliary task of extractive summarization is thereby a common and important task within the field of Natural Language Processing (NLP), aiming to produce summaries for multi-sentential documents by selecting
a subset of textual units from a source document. Inspired by previous work in pre-neural times, indicating that discourse information, especially discourse trees according to the Rhetorical Structure Theory (RST), can benefit the summarization task, several neural summarizers have tried to explicitly encode discourse information to support summarization. In particular, injecting discourse has shown to improve the performance of the extractive summarization task and allows for a substantial reduction in the number of model parameters.

As a result, in this chapter, we explore if the synergy between discourse parsing and summarization is bidirectional. In other words, we examine if summarization is a useful auxiliary task to infer discourse structures. Along these lines, Liu et al. performed a preliminary investigation, showing that structural information can be inferred from attention mechanisms while training a neural model on auxiliary tasks. However, they did not perform any comparison against ground-truth discourse trees. Further, recent work showed that discourse trees implicitly induced during training are oftentimes trivial and shallow, not representing valid discourse structures.

Here, we address this limitation of previous work by explicitly exploring the relationship between summarization and discourse parsing through the inference of document-level discourse trees from pre-trained summarization models, comparing the results against ground-truth RST discourse trees. Besides Liu et al., our idea and approach are inspired by recent works on extracting syntactic trees from pre-trained language models or machine translation approaches, as well as previous work on knowledge graph construction from pre-trained language models. Specifically, we generate full RST-style discourse trees from self-attention matrices of a pre-trained transformer-based summarization model. We use three different tree-aggregation approaches (CKY, Eisner and CLE), generating a set of constituency and dependency trees. We empirically evaluate our distantly supervised method on three datasets with human RST-style annotations, covering different text genres. Multiple experiments show that the summarization model learns discourse information implicitly and that

---

1For more details on the auxiliary task of summarization, please refer to Section 2.4.3.
more dependency information is captured, compared to structural (i.e., constituency) signals. Interestingly, additional exploration of the attention matrices of individual heads suggests that, for all models, most of the discourse information is concentrated in a single head, and the best-performing head is consistent across all datasets. We further find that the dependency information learned in the attention matrix covers long-distance discourse dependencies. Overall, the results are consistent across datasets and models, indicating that the discourse information learned by the summarizer is general and transferable inter-domain.

Following the objective of this thesis at large, our proposal in this chapter again addresses the lack of training data in the area of discourse parsing. We aim to overcome this limitation by generating a large number of discourse trees from a pre-trained summarization model, similar in spirit to Chapter 3.

5.2 Related Work

The Rhetorical Structure Theory (RST) [107], as described in Section 2.1, postulates complete document trees containing structure, nuclearity and relation properties between parts of text. Given the limited amount of available training data for RST-style discourse parsers, we previously tackled the data-sparsity issue through automatically generated discourse structures from distant supervision, showing that sentiment information and topic segmentation information can be used to infer discourse trees in Chapters 3 and 4, respectively.

In this chapter, we tackle the data sparsity issue by leveraging the task of extractive summarization. Recent neural summarization models are thereby typically based on transformers [99, 186]. One advantage of these models is that they learn the relationship between input units explicitly using the dot-product self-attention, which allows for some degree of exploration of the inner working of these complex and distributed models. Here, we investigate if the attention matrices of a transformer-based summarizer effectively capture discourse information (i.e., how strongly EDUs are related) and therefore
can be used to derive discourse trees for arbitrary documents. Admittedly, Liu and Lapata [97] and Liu et al. [102] presented preliminary work on inferring discourse structures from attention mechanisms, while training a neural model on auxiliary tasks, like text classification and summarization. However, they did not perform any comparison against ground-truth discourse trees as we do here. More importantly, we employ a more explicit approach to infer discourse structures, not as part of the learning process, but by extracting the trees after the summarization model is completely trained and applied to new documents.

While our focus is on discourse, extracting syntactic constituency and dependency trees from transformer-based models has been recently attempted in both, machine translation and language modelling. In machine translation, Mareček and Rosa [111] and Raganato and Tiedemann [135] show that trained machine translation models can capture syntactic information within their attention heads, using the CKY and CLE algorithms, respectively. For pre-trained language models, Wu et al. [169] propose a parameter-free probing method to construct syntactic dependency trees based on a pre-trained BERT model, briefly elaborating on possible applications to discourse. In contrast to our work, they do not directly use attention heads, but instead, build an impact matrix based on the distance between token representations. Furthermore, while their BERT-based model can not deal with long sequences, our model effectively deals with sequences of any length, especially critical for discourse analysis.

5.3 Our Discourse Inference Method

5.3.1 Framework Overview

Our main goal is to show the ability of a previously trained summarization model to be directly applied to the task of RST-style discourse parsing. Along this line, we explore the relationship between information learned by the transformer-based summarizer and the task of discourse parsing.
Figure 5.1: Overview of our tree inference method based on transformer self-attention matrices trained on the summarization task.

We leverage the synergies between units learned in the transformer model following our work in Xiao et al. [171], where we propose the use of a transformer document-encoder on top of a pre-trained BERT EDU encoder. The summarization model is presented in Figure 5.1 (left). In the transformer-based document encoder, each head internally contains a self-attention matrix, learned during the training of the summarization model, representing the relationship between EDUs (Figure 5.1 (center)). In this chapter, we analyze these learned self-attention matrices, not only to confirm our intuition that they contain relevant discourse information but also to computationally exploit such information for discourse parsing. We, therefore, generate a set of different (constituency/dependency) discourse trees from the self-attention matrices, focusing on different attributes of discourse, as shown in Figure 5.1 (right). Our generated constituency trees reveal the discourse tree structure without additional nuclearity and relation attributes. We complement the constituency interpretation of the self-attention matrices by additionally inferring a dependency tree, driven by the RST nuclearity attribute, critical for the importance prediction of text spans [57].

5.3.2 Parsing Algorithms

Formally, for an input document \( D = \{u_1, ..., u_n\} \) with \( n \) EDUs, each attention head returns an attention matrix \( A \in \mathbb{R}^{n \times n} \) where entry \( A_{ij} \) contains a score measuring how much the \( i \)-th EDU relies on the \( j \)-th EDU. Given those
bidirectional scores defining the relationship between every two EDUs in a document, we build a tree such that EDU pairs with higher reciprocal attention scores are more closely associated in the resulting tree. In the constituency case, this means that EDUs with higher mutual attention should belong to sub-trees on lower levels of the tree, while in the dependency case this implies that the path between such EDUs should contain fewer intermediate nodes. In essence, these requirements can be formalized as searching for the tree within the set of possible trees, which maximizes a combined score.

**Constituency Tree (C-Tree) Parsing**

To generate a constituency tree from the attention matrix, we follow a large body of previous work in discourse parsing (e.g., Joty et al. [72]), where constituency discourse trees are generated using the CKY algorithm [74]. Specifically, we fill a \( n \times n \) matrix \( P \in \mathbb{R}^{n \times n} \) generating the optimal tree in bottom-up fashion using the dynamic programming approach according to:

\[
P_{ij} = \begin{cases} 
0, & i > j \\
\sum_{k=1}^{n} (A_{ki}), & i = j \\
\max_{k=i}^{j-1} (P_{ik} + P_{(k+1)j}) \\
\quad + \text{avg}(A_{i; k;(k+1):j}) \\
\quad + \text{avg}(A_{(k+1):j; i;k})/2, & i < j 
\end{cases}
\]

where \( P_{ij} \) with \( i = j \) contains the overall importance of EDU \( i \), computed as the attention paid by others to unit \( i \). \( P_{ij} \) with \( i < j \) represents the score of the optimal sub-tree spanning from EDU \( i \) to EDU \( j \). We select the best combination of sub-trees \( k \), such that the sum of the left sub-tree spanning \([i : k]\) and the right one spanning \([(k + 1) : j]\), along with the average score of connections between the two sub-trees is maximized.

For example, to pick the structure of the sub-tree spanning EDUs \([3 : 5]\) (see Fig. 5.2), we need to decide between the potential sub-tree aggregation of \(((34)5)\) and \((3(45))\). The respective scores are computed based on the
Figure 5.2: Small-scale example of a CKY parsing matrix (left) aggregated according to the attention matrix (right). To aggregate the sub-tree containing EDUs 3-5, the green \((34)5\) and blue \((3(45))\) aggregations are considered based on their reciprocal self-attention scores in the self-attention matrix.

Following this algorithm, two sub-trees with a high attention score between them tend to be combined on lower levels of the tree, indicating they are more related in the discourse tree.

Besides the standard CKY algorithm described above, we also explore a hierarchical CKY approach with sentence and paragraph constraints. Specifically, we do not aggregate \(P_{ij}\) if the span \([i : j]\) crosses a sentence’s boundary where either sentence is incomplete. In the previous example, if \(EDU_3\) and \(EDU_4\) were in the same sentence, even if the score of the blue aggregation candidate was higher, we would choose the green sub-tree aggregation. Plausibly, this hierarchical approach is expected to further improve the performance, since additional structural constraints are added, following the ground-truth treebanks, where the vast majority of sentences and paragraphs are covered by a single-rooted, complete discourse sub-tree.

**Dependency Tree (D-Tree) Parsing**

For the dependency tree generation, we use the Eisner [37] and Chu-Liu-Edmonds algorithm [23, 35] to generate projective and non-projective depen-
dency trees, respectively. First, we convert the attention matrix $A$ into a fully connected graph $G = (N, E)$, where $N$ contains all the EDUs, and $e_{ij}$, indicating how much the $i$-th EDU influences the $j$-th EDU, corresponds to $A_{ji}$, which is the attention that the $j$-th EDU pays to the $i$-th EDU. Based on this graph, we apply the following algorithms:

**Eisner Algorithm:** We apply this dynamic programming algorithm to generate projective dependency trees. Thereby, we build a matrix $P \in \mathbb{R}^{n \times n \times 2 \times 2}$, in which the first and second dimensions contain the start and end indexes of sub-trees, similar to the CKY algorithm; while the third and fourth dimensions indicate whether the head is the start or the end unit and whether the sub-tree is completed. As done for constituency parsing, we also use a hierarchical version of Eisner’s algorithm, in which we restrict inter-sentence connections for incomplete sentence trees.

**Chu-Liu-Edmonds (CLE) Algorithm:** Originally proposed as a recursive approach to find the maximum spanning tree of a graph given its root, CLE can generate non-projective trees. In the unconstrained case, we simply follow the standard CLE algorithm, selecting the EDU with the highest importance score, i.e. $\text{root} = \arg \max_i \sum_{k=1}^n (A_{ki})$, as the root. From there, the algorithm selects the “optimal edges”, i.e. the maximum in-edges for each node except the root, while breaking cycles recursively.

Again, as we did for CKY and Eisner, we also apply the additional sentence constraint. Unlike the dynamic programming approaches, which build the trees in a bottom-up fashion and can directly be constrained to avoid cross-sentence aggregations of incomplete sentences, we need to modify the CLE algorithm to allow for sentence constraints. In particular, we first build a sentence graph $G^s = \{N^s, E^s\}$ from the EDU graph (Figure 5.3 (b)), in which $e^s_{sd} = \text{avg}_{s \in S, d \in D} e_{sd}$, and record the maximum edge corresponding to the edge between sentences, i.e. $\arg \max_{s \in S, d \in D} e_{sd}$. After that, we use the CLE algorithm within the sentence containing the root EDU as the root sentence to find the maximum spanning tree in $G^s$ (Figure 5.3 (c)).
add the corresponding EDU edges to the final tree (Figure 5.3 (d)). For example, the edge \((s_0, s_1)\) in \(G^s\) corresponds to the EDU edge \((e_0, e_2)\) in \(G\). Next, we treat nodes with incoming edges from other sentences as the root of the sentence itself and run the CLE algorithm within each sentence (Figure 5.3 (e)). The final tree (Figure 5.3 (f)) is formed as the combination of inter-sentence edges derived in sentence graph \(G^s\) and intra-sentence edges found within each sentence.

5.4 Evaluation

5.4.1 The Summarization Task

In order to show the generality of the discourse structures learned in the summarization model, we train our summarizer across a variety of datasets
Table 5.1: Key RST-style discourse dataset dimensions of the three evaluation corpora used in this chapter.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Docs</th>
<th>#EDU/doc</th>
<th>#Sent/doc</th>
<th>#words/doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST-DT</td>
<td>385</td>
<td>56.6</td>
<td>22.5</td>
<td>549</td>
</tr>
<tr>
<td>Instruction</td>
<td>176</td>
<td>32.7</td>
<td>19.5</td>
<td>318</td>
</tr>
<tr>
<td>GUM</td>
<td>127</td>
<td>107.0</td>
<td>45.0</td>
<td>874</td>
</tr>
</tbody>
</table>

and hyper-parameter settings. More specifically, we train on two separate, widely-used news corpora – CNN Daily Mail (CNNDM) [117] and NYT [140] –, as well as under three hyper-parameter settings with different numbers of layers and attention heads: (i) A simple model with 2 layers and a single head. (ii) 6 layers with 8 heads each, proposed in the original transformer model [156]. (iii) 2 layers with 8 heads each, constituting a middle ground between the previous two settings. By considering two corpora (CNNDM and NYT) and the three settings, we train six models.

5.4.2 Discourse Datasets

We assess the quality of the summarization generated trees on the three discourse datasets described in Section 2.2, namely RST-DT, Instruction-DT and GUM. Key dataset statistics of the respective corpora are further presented in Table 5.1.

All three discourse datasets contain ground-truth RST-style constituency trees. While all corpora contain potential non-binary sub-trees, Instruction-DT also includes multi-root documents. To account for these cases, we apply the right-branching binarization following Huber and Carenini [59]. Furthermore, we convert constituency trees with nuclearity into ground truth dependency trees using the algorithm proposed in Li et al. [93].

5.4.3 Evaluation Metric

To evaluate how well the generated trees align with ground-truth trees, we use RST Parseval Scores for constituency trees and Unlabeled Attachment

---

3The complete set of evaluations can be found in Appendix A.4.
<table>
<thead>
<tr>
<th>Model</th>
<th>No Cons.</th>
<th>Sent Cons.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A0</td>
<td>A1</td>
</tr>
<tr>
<td></td>
<td>A0</td>
<td>A1</td>
</tr>
<tr>
<td>RST-DT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNNNDM-2-1</td>
<td>61.2</td>
<td>59.7</td>
</tr>
<tr>
<td></td>
<td>76.2</td>
<td>74.6</td>
</tr>
<tr>
<td>CNNNDM-6-8</td>
<td>60.3</td>
<td>60.8</td>
</tr>
<tr>
<td></td>
<td>75.4</td>
<td>75.0</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>62.4</td>
<td>62.2</td>
</tr>
<tr>
<td></td>
<td>76.7</td>
<td>75.6</td>
</tr>
<tr>
<td>Random</td>
<td>58.6±0.1</td>
<td>74.1±0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A0</td>
<td>A1</td>
</tr>
<tr>
<td></td>
<td>A0</td>
<td>A1</td>
</tr>
<tr>
<td>CNNNDM-2-1</td>
<td>61.1</td>
<td>59.8</td>
</tr>
<tr>
<td></td>
<td>71.4</td>
<td>↓70.3</td>
</tr>
<tr>
<td>CNNNDM-6-8</td>
<td>60.3</td>
<td>61.2</td>
</tr>
<tr>
<td></td>
<td>71.2</td>
<td>70.9</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>61.3</td>
<td>61.3</td>
</tr>
<tr>
<td></td>
<td>71.3</td>
<td>↓70.0</td>
</tr>
<tr>
<td>Random</td>
<td>59.5±0.3</td>
<td>70.5±0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GUM</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A0</td>
<td>A1</td>
</tr>
<tr>
<td></td>
<td>A0</td>
<td>A1</td>
</tr>
<tr>
<td>CNNNDM-2-1</td>
<td>58.7</td>
<td>57.7</td>
</tr>
<tr>
<td></td>
<td>72.7</td>
<td>71.9</td>
</tr>
<tr>
<td>CNNNDM-6-8</td>
<td>58.9</td>
<td>59.3</td>
</tr>
<tr>
<td></td>
<td>72.4</td>
<td>72.7</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>59.6</td>
<td>59.3</td>
</tr>
<tr>
<td></td>
<td>72.2</td>
<td>71.6</td>
</tr>
<tr>
<td>Random</td>
<td>57.5±0.1</td>
<td>71.5±0.2</td>
</tr>
</tbody>
</table>

Table 5.2: RST Parseval Scores of generated constituency trees on the three datasets, expressed as 'Avg. ± Std'. ↓=Result is worse than Random. Results for Random are obtained by applying the parser to random matrices 10 times. A0\|A1 are the first two layers.

Score for dependency trees, measuring the ratio of matched spans and the ratio of matched dependency relations, respectively.

### 5.4.4 Main Results

For each model configuration, we run a set of experiments using the average attention matrix across all heads in a layer, i.e. $A_{avg} = \sum_h A^h / H$, with $H$ as the number of heads. This initial setup is intended to provide insights into the discourse information learned in each layer.

The results of the three tree-generation algorithms are shown in Tables 5.2, 5.3 and 5.4 along with the performance of a random baseline obtained
<table>
<thead>
<tr>
<th>Model</th>
<th>No Cons.</th>
<th>Sent Cons.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A0</td>
<td>A1</td>
</tr>
<tr>
<td>RST-DT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNNDM-2-1</td>
<td>23.7</td>
<td>↓ 4.8</td>
</tr>
<tr>
<td>CNNDM-6-8</td>
<td>↓ 7.9</td>
<td>20.5</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>15.7</td>
<td>12.5</td>
</tr>
<tr>
<td>Random</td>
<td>11.2±0.2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction</th>
<th>CNNDM-2-1</th>
<th>CNNDM-6-8</th>
<th>NYT-6-8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31.1</td>
<td>↓ 4.4</td>
<td>29.3</td>
</tr>
<tr>
<td></td>
<td>↓ 8.5</td>
<td>19.5</td>
<td>↓ 9.9</td>
</tr>
<tr>
<td></td>
<td>16.2</td>
<td>↓ 12.1</td>
<td>22.8</td>
</tr>
<tr>
<td>Random</td>
<td>13.1±0.3</td>
<td></td>
<td>19.3±0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GUM</th>
<th>CNNDM-2-1</th>
<th>CNNDM-6-8</th>
<th>NYT-6-8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>21.3</td>
<td>↓ 2.24</td>
<td>27.3</td>
</tr>
<tr>
<td></td>
<td>↓ 4.7</td>
<td>15.8</td>
<td>↓ 11.5</td>
</tr>
<tr>
<td></td>
<td>12.6</td>
<td>↓ 9.6</td>
<td>23.4</td>
</tr>
<tr>
<td>Random</td>
<td>10.4±0.2</td>
<td></td>
<td>19.2±0.3</td>
</tr>
</tbody>
</table>

**Table 5.3:** Unlabeled Attachment Scores of dependency trees generated by the Eisner algorithm. ↓=Result is worse than Random. A0/1 are the first two layers.

by running the algorithms on 10 random matrices. Here, we present the results of three selected models, limited to the performance of the first two layers for the 6-layer models, to allow for a direct comparison with the 2-layer models\(^4\). Across evaluations, the layer-wise performance within the same models are rather distinct, indicating that different properties are learned in specific layers. This finding is in line with previous work [95], especially given that the performance of each layer is consistent across the constituency and dependency parsing outputs for all datasets. Furthermore, the more layers the summarization model contains, the smaller the performance gap between layers becomes. We believe that this could be caused by the discourse

\(^4\)Results for all six models can be found in Appendix A.5
<table>
<thead>
<tr>
<th>Model</th>
<th>No Cons.</th>
<th>Sent Cons.</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A0</td>
<td>A1</td>
<td>A0</td>
<td>A1</td>
</tr>
<tr>
<td>RST-DT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNNDM-2-1</td>
<td><strong>21.6</strong></td>
<td><strong>29.3</strong></td>
<td>19.6</td>
<td></td>
</tr>
<tr>
<td>CNNDM-6-8</td>
<td>7.3</td>
<td>17.3</td>
<td>16.1</td>
<td>28.5</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>13.7</td>
<td>10.6</td>
<td>25.0</td>
<td>21.1</td>
</tr>
<tr>
<td>Random</td>
<td>1.7±0.1</td>
<td>18.7±0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNNDM-2-1</td>
<td><strong>28.1</strong></td>
<td><strong>37.4</strong></td>
<td>18.1</td>
<td></td>
</tr>
<tr>
<td>CNNDM-6-8</td>
<td>6.9</td>
<td>15.9</td>
<td>14.9</td>
<td>25.8</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>14.8</td>
<td>9.8</td>
<td>25.4</td>
<td>21.1</td>
</tr>
<tr>
<td>Random</td>
<td>2.9±0.2</td>
<td>17.9±0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNNDM-2-1</td>
<td><strong>19.5</strong></td>
<td><strong>28.8</strong></td>
<td>17.9</td>
<td></td>
</tr>
<tr>
<td>CNNDM-6-8</td>
<td>4.0</td>
<td>13.1</td>
<td>14.9</td>
<td>25.4</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>10.7</td>
<td>8.2</td>
<td>23.0</td>
<td>19.5</td>
</tr>
<tr>
<td>Random</td>
<td>0.9±0.05</td>
<td>17.0±0.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Unlabeled Attachment Scores of dependency trees generated by the CLE algorithm. ↓=Result is worse than Random. A0|1 are the first two layers.

information being further spread across different layers. Generally, we observe that models trained on the CNNDM dataset perform better than models trained on the NYT corpus, despite the larger size of the NYT dataset. Plausibly, the superior performance of our models trained on CNNDM reflects a higher diversity within documents in the CNNDM dataset.

Comparing the constituency tree performance in Table 5.2 against the dependency tree results in Tables 5.3 and 5.4, it becomes obvious that the improvement of the constituency parsing approach over the random baseline is much smaller than the improvements for the generated dependency trees. Presumably, this larger improvement of the dependency trees is due to the fact that dependency relationships (strongly encoding the nuclearity attribute)
are more directly related to the summarization task than plain structure information. This is in line with previous work on applying dependency trees to the summarization task [57, 176] and indicates that the learned attention matrices contain valid discourse information.

As for the two approaches to dependency parsing, although Eisner generally outperforms CLE, the improvement over random trees is larger for CLE. We believe that this effect is due to the reduced constraints imposed on the CLE algorithm, which is not limited to generating projective trees.

Considering all three methods, the results of the CLE-generated dependency tree seem most promising. A possible explanation is that both CKY and Eisner build the discourse tree in a bottom-up fashion with dynamic programming. This way, only local information is used on lower levels of the tree. On the other hand, the CLE algorithm uses global information, potentially more aligned with the summarization task, where all EDUs are considered to predict importance scores.

### 5.4.5 Performance of Heads

With all previous results relying on the average attention matrix across layers, we now analyze whether discourse information is evenly distributed across attention heads, or if a subset of the heads contains the majority of
discourse-related information.

We describe this analysis only for CLE for two reasons: (i) the summarization model seemingly captures more dependency-related discourse information than structure information; (ii) compared with Eisner, the CLE approach is more flexible, by also covering non-projective dependency trees.

Since the results across all summarization models are consistent, we only show the accuracy heatmap for the 

\textit{CNNDM-6-8} model on the three RST-style discourse datasets in Figure 5.4. Remarkably, for all three datasets, there is one head in the model capturing the vast majority of discourse information, especially in the unconstrained case. Furthermore, the performance of the best single attention head is much better than the one of the average attention matrix shown in Section 5.4.4, e.g. 34.53 compared to 19.51 on the GUM dataset without sentence constraints.

**5.4.6 Analysis of Generated Trees**

**Localness of Trees:** To verify that the generated trees are non-trivial, for instance simply connecting adjacent EDUs, we analyze the quality of the trees produced by the top-performing attention head in Figure 5.4. First, we separate all dependency relationships into two classes: local, holding between two adjacent EDUs, and distant, including all other relations between non-adjacent EDUs. We then compute the ratio of the correctly predicted dependencies which are local (Local Ratio Corr.), as well as the ratio of local dependencies in the generated (Local Ratio Ours), and ground-truth trees (Local Ratio GT). The results of this analysis are shown in Table 5.5. For all datasets, the ratio of correctly predicted local dependencies (Local Ratio Corr.) is larger than the ratio for distant relations, which seems reasonable, since local dependency predictions are easier than distant ones. Further, comparing “Local Ratio GT” and “Local Ratio Ours” without the sentence constraint shows that the number of local dependency relations in the ground-truth discourse trees are consistently larger than our predictions. This indicates that the discourse information learned in the attention matrices goes beyond the oftentimes predominant local information. However, even
without the sentence constraint, more than 40% of the relations are predicted as local, suggesting that the standard CLE approach can already capture local information well.

Adding the sentence constraint, we find that the local dependency ratio of the generated trees (Local Ratio Ours) further increases by more than 10% across all three datasets. This makes intuitive sense since the sentence constraint forces the generated trees to focus on local aspects within each sentence. To sum up, we find that the learned attention matrices contain both local and distant dependency information, although local dependency predictions perform better.

**Properties of Trees:** Following Ferracane et al. [40], we structurally inspect the generated dependency trees and compare them to gold trees on all three datasets. The comparison is presented in Table 5.6, showing the average branch width, average height, average leaf ratio (micro) and average

<table>
<thead>
<tr>
<th>Measurement(%)</th>
<th>No Cons.</th>
<th>Sent Cons.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RST-DT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Ratio Corr.</td>
<td>77.78</td>
<td>79.17</td>
</tr>
<tr>
<td>Local Ratio GT</td>
<td>53.22</td>
<td>53.22</td>
</tr>
<tr>
<td>Local Ratio Ours</td>
<td>46.52</td>
<td>58.35</td>
</tr>
<tr>
<td><strong>Instruction</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Ratio Corr.</td>
<td>81.15</td>
<td>84.90</td>
</tr>
<tr>
<td>Local Ratio GT</td>
<td>59.82</td>
<td>59.82</td>
</tr>
<tr>
<td>Local Ratio Ours</td>
<td>47.90</td>
<td>60.54</td>
</tr>
<tr>
<td><strong>GUM</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Ratio Corr.</td>
<td>77.99</td>
<td>80.20</td>
</tr>
<tr>
<td>Local Ratio GT</td>
<td>53.28</td>
<td>53.28</td>
</tr>
<tr>
<td>Local Ratio Ours</td>
<td>39.97</td>
<td>53.76</td>
</tr>
</tbody>
</table>

*Table 5.5: Measurements on the locality of the generated dependency trees. Numbers are %. Corr.=correct predictions, GT=ground-truth trees, Ours=generated tree.*
Table 5.6: Statistics of our generated trees and the gold standard trees in terms of the average branch width, average height, average leaf ratio (micro), average normalized arc length of the trees and percentage of the Vacuous trees.

<table>
<thead>
<tr>
<th></th>
<th>Branch</th>
<th>Height</th>
<th>Leaf</th>
<th>Arc</th>
<th>Vac.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RST-DT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours(Sent Cons)</td>
<td>1.50</td>
<td>27.06</td>
<td>0.37</td>
<td>0.10</td>
<td>3%</td>
</tr>
<tr>
<td>Ours(No Cons)</td>
<td>1.74</td>
<td>25.76</td>
<td>0.49</td>
<td>0.12</td>
<td>3%</td>
</tr>
<tr>
<td>GT Tree</td>
<td>2.10</td>
<td>8.19</td>
<td>0.51</td>
<td>0.13</td>
<td>2%</td>
</tr>
<tr>
<td><strong>Instruction</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours(Sent Cons)</td>
<td>1.56</td>
<td>15.74</td>
<td>0.39</td>
<td>0.13</td>
<td>3%</td>
</tr>
<tr>
<td>Ours(No Cons)</td>
<td>1.80</td>
<td>14.35</td>
<td>0.50</td>
<td>0.14</td>
<td>3%</td>
</tr>
<tr>
<td>GT Tree</td>
<td>1.59</td>
<td>8.49</td>
<td>0.41</td>
<td>0.15</td>
<td>1%</td>
</tr>
<tr>
<td><strong>GUM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours(Sent Cons)</td>
<td>1.61</td>
<td>44.94</td>
<td>0.40</td>
<td>0.05</td>
<td>0%</td>
</tr>
<tr>
<td>Ours(No Cons)</td>
<td>2.14</td>
<td>43.08</td>
<td>0.54</td>
<td>0.08</td>
<td>0%</td>
</tr>
<tr>
<td>GT Tree</td>
<td>2.02</td>
<td>12.17</td>
<td>0.51</td>
<td>0.04</td>
<td>0%</td>
</tr>
</tbody>
</table>

Looking at Table 5.6, it appears that our tree structures are similar to ground-truth trees in all regards presented, except the average height of trees. This indicates that our trees tend to be generally deeper than gold standard trees, despite having a similar branch width and leaf ratio. Furthermore, our trees become even deeper when using the additional sentence constraint. Plausibly, forcing each sentence to be contained in its own sub-tree makes shallow inter-sentential structures less likely.

A vacuous tree is a special tree in which the root is one of the first two EDUs, with all nodes being children of the root.
5.4.7 Model Sensitivity and Summarizer Quality

To investigate whether the discourse parsing performance of our model is consistent across different random initializations and to explore the influence of the summarizer on our inferred discourse structures, we perform additional experiments with the “CNNDM-6-8” model. Overall, we find that the performance is robust across random initializations. Interestingly, a single head consistently reaches the best performance across initialization and datasets; however, while the position of the top-performing head is not always the same, it is often located in the second layer of the model. Regarding the second experiment, exploring the sensitivity of discourse structures to the summarizer quality, we create decreasingly well-trained summarizers by limiting the number of training steps. As expected, we find that the captured discourse information decreases accordingly.

5.5 Contributions

In this chapter, we present a novel framework to infer discourse trees from transformer self-attention matrices trained on the task of extractive summarization. Experiments across models and datasets indicate that (i) dependency and structural discourse information are learned, (ii) the information is typically concentrated in a single head, and (iii) the attention matrices also cover long-distance discourse dependencies. Overall, consistent results across datasets and models suggest that the learned discourse information is general and transferable inter-domain.

In conclusion, our novel approach to extract discourse structures from the self-attention matrices of transformer models shows (i) empirical proof of readily available discourse information in summarization trained transformer models, especially regarding the nuclearity attribute, (ii) synergistic relationships between discourse structures and summarizations, exploited in the opposite direction by Marcu [109], here shown to be bidirectional – in line with our central hypothesis. Targeting the discourse nuclearity prediction, the work presented in this chapter aligns with our previous projects.

6 More details regarding this additional experiment can be found in Appendix A.6.
described in Chapters 3 and 4. While the auxiliary task of sentiment analysis has been shown to successfully capture low- and mid-level discourse trees, the topic segmentation auxiliary task achieved promising performance on high-level discourse structures (especially on/above paragraph-level). Exploring the summarization auxiliary task here, we show promising results for another important component of complete discourse trees, namely the nuclearity attribute. Furthermore, with this chapter showing the potential of the transformer architecture to implicitly capture some notion of discourse information when trained on the auxiliary task of extractive summarization, our work opens up a new direction to extract discourse information from other transformer-based models (e.g., presented in Chapter 6).

Casting light on Chapter 5 in the context of our hypothesis and key research questions, we corroborate the hypothesis for the auxiliary task of summarization by further answering (RQ1). We present a novel approach to effectively extract discourse structures from square matrices, here the self-attention matrices of a summarization trained transformer model. We thereby show discourse evaluations of our model, not reaching competitive performance compared to supervised discourse parsers yet, however, outperforming random baselines and, to some extent, showing the existence of relevant signals for the task of discourse tree generation.

Despite this chapter showing the potential of exploiting self-attention matrices of transformer models and their applicability to extract discourse structures, the performance achieved in our experiments is, admittedly, low. While this is partially expected, given the small-scale of the explored models and the specificity of the auxiliary task of summarization, the general approach presented in this work allows for a variety of follow-up work. Specifically, transformer-based architectures, now proven to be able to capture discourse-related information, can be further explored, for example using larger and more well-established models, such as BERT [28] and BART [89]. To follow up on our promising methodology shown in this chapter, we explore both, the BERT and BART large-scale, pre-trained and fine-tuned language models and their ability to capture valid discourse structures in the self-attention matrices in Chapter 6.
Part II

Discourse Inference from Self-Supervised Language Models
After discussing different auxiliary tasks and their ability to provide distantly supervised signals for the task of discourse tree generation in Part I, we now take this line of thought one step further, exploring a variety of self-supervised models and their alignment with the task of discourse structure inference. While we argue that distant supervision from auxiliary NLP tasks with available, natural and large-scale annotated datasets is a valid and valuable approach to infer (at least partial) discourse structures, one general limitation is the task-dependency of the inferred trees. To this end, language modelling can (in the broadest sense) be considered another auxiliary task, however, with a major advantage: Language modelling does not require any annotation. Instead, language modelling systems leverage the textual form itself, making them self-supervised. As a result of the lifted dependency on supervision signals, any available text (ranging from unstructured websites to well-curated books, news articles, and reviews) constitutes valid training data for self-supervised language models. Hence, these models and methods can leverage even larger datasets to infer task-independent discourse structures.

As such, self-supervised approaches have previously shown great potential for a wide range of NLP tasks, oftentimes relying on either autoencoder objectives [31], or pre-trained language models (PLMs), such as the popular BERT [28], BART [89], and GPT-X [17] [132, 133] architectures. In Part II, we take inspiration from these prior findings, proposing two projects exploiting self-supervised datasets to infer discourse structures. Specifically, we take a look at the self-attention matrices of pre-trained and fine-tuned PLMs, similar to the method proposed in Chapter 5, to investigate the amount of discourse-related information encapsulated in the top-performing BERT and BART models (see Chapter 6). Despite their strong performance in generating fluent text, one of the key weaknesses of standard PLMs is the lack of linguistic priors [13]. Following the argument in Bender and Koller [13], claiming that solely using the textual form to train a computational model is insufficient to learn meaning, we further explore an autoencoder-based, more linguistically inspired language modelling approach in Chapter 7.
Chapter 6

Discourse Inference from Pre-Trained Language Models

The work presented in this chapter was published and presented at NAACL 2022 [63]. Throughout the project, I was the lead investigator, responsible for all major areas of concept formation, statement of research questions, data collection, implementation as well as paper composition. Giuseppe Carenini was the supervisory author, involved throughout the project in the areas of concept formation, discussions, and paper composition.

6.1 Motivation

Transformer-based machine learning models are an integral part of many recent improvements in Natural Language Processing (NLP). With their rise spearheaded by Vaswani et al. [156], the pre-training/fine-tuning paradigm has gradually replaced previous approaches based on architecture engineering, with transformer models such as BERT [28], BART [89], RoBERTa [101] and others delivering state-of-the-art performance on a wide variety of tasks. Besides their strong empirical results on most real-world problems, such as summarization [172, 184], question-answering [71, 123] and sentiment analysis
uncovering what kind of linguistic knowledge is captured by this new type of pre-trained language models (PLMs) has become a prominent question by itself. As part of this line of research, called BERTology [139], researchers explore the amount of linguistic understanding encapsulated in PLMs, exposed through either external probing tasks [85, 135, 187] or unsupervised methods [124, 169]. Previous work thereby either focuses on analyzing the syntactic structures (e.g., Hewitt and Manning [56], Wu et al. [169]), relations [125], ontologies [113] or, to a more limited extent, discourse-related behaviour [85, 124, 187].

Generally speaking, while most previous BERTology works have focused on either sentence-level phenomena or connections between adjacent sentences, large-scale semantic and pragmatic structures (oftentimes represented as discourse trees or graphs) have been less explored. These structures (e.g., discourse trees) play a fundamental role in expressing the intent of multi-sentential documents and, not surprisingly, have been shown to benefit many NLP tasks such as summarization [44], sentiment analysis [14, 58, 119] and text classification [69].

With multiple discourse theories proposed in the past, we follow the RST framework [107], described in Section 2.1, in this chapter. As such, we extend the area of BERTology research with novel insights regarding the amount of intrinsic discourse information captured in established PLMs. More specifically, we aim to better understand to what extent RST-style discourse information is stored as latent trees in encoder self-attention matrices\(^1\). While we focus on the RST formalism in this work, our presented methods are theory-agnostic and, hence, also applicable to alternative discourse theories. Our contributions in this chapter are:

1. A novel approach to extract discourse information from arbitrarily long documents with standard transformer models, inherently limited by their input size. This is a non-trivial issue, which has been bypassed in previous work through the use of proxy tasks.

\(^1\)Please note that we focus on discourse structure and nuclearity here, leaving relation classification for future work.
2. An exploration of discourse information locality across pre-trained and fine-tuned language models, finding that discourse structures are consistently captured in a fixed subset of self-attention heads.

3. An in-depth analysis of the discourse quality in pre-trained language models and their fine-tuned extensions. We compare constituency and dependency structures of 2 PLMs fine-tuned on 4 tasks and 7 fine-tuning datasets to gold-standard discourse trees, finding that the captured discourse structures outperform simple baselines by a large margin, even showing superior performance compared to distantly supervised models.

4. A similarity analysis between PLM inferred discourse trees and supervised, distantly supervised and simple baselines. We reveal that PLM constituency discourse trees do align relatively well with previously proposed supervised models, but also capture complementary information.

5. A detailed look at information redundancy in self-attention heads to better understand the structural overlap between self-attention matrices and models. Our results indicate that similar discourse information is consistently captured in the same heads, even across fine-tuning tasks.

6.2 Related Work

At the base of our work are two of the most popular and frequently used PLMs: BERT [28] and BART [89]. We choose these two popular approaches in our study due to their complementary nature (encoder-only vs. encoder-decoder) and based on previous work by Zhu et al. [187] and Koto et al. [85], showing the effectiveness of BERT and BART models for discourse-related tasks.

Our work in this chapter is further related to the field of discourse parsing, which slowly shifted to successfully incorporate a variety of PLMs into the process of discourse prediction. Casting the task into a “local” problem using only partial information, previous models have incorporated
ELMo embeddings \[82\] as well as the XLNet \[120\], BERT \[86\], RoBERTa \[51\] and SpanBERT \[49\] models. Here, we follow the intuition that the true benefit of discourse information emerges when complete documents are considered, leading us to propose a new approach to connect PLMs and discourse structures in a “global” manner by enabling systems to process arbitrarily long documents.

Aiming to better understand what information is captured in PLMs, the line of \textit{BERTology} research has recently emerged \[139\], with early work mostly focusing on the syntactic capacity of PLMs \[56, 67, 78\], in parts also exploring the internal workings of transformer-based models (e.g., self-attention matrices \[111, 135\]). More recent work started to explore the alignment of PLMs with discourse information, encoding semantic and pragmatic knowledge. Along those lines, Wu et al. \[169\] present a parameter-free probing task for both, syntax and discourse. With their tree inference approach being computationally expensive and limited to the exploration of the outputs of the BERT model, we significantly extend this line of research by exploring the internal self-attention matrices of PLMs with a more computationally feasible approach. More traditionally, Zhu et al. \[187\] use 24 hand-crafted rhetorical features to execute three different supervised probing tasks, showing promising performance of the BERT model. Similarly, Pandia et al. \[124\] aim to infer pragmatics through the prediction of discourse connectives by analyzing the model inputs and outputs and Koto et al. \[85\] analyze discourse in seven PLMs through seven supervised probing tasks, finding that BART and BERT contain most information related to discourse. In contrast to the approach taken by both Zhu et al. \[187\] and Koto et al. \[85\], we use an unsupervised methodology to test the amount of discourse information stored in PLMs (which can also conveniently be used to infer discourse structures for new and unseen documents) and extend the work by Pandia et al. \[124\] by taking a closer look at the internal workings of the self-attention component.

Looking at prior work analyzing the amount of discourse information in

\[^2\text{For more information on pre-trained language models and previous approaches exploring their ability to capture discourse information, please refer to Section 2.4.4.}\]
PLMs, structures are solely explored through the use of proxy tasks, such as connective prediction [124], relation classification [87], and others [85]. However, despite the difficulties of encoding arbitrarily long documents, we believe that to systematically explore the relationship between PLMs and discourse, considering complete documents is imperative. Along these lines, recent work started to tackle the inherent input-length limitation of general transformer models through additional recurrence in the Transformer-XL model [27], compression modules [134] or sparse patterns (e.g., as in the Reformer [81], BigBird [181], and Longformer [12] models). While all these approaches to extend the maximum document length of transformer-based models are important to create more globally inspired models, the document-length limitation is still practically and theoretically in place, with models being limited to a fixed number of pre-defined tokens the model can process. Furthermore, with many proposed systems still based on more established PLMs (e.g., BERT) and with no single dominant solution for the general problem of the input length limitation yet, we believe that even with the restriction being actively tackled, an in-depth analysis of traditional PLMs with discourse is highly valuable to establish a solid understanding of the amount of semantic and pragmatic information captured.

Besides the described BERTology work, we got encouraged to explore fine-tuned extensions of standard PLMs through previous work showing the benefit of discourse parsing for many downstream tasks, such as summarization [44], sentiment analysis [14, 58, 119] and text classification [69]. Conversely, we draw further intuition from Chapters 3 and 5 presented in this thesis, showing promising results when inferring discourse structures from the related downstream tasks of sentiment analysis and summarization, respectively. Given this bidirectional synergy between discourse and the mentioned downstream tasks, we move beyond traditional experiments focusing on standard PLMs and additionally explore discourse structures of PLMs fine-tuned on a variety of auxiliary tasks.
6.3 Discourse Extraction Method

With PLMs rather well-analyzed according to their syntactic capabilities, large-scale discourse structures have been less explored. One reason for this is the input length constraint of transformer models. While this is generally not prohibitive for intra-sentence syntactic structures (e.g., presented in Wu et al. [169]), it does heavily influence large-scale discourse structures, operating on complete (potentially long) documents. Overcoming this limitation is non-trivial, since traditional transformer-based models only allow for fixed, short inputs.

Aiming to systematically explore the ability of PLMs to capture discourse, we investigate a novel way to effectively extract discourse structures from the self-attention component of the BERT and BART models. We thereby extend our previously proposed tree-generation methodology from Chapter 5 to support the input length constraints of standard PLMs using a sliding-window approach in combination with matrix frequency normalization and an EDU aggregation method. Figure 6.1 visualizes the complete process on a small-scale example with 3 EDUs and 7 sub-word embeddings.

![Figure 6.1: Small-scale example of the discourse extraction approach.Purple=EDUs, green=sub-word embeddings, red=input slices of size $t_{max}$, orange=PLM, blue=self-attention values, grey-scale=frequency count.](image)
The Tree Generation Procedure we previously proposed in Chapter 5 explores a two-stage approach to obtain discourse structures from a transformer model, by-passing the input-length constraint. Using the intuition that the self-attention score between any two EDUs is an indicator of their semantic/pragmatic relatedness, influencing their distance in a projective discourse tree, they use the CKY dynamic programming approach [74] to generate constituency trees based on the internal self-attention of the transformer model. To generate dependency trees, we apply the same intuition used to infer discourse trees with the Eisner algorithm [37]. Since we explore the discourse information captured in standard PLMs, we can’t directly transfer our two-stage approach in Chapter 5, first encoding individual EDUs using BERT and subsequently feeding the dense representations into a fixed-size transformer model. Instead, we propose a new method to overcome the length limitation of the transformer model.

The Sliding-Window Approach is at the core of our new methodology to overcome the input-length constraint. We first tokenize arbitrarily long documents with \( n \) EDUs \( E = \{e_1, ..., e_n\} \) into the respective sequence of \( m \) sub-word tokens \( T = \{t_1, ..., t_m\} \) with \( n \ll m \), according to the PLM tokenization method (WordPiece for BERT, Byte-Pair-Encoding for BART), as shown at the top of Figure 6.1. Using the sliding window approach, we subdivide the \( m \) sub-word tokens into sequences of maximum input length \( t_{\text{max}} \), defined by the PLM (\( t_{\text{max}} = 512 \) for BERT, \( t_{\text{max}} = 1024 \) for BART). Using a stride of 1, we generate \( (m - t_{\text{max}}) + 1 \) sliding windows \( W \), feed them into the PLM, and extract the resulting \( t_{\text{max}} \times t_{\text{max}} \) partial self-attention matrices (\( M_P \) in Figure 6.1) for a specific self-attention head.

The Frequency Normalization Method allows us to combine the partially overlapping self-attention matrices \( M_P \) into a single document-level matrix \( M_D \) of size \( m \times m \). To this end, we combine multiple overlapping

---

3 For more information on the general tree-generation approach using the Eisner algorithm we refer interested readers to Chapter 5.

4 We omit the self-attention indexes for better readability.
windows, generated due to the stride size of 1, by adding up the self-attention cells, while keeping track of the number of overlaps in a separate $m \times m$ frequency matrix $M_F$. We then divide $M_D$ by the frequency matrix $M_F$, to generate a frequency normalized self-attention matrix $M_A$ (see bottom of Figure 6.1).

The EDU Aggregation is the final processing step to obtain the document-level self-attention matrix. In this step, the $m$ sub-word tokens $T = \{t_1, ..., t_m\}$ are aggregated back into $n$ EDUs $E = \{e_1, ..., e_n\}$ by computing the average bidirectional self-attention score between any two EDUs in $M_A$. For example, in Figure 6.1 we aggregate the scores in cells $M_A[0:1, 5:6]$ to compute the final output of cell $[0, 2]$ (purple matrix in Figure 6.1) and $M_A[5:6, 0:1]$ to generate the value of cell $[2, 0]$. This way, we obtain the average bidirectional self-attention scores between $EDU_1$ and $EDU_3$. We use the resulting $n \times n$ matrix as the input to the CKY/Eisner discourse tree generation methods.

6.4 Experimental Setup

6.4.1 Pre-Trained Models

We select the BERT-base (110 million parameters) and BART-large (406 million parameters) models for our experiments. We choose these models for their diverse objectives (encoder-only vs. encoder-decoder), popularity for diverse fine-tuning tasks, and their prior successful exploration in regards to discourse information [85, 187]. For the BART-large model, we limit our analysis to the encoder, as motivated by Koto et al. [85].

6.4.2 Fine-Tuning Tasks and Datasets

We explore the BERT model fine-tuned on two classification tasks, namely sentiment analysis and natural language inference (NLI). For our analysis on BART, we select the abstractive summarization and question-answering tasks. Table 6.1 summarizes the 7 datasets used to fine-tune PLMs in this
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Task</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDB[29]</td>
<td>Sentiment</td>
<td>Movie Reviews</td>
</tr>
<tr>
<td>Yelp[185]</td>
<td>Sentiment</td>
<td>Reviews</td>
</tr>
<tr>
<td>SST-2[145]</td>
<td>Sentiment</td>
<td>Movie Reviews</td>
</tr>
<tr>
<td>MNLI[165]</td>
<td>NLI</td>
<td>Range of Genres</td>
</tr>
<tr>
<td>CNN-DM[117]</td>
<td>Summarization</td>
<td>News</td>
</tr>
<tr>
<td>XSUM[118]</td>
<td>Summarization</td>
<td>News</td>
</tr>
</tbody>
</table>

Table 6.1: The seven fine-tuning datasets used in this work along with the underlying tasks and domains.

work, along with their underlying tasks and domains\(^5\)

6.4.3 Evaluation Treebanks

All evaluations shown in this chapter are executed on the 38 and 20 documents in the RST-DT and GUM test sets, described in Section 2.2, to be comparable with previous baselines and supervised models. A similarly-sized validation set is used where mentioned to determine the best-performing self-attention head.

6.4.4 Baselines and Evaluation Metrics

Simple Baselines: We compare the inferred constituency trees against right- and left-branching structures. For dependency trees, we evaluate against the simple chain and inverse chain structures.

Distantly Supervised Baselines: We compare our results obtained in this chapter against our previous approach presented in Chapter 5, using similar CKY and Eisner tree-generation methods to infer constituency and dependency tree structures from a summarization model trained on the CNN-DM and New York Times (NYT) corpora (referred to as Sum\text{CNN-DM} and Sum\text{NYT}\(^6\))

\(^5\)We exclusively analyze published models provided on the Huggingface platform, further specified and linked to in Appendix B.1

\(^6\)www.github.com/Wendy-Xiao/summ_guided_disco_parser
Supervised Baseline: We select the popular Two-Stage discourse parser [160] as our supervised baseline, due to its strong performance, available model checkpoints and code[7], as well as the traditional architecture. We use the published Two-Stage parser checkpoint on RST-DT (from here on called \textit{Two-Stage}_{RST-DT}) and re-train the discourse parser on GUM (\textit{Two-Stage}_{GUM}). We convert the generated constituency structures into dependency trees following Li et al. [93].

Evaluation Metrics: We apply the original parseval score to compare discourse constituency structures with gold-standard treebanks, as argued in Morey et al. [115]. To evaluate the generated dependency structures, we use the Unlabeled Attachment Score (UAS).

6.5 Evaluation

6.5.1 Discourse Locality

Our discourse tree generation approach described in Section 6.3 uses self-attention matrices to infer discourse structures. The standard BERT model contains 144 of those self-attention matrices (12 layers, 12 self-attention heads each), all of which potentially encode discourse structures. For the BART model, this number is even higher, consisting of 12 layers with 16 self-attention heads each. With prior work suggesting the locality of discourse information in PLMs (e.g., Mareček and Rosa [111], Raganato and Tiedemann [135], Xiao et al. [173]), we analyze every self-attention matrix individually to gain a better understanding of their alignment with discourse information.

Besides investigating standard PLMs, we also explore the robustness of discourse information across fine-tuning tasks. We believe that this is an important step to better understand if the captured discourse information is general and robust, or if it is “re-learned” from scratch for downstream tasks. To the best of our knowledge, no previous analysis of this kind has been performed in the literature.

Figure 6.2 shows the constituency and dependency structure overlap of

\footnote{www.github.com/yizhongw/StageDP}
the generated discourse trees from individual self-attention heads with the gold-standard tree structures of the GUM dataset\textsuperscript{8} The heatmaps clearly show that constituency discourse structures are mostly captured in higher layers, while dependency structures are more evenly distributed. Comparing the patterns between models, we find that, despite being fine-tuned on different downstream tasks, the discourse information is consistently encoded in the same self-attention heads. Even though the best-performing self-attention matrix is not consistent, discourse information is clearly captured in a “local” subset of self-attention heads across all presented fine-tuning tasks. This plausibly suggests that the discourse information in pre-trained BERT and BART models is robust and general, requiring only minor adjustments

\textsuperscript{8}The analysis on RST-DT shows similar trends and can be found in Appendix B.2.
<table>
<thead>
<tr>
<th>Model</th>
<th>RST-DT</th>
<th>GUM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Span</td>
<td>UAS</td>
</tr>
<tr>
<td>BERT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rand. init</td>
<td>↓ 25.5</td>
<td>↓ 13.3</td>
</tr>
<tr>
<td>PLM</td>
<td>• 35.7</td>
<td>• 45.3</td>
</tr>
<tr>
<td>+ IMDB</td>
<td>↓ 35.4</td>
<td>↓ 42.8</td>
</tr>
<tr>
<td>+ Yelp</td>
<td>↓ 34.7</td>
<td>↓ 42.3</td>
</tr>
<tr>
<td>+ SST-2</td>
<td>↓ 35.5</td>
<td>↓ 42.9</td>
</tr>
<tr>
<td>+ MNLI</td>
<td>↓ 34.8</td>
<td>↓ 41.8</td>
</tr>
<tr>
<td>BART</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rand. init</td>
<td>↓ 25.3</td>
<td>↓ 12.5</td>
</tr>
<tr>
<td>PLM</td>
<td>• 39.1</td>
<td>• 41.7</td>
</tr>
<tr>
<td>+ CNN-DM</td>
<td>↑ 40.9</td>
<td>↑ 44.3</td>
</tr>
<tr>
<td>+ XSUM</td>
<td>↑ 40.1</td>
<td>↑ 41.9</td>
</tr>
<tr>
<td>+ SQuAD</td>
<td>↑ 40.1</td>
<td>↑ 43.2</td>
</tr>
<tr>
<td>Baselines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RB / Chain</td>
<td>9.3</td>
<td>40.4</td>
</tr>
<tr>
<td>LB / Chain(^{-1})</td>
<td>7.5</td>
<td>12.7</td>
</tr>
<tr>
<td>Two-Stage(\text{MEGA-DT}[61])</td>
<td>55.8</td>
<td>40.0</td>
</tr>
<tr>
<td>Sum(\text{CNN-DM})</td>
<td>21.4</td>
<td>20.5</td>
</tr>
<tr>
<td>Sum(\text{NYT})</td>
<td>24.0</td>
<td>15.7</td>
</tr>
<tr>
<td>Two-Stage(\text{RST-DT})</td>
<td>72.0</td>
<td>71.2</td>
</tr>
<tr>
<td>Two-Stage(\text{GUM})</td>
<td>65.4</td>
<td>61.7</td>
</tr>
</tbody>
</table>

Table 6.2: Original parseval (Span) and Unlabelled Attachment Score (UAS) of the single best-performing self-attention matrix of the BERT and BART models compared with baselines and previous work. ↑, •, ↓ indicate better, same, or worse performance compared to the PLM. rand. init=Randomly initialized transformer model of similar architecture as the PLM, RB=Right-Branching, LB=Left-Branching, Chain\(^{-1}\)=Inverse chain.

depending on the fine-tuning task.
6.5.2 Discourse Quality

We now focus on assessing the discourse information captured in the single best-performing self-attention head. In Table 6.2, we compare the discourse structure quality of pre-trained and fine-tuned PLMs in the context of supervised models, distantly supervised approaches and simple baselines. We show the oracle-picked best head on the test set, analyzing the upper bound for the potential performance of PLMs on RST-style discourse structures. This is not a realistic scenario, as the best-performing head is generally not known a priori. Hence, we also explore the performance using a small-scale validation set to pick the best-performing self-attention matrix. In this more realistic scenario for discourse parsing, we find that scores on average drop by 1.55 points for BERT and 1.33% for BART compared to the oracle-picked performance of a single self-attention matrix. We show detailed results of this degradation in Appendix B.3. Our results in Table 6.2 are separated into three sub-tables, showing the results for BERT, BART and baseline models on the RST-DT and GUM treebanks, respectively. In the BERT and BART sub-table, we further annotate each performance with ↑, •, and ↓, indicating the relative performance to the standard pre-trained model as superior, equal, or inferior.

Taking a look at the top sub-table (BERT) we find that, as expected, the randomly initialized transformer model achieves the worst performance. Fine-tuned models perform equal to or worse than the standard PLM. Despite the inferior results of the fine-tuned models, the drop is rather small, with the sentiment analysis models consistently outperforming NLI. This seems reasonable, given that the sentiment analysis objective is intuitively more aligned with discourse structures (e.g., long-form reviews with potentially complex rhetorical structures) than the between-sentence NLI task, not involving multi-sentential text.

In the center sub-table (BART), a different trend emerges. While the worst performing model is still (as expected) the randomly initialized system, fine-tuned models mostly outperform the standard PLM. Interestingly, the

\[ \text{For a more detailed analysis of the min., mean, median and max. self-attention performances see Appendix B.3.} \]
model fine-tuned on the CNN-DM corpus consistently outperforms the BART baseline, while the XSUM model performs better on all but the GUM dependency structure evaluation. On one hand, the superior performance of both summarization models on the RST-DT dataset seems reasonable, given that the fine-tuning datasets and the evaluation treebank are both in the news domain. The strong results of the CNN-DM model on the GUM treebank, yet the inferior performance of XSUM, potentially hint towards dependency discourse structures being less prominent when fine-tuning on the extreme summarization task, compared to the longer summaries in the CNN-DM corpus. The question-answering task evaluated through the SQuAD fine-tuned model underperforms the standard PLM on GUM, however, reaches superior performance on RST-DT. Since the SQuAD corpus is a subset of Wikipedia articles, more aligned with news articles than the 12 genres in GUM, we believe the stronger performance on RST-DT (i.e., news articles) is again reasonable, yet shows weaker generalization capabilities across domains (i.e., on the GUM corpus). Interestingly, the question-answering task seems more aligned with dependency than constituency trees, in line with what would be expected from a factoid-style question-answering model, focusing on important entities, rather than global constituency structures.

Directly comparing the BERT and BART models, the former performs better on three out of four metrics. At the same time, fine-tuning hurts the performance of BERT, however, improves BART models. Plausibly, these seemingly unintuitive results may be caused by the following co-occurring circumstances: (i) The inferior performance of BART can potentially be attributed to the decoder component capturing parts of the discourse structures, as well as the larger number of self-attention heads “diluting” the discourse information. (ii) The different trends regarding fine-tuned models might be directly influenced by the input-length limitation to 512 (BERT) and 1024 (BART) sub-word tokens during the fine-tuning stage, hampering the ability to capture long-distance semantic and pragmatic relationships. This, in turn, limits the amount of discourse information captured, even for document-level datasets (e.g., Yelp, CNN-DM, SQuAD). With this restriction being more prominent in BERT, it potentially explains the comparably low
performance of the fine-tuned models.

Finally, the bottom sub-table puts our results in the context of previously proposed supervised and distantly supervised models, as well as simple baselines. Compared to simple right- and left-branching trees (Span), the PLM-based models reach clearly superior performance. Looking at the chain/inverse chain structures (UAS), the improvements are generally lower, however, the vast majority still outperforms the baseline. Comparing the first two sub-tables against completely supervised methods (Two-StageRST-DT, Two-StageGUM), the BERT- and BART-based models are, unsurprisingly, inferior. Lastly, compared to the distantly supervised SumCNN-DM and SumNYT models proposed in Chapter 5, the PLM-based discourse performance shows clear improvements over the 6-layer, 8-head standard transformer.

6.5.3 Discourse Similarity

Further exploring what kind of discourse information is captured in the PLM self-attention matrices, we directly compare the emergent discourse structures with trees inferred from existing discourse parsers and simple baselines. This way, we aim to better understand if the information encapsulated in PLMs is complementary to existing methods, or if the PLMs solely capture trivial discourse phenomena and simple biases (e.g., resemble right-branching constituency trees). Since the GUM dataset contains a more diverse set of test documents (12 genres) than the RST-DT corpus (exclusively news articles), we perform our experiments from here on exclusively on the GUM treebank.

Figure 6.3 shows the micro-average structural overlap of discourse constituency (left) and dependency (right) trees between the PLM-generated discourse structures and existing methods, baselines, as well as gold-standard trees. Noticeably, the generated constituency trees (on the left) are most aligned with the structures predicted by supervised discourse parsers, showing only minimal overlap to simple structures (i.e., right- and left-branching trees). Taking a closer look at the generated dependency structures presented on the right side in Figure 6.3, the alignment between PLM inferred
discourse trees and the simple chain structure is predominant, suggesting a potential weakness of the Eisner algorithm to extract discourse from the BERT and BART models. Not surprisingly, the highest overlap between PLM-generated trees and the chain structure occurs when fine-tuning on the CNN-DM dataset, well-known to contain a strong lead bias [175].

To better understand if the PLM-based constituency structures are complementary to existing, supervised discourse parsers, we further analyze the correctly predicted overlap. Specifically, we compute the intersection between PLM-generated structures and gold-standard trees as well as previously proposed models and the gold-standard. Subsequently, we intersect the two resulting sets (e.g., BERT ∩ Gold Trees ↔ Two-Stage (RST-DT) ∩ Gold Trees). This way, we explore if the correctly predicted PLM discourse structures are a subset of the correctly predicted trees by supervised approaches, or if complementary discourse information is captured. We find that ≈20% and ≈16% of the correctly predicted constituency and dependency structures of our PLM discourse inference approach are not captured by supervised models, making the exploration of ensemble methods a promising future
avenue. A detailed version of Figure 6.3, as well as more specific results regarding the correctly predicted overlap of discourse structures, are shown in Appendix B.5.

### 6.5.4 Discourse Redundancy

Up to this point, our quantitative analysis of the ability of PLMs to capture discourse information has been limited to the single best-performing head. However, looking at individual models, the discourse performance distribution in Figure 6.2 suggests that a larger subset of self-attention heads performs similarly well (i.e., there are several dark purple cells in each heatmap). This leads to the interesting question if the information captured in different, top-performing self-attention heads is redundant or complementary. Similarly, Figure 6.2 indicates that the same heads perform well across different fine-tuning tasks, leading to the question if the discourse structures captured in a single self-attention matrix of different fine-tuned models are consistent or vary depending on the underlying task. Hence, we take a detailed look at the similarity of model self-attention heads in regards to their alignment with discourse information and explore if (i) the top-performing heads $h_i, ..., h_k$ of a specific model $m_m$ capture redundant discourse structures, and if (ii) the discourse information captured by a specific head $h_i$ across different models

---

**Figure 6.4:** Nested aggregation approach for discourse similarity. (a) Grey cells contain same head, white cells indicate different heads. (b) Grey cells contain same model, white cells indicate different models. Column indices equal row indices.
(a) Constituency Similarity  
(b) Dependency Similarity

**Figure 6.5:** BERT self-attention similarities on GUM. Top: Visual analysis of head-aligned (I&III) and model-aligned (II&IV) heatmaps. Yellow=high structural overlap, purple=low structural overlap. Bottom: Aggregated similarity of different models, same models, different heads and same heads showing the min, max and quartiles of the underlying distribution. *Significantly better than respective ≠Head /≠Model performance with p-value < 0.05.

$m_m, ..., m_o$ contain similar discourse trees.

We pick the top 10 best-performing self-attention matrices of each model, remove self-attention heads that don’t appear in at least two models (since no comparisons can be made) and compare the generated discourse structures in a nested aggregation approach. Figure 6.4 shows a small-scale example of our nested visualization methodology. For the self-attention head-aligned approach (Figure 6.4 (a)), high similarity values (calculated as the micro-average structural overlap) along the diagonal (grey cells) would be expected if the same head $h_i$ encodes consistent discourse information across different fine-tuning tasks and datasets. Inversely, the model-aligned matrix (Figure 6.4 (b)) should show high values along the diagonal if different heads $h_i, ..., h_k$ in the same model $m_k$ capture redundant discourse information. Besides
the visual inspection methodology presented in Figure 6.4, we also compare aggregated similarities between the same head (=Head) against different heads (≠Head) and between the same model (=Model) against different models (≠Model) (i.e., grey cells (=) and white cells (≠) in Figure 6.4 (a) and (b)). In order to assess the statistical significance of the resulting differences in the underlying distributions, we compute a two-sided, independent t-test between same/different models and same/different heads\(^{10}\).

The resulting redundancy evaluations for BERT are presented in Figure 6.5\(^{11}\). It appears that the same self-attention heads \(h_i\) consistently encode similar discourse information across models indicated by: (i) High similarities (yellow) along the diagonal in heatmaps I&III and (ii) through the statistically significant difference in distributions at the bottom of Figure 6.5 (a) and (b). However, different self-attention heads \(h_i, ..., h_k\) of the same model \(m_m\) encode different discourse information (heatmaps II&IV). While the trend is stronger for constituency tree structures, there is a single dependency self-attention head which does generally not align well between models and heads (purple line in heatmap III). Plausibly, this specific self-attention head might encode fine-tuning task-specific discourse information, making it a prime candidate for further investigations in future work. Furthermore, the similarity patterns observed in Figure 6.5 (a) and (b) point towards an opportunity to combine model self-attention heads to improve the discourse inference performance compared to the scores shown in Table 6.2, where each self-attention head was assessed individually.

### 6.6 Contributions

In this chapter, we extend the line of BERTology work by focusing on the important, yet less explored, alignment of pre-trained and fine-tuned PLMs with discourse structures. We propose a novel approach to infer discourse information for arbitrarily long documents by combining the largest possible sliding window for every input unit, maximizing the contextual information

---

\(^{10}\)Prior to running the t-test we confirm similar variance and the assumption of normal distribution (Shapiro-Wilk test).

\(^{11}\)Evaluations for BART can be found in Appendix B.6.
throughout documents for limited input size transformer models. In our experiments, we find that the captured discourse information is consistent, local and general, even across a collection of fine-tuning tasks. Comparing our results against simple baselines, we find that the generated structures are non-trivial. Further evaluating the inferred discourse trees against supervised and distantly supervised models, the obtained performance is promising, partially outperforming distantly supervised approaches. To gain a better understanding of the information captured, we explore the structural overlap of the inferred trees with supervised, distantly supervised and simple baselines, finding that constituency discourse trees align well with supervised models, however, contain complementary discourse information. Lastly, we individually explore self-attention matrices to analyze information redundancy. We find that similar discourse structures are consistently captured in the same heads across different fine-tuning tasks, confirming the robust representation of discourse information in pre-trained language models. Furthermore, we show that different, top-performing heads within the same model capture diverse discourse structures, pointing towards opportunities to combine information from these heads to improve the performance compared to our single-head results presented in this chapter.

In conclusion, we show first insights and evidence regarding the potential of pre-trained language models (namely BERT and BART) as domain-independent sources for general-purpose discourse structures. As such, this work extends our previous efforts in using distant supervision signals from auxiliary tasks (e.g., sentiment analysis in Chapter 3, topic segmentation in Chapter 4 and summarization in Chapter 5) into the area of self-supervised models. Not relying on any task-specific annotation, we argue that the generated structures are more general, however potentially weaker, due to the lack of explicit supervision. Consequently, we believe that the structures inferred in this chapter present a promising addition to be combined with more task-specific, distantly supervised approaches presented in Part I.

Considering language modelling as the auxiliary task used in this chapter, we confirm our central hypothesis presented in Chapter 1 by successfully extracting valid discourse structures from pre-trained language models. In
regard to the central research questions posed in this thesis, this chapter extends our answer to \((RQ1)\) by proposing a self-supervised methodology to infer discourse structures from popular pre-trained language models. Further, we show the effectiveness of our extraction method to obtain discourse trees from the PLM self-attention. Exploring the complementary nature of extracted discourse structures to top-performing supervised approaches further confirms the usefulness of our approach.

Despite the strong performance of pre-trained and fine-tuned language models to generate fluent text, one of the key weaknesses shared by all recently proposed PLMs is their sequentially inspired nature, not taking more complex linguistic structures (such as trees or graphs) into account and, in turn, limiting their ability to generate coherent long-form text. With some researchers even arguing that “the language modelling task, because it only uses form as training data, cannot in principle lead to learning of meaning” [13], we believe that a more structurally inspired approach to model language can potentially result in better meaning representations. We explore a novel approach along these lines in Chapter 7, proposing a tree-structured language modelling framework.
Chapter 7

Discourse Inference from Tree-Style AutoEncoders

The work described in this chapter has been published at AAAI 2021 [62]. I have been the lead investigator, responsible for all major areas of concept formation, statement of research questions, data collection, implementation as well as paper composition. Giuseppe Carenini was the supervisory author.

7.1 Motivation

As outlined in Section 2.1, discourse parsing is a key Natural Language Processing (NLP) task for multi-sentential text. The RST discourse theory (and human annotated discourse frameworks at large) thereby have two major drawbacks: (i) Since the theories rely on annotation guidelines rather than data-driven algorithms, the human factor plays a substantial role in generating treebanks, posing a difficult task on linguistic experts. In this chapter, we are eliminating the human component from the annotation process by employing a data-driven approach to generate discourse trees directly from natural language, capturing commonly occurring phenomena in a self-supervised manner. (ii) The annotation process following human-generated guidelines is expensive and tedious, limiting available RST-style discourse corpora in both, the size and number of domains. Using an
automated, data-driven approach as described in this chapter allows us to crucially expand the size and domain coverage of datasets annotated with RST-style discourse structures.

Looking back at Part I, proposing three approaches to infer discourse from distant supervision, it becomes apparent that one critical limitation of all aforementioned models is the task-specificity, possibly only capturing downstream-task-related information. This potentially compromises the generality of the resulting trees. In order to alleviate this limitation of task-specificity, we propose a new strategy to generate tree structures in a task-agnostic, self-supervised fashion by extending the latent tree induction framework proposed by Choi et al. [21] with an auto-encoding objective. Our system thereby extracts important knowledge from natural text by optimizing both the underlying tree structures and the distributed representations. We believe that the resulting discourse structures effectively aggregate related and commonly appearing patterns in the data by merging coherent text spans into intermediate sub-tree encodings, similar to the intuition presented in Drozdov et al. [31]. However, in contrast to the approach by Drozdov et al. [31], our model makes discrete structural decisions, rather than joining possible sub-trees using a soft attention mechanism. We believe that our discrete tree structures allow the model to more efficiently achieve the autoencoder objective in reconstructing the inputs, directly learning how written language can be aggregated in the wild (comparable to previous work in language modelling [73]). In general, the proposed approach can be applied to any tree-structured objective, such as syntactic parsing, discourse parsing and further problems outside of NLP, like tree-planning [48] and decision-tree generation [188]. Yet, due to the especially difficult discourse annotation process, we initially generate larger and more diverse discourse treebanks.

7.2 Related Work

The related work on autoencoders and their ties to popular NLP models is described in Section 2.4.4. Compared to Socher et al. [144], our approach
proposes three major improvements: (i) Socher et al. [144] make sequential, local decisions on the aggregation of spans to generate a tree structure, rather than optimizing the complete process holistically. (ii) Their model uses a self-supervised objective in the initial step but requires supervision in later stages and (iii) The model has been only applied to syntactic parsing. In contrast, we apply our model to discourse parsing, which arguably introduces further difficulties, as we will discuss in Section 7.4.

Further, Choi et al. [21] showed a promising approach to infer tree structures in a holistic and parallelizable manner, generating task-depended trees solely relying on sentiment-related information. In their model, they make use of the Gumbel-Softmax [66] (also used in similar ways in Corro and Titov [25, 26]), allowing the neural network to make discrete decisions while still being able to use standard approaches like back-propagation and gradient descent to optimize the model. By combining a similar objective to Socher et al. [144] and Chen et al. [20], we utilize the discrete decision process in Choi et al. [21], positioning our work at the intersection of these two lines of research.

The general task of tree inference has been mostly explored on the sentence level. For instance in Choi et al. [21] and Socher et al. [144] as described above, or by applying a reinforcement approach [179] or CKY methodology [106] to syntactic parsing. Our work employs a novel, fully differentiable approach to a similar problem in the area of discourse parsing.

In discourse parsing, multiple attempts to overcome the aforementioned limitation to small-scale human-annotated datasets have been made. However, all previous models (such as Huber and Carenini [59, 61], Liu and Lapata [97], Liu et al. [102]) use downstream tasks to infer discourse structures. While this is a valid strategy, shown to achieve SOTA results on the inter-domain discourse parsing task [61], as well as performance gains on downstream tasks (e.g., Liu and Lapata [97], Liu et al. [102]), those discourse structures are likely task-depended and need to be either combined across multiple downstream tasks or can only be applied in similar domains. Further work has been trying to infer RST-style discourse structures in a linguistically supervised manner [122], showing good performance when heavily exploiting
syntactic markers in combination with general linguistic priors. Yet, the approach appears to be very specific to the data at hand – news articles from the Wall Street Journal – raising questions in regard to overfitting.

In this chapter, we explore a purely self-supervised approach: instead of relying on domain-specific syntactic features, we infer general discourse trees (structure only) by exploiting inherently available information from natural data (not requiring any supervision), making our model similar to approaches in language modelling [73]. More specifically, our novel methodology extends the previously proposed Gumbel-TreeLSTM approach [21] by substituting the original downstream task with an autoencoder-style reconstruction.
Figure 7.1: T-AE (Tree-AutoEncoder) topology for self-supervised tree inference. Inputs and outputs are dense encodings. $\hat{E}_{nc_x}$ represents reconstruction of spans. $\varnothing$ represents the pointer-network, $g \sim G(0,1)$ denotes the Gumbel-softmax (in the forward-pass with an additional straight-through computation, not shown here). Grey/Dashed components represent actions outside the computational path chosen. Red=Model inputs/outputs, blue=TreeLSTM cells, green=Discrete structure selector as in Choi et al. [21], yellow=Hidden sub-tree encodings, orange=Hidden state $z$ of the complete input.
7.3 Self-Supervised Tree-AutoEncoder (T-AE)

We now outline our general tree-autoencoder model. The description is purposely general, as the model is independent of a specific application and we believe can be utilized in manifold scenarios. Generally speaking, our proposed model induces tree structures through compression and reconstruction of raw inputs in a tree autoencoder-style architecture. The model is similar in spirit to the commonly used sequence-to-sequence (seq2seq) architecture [150], which has also been interpreted as a sequential autoencoder [90]. However, our approach generalizes on the seq2seq model, which is essentially a special (left-branching) case of a tree-structured autoencoder. While the sequential structure of a document is naturally given by the order of words, EDUs, and sentences, moving towards more general tree representations adds the additional difficulty of inferring valid tree structures alongside the hidden states. To generate these discrete tree structures during training, in conjunction with the hidden states of the neural network, we make use of the Gumbel-softmax decision framework, allowing us to discretely generate tree-aggregations alongside intermediate sub-tree encodings [47, 66, 105]. As presented in Figure 7.1, the structure of our novel T-AE (Tree-AutoEncoder) model comprises of an encoder, compressing the input into a fixed-size hidden vector (top in 7.1) and a subsequent decoder component, reconstructing the inputs in an autoencoder-style fashion (bottom in 7.1).

7.3.1 Encoder Component

The computational steps performed in our encoder are akin to the approach described in Choi et al. [21], computing a single document encoding through a tree-style aggregation procedure. Our approach generates a hidden state $Enc_{l,r} = [c_p, h_p] = LSTM_{compress}(l, r)$ for every two adjacent input embeddings $l = [c_l, h_l]$ (left) and $r = [c_r, h_r]$ (right) using a binary TreeLSTM cell as proposed by Tai et al. [153].

---

$^1$Equation 7.1 is modified from Choi et al. [21] and Tai et al. [153].
\[
\begin{bmatrix}
  i \\
  f_t \\
  f_r \\
  o \\
  u
\end{bmatrix}
= \begin{bmatrix}
  \sigma \\
  \sigma \\
  \sigma \\
  \sigma \\
  \tanh
\end{bmatrix} \cdot (W \begin{bmatrix}
  h_l \\
  h_r
\end{bmatrix} + b) 
\] (7.1)

\[c_p = f_t \cdot c_l + f_r \cdot c_r + i \cdot u\]

\[h_p = o \cdot \tanh(c_p)\]

With \(W \in \mathbb{R}^{5|\mathcal{H}_p| \times 2|\mathcal{H}_p|}\) and \(b \in \mathbb{R}^{2|\mathcal{H}_p|}\). Based on the \((n - 1)\) sub-tree candidates \(\mathcal{E}_{n_{k_1,r}}\) with \(0 \leq l < (n - 1)\) and \(r = l + 1\) of the given inputs \(I (|I| = n)\), an un-normalized attention computation (or pointer network) \(\varphi = \text{Pointer}(:,\cdot) [157]\) is used to predict which two adjacent units should be merged. Randomly uniform Gumbel noise, obtained from the Gumbel distribution \(G(0,1)\), effectively sampling \(g \sim G(0,1)\) as \(g_i = -\log(-\log(u_i))\) and \(u_i = \text{Uniform}(0,1)\) is added to the un-normalized scores. Subsequently, the scores are normalized across aggregation candidates according to the temperature coefficient \(\tau\) to obtain \(p(l,r)\) (see Equation 7.2).

\[p(l,r) = \frac{\exp[(\varphi(l,r) + g)/\tau]}{\sum_{k=0}^{n-1} \exp[(\varphi(I_k, I_{k+1}) + g)/\tau]}\] (7.2)

In the forward pass, the straight-through (ST) Gumbel-distribution is used to enforce a discrete selection \(p_{st}\), as commonly done using the Gumbel-softmax trick (see Equations 7.3 [21, 25, 26, 66]).

\[p_{st}(l,r) = \begin{cases} 
1, & \text{if } \arg \max_{k=0,...,n-2} p(I_k, I_{k+1}) = l \\
0, & \text{otherwise}
\end{cases}\] (7.3)

Given this one-hot encoding for a set of aggregation candidates, the most appropriate aggregation, as predicted by the pointer component and perturbed with the Gumbel-softmax, is executed. All other inputs with
\[ p_{st} = 0 \] are directly forwarded to the next step and the respective TreeLSTM computations are discarded (grey/dashed boxes in Figure 7.1). In the example shown in Figure 7.1, Enc₁ and Enc₂ are aggregated, while Enc₃ is directly forwarded to the next step without any aggregation computation.

We recursively generate \( n - 1 \) tree candidates using the TreeLSTM cell in conjunction with the pointer-component and the Gumbel-softmax to build a discrete tree along with sub-tree hidden states in bottom-up fashion\(^2\). Once the tree is aggregated, a single hidden-state \( z \) represents the complete input. Given this dense hidden-state (orange in Fig. 7.1), Choi et al. [21] add a multi-layer-perceptron (MLP) to predict the sentence-level sentiment on the Stanford Sentiment Treebank (SST) [145]. As a result, the obtained tree structures in their model are task-dependent, as shown in Williams et al. [164]. With the goal to generate task-independent structures, we replace the task-dependant MLP layer with our autoencoder objective to reconstruct the original inputs.

### 7.3.2 Decoder Component

The decoder component is implemented as an inverse TreeLSTM with two independent LSTM cells, recursively splitting hidden states into two separate encodings, reconstructing the left and right child nodes (see Equation 7.4).\(^2\)

\(^2\)Please note that the computation of the hidden states in the TreeLSTM cell and the tree structure prediction using the pointer-network with Gumbel perturbation are non-overlapping, allowing for independent optimization of either component.
\[
\begin{bmatrix}
  i_l \\
  f_l \\
  o_l \\
  u_l \\
  i_r \\
  f_r \\
  o_r \\
  u_r
\end{bmatrix}
= \begin{bmatrix}
  \sigma \\
  \sigma \\
  \sigma \\
  \tanh \\
  \sigma \\
  \sigma \\
  \sigma \\
  \tanh
\end{bmatrix}
\cdot (W h_p + b)
\]

(7.4)

\[
c_l = f_l \cdot c_p + i_l \cdot u_l
\]
\[
c_r = f_r \cdot c_p + i_r \cdot u_r
\]
\[
h_l = o_l \cdot \tanh(c_l)
\]
\[
h_r = o_r \cdot \tanh(c_r)
\]

With \( W \in \mathbb{R}^{8|h_p| \times |h_p|} \) and \( b \in \mathbb{R}^{|h_p|} \). Guided by the predicted tree structure of the ST Gumbel-softmax, as shown in Figure 7.1 and Equations 7.2 and 7.3, the structural decision process in the reconstruction phase selects the highest scoring node to be further subdivided into a local sub-tree. This reconstruction approach, generating two child node encodings given the parent encoding \( Enc_p \rightarrow [Enc_l, Enc_r] \) is recursively applied top-down until the original number of inputs \( |I| = n \) is reached. Finally, the reconstructed dense encodings \([\hat{Enc}_1, \ldots, \hat{Enc}_n]\) are evaluated against the model input encodings, following the autoencoder objective.

### 7.4 Discourse Tree Generation

The T-AE approach described above has been kept deliberately general. In this section, we outline the application-specific extensions required in order to deal with the inputs, assumptions, and granularity of the discourse parsing task. First, for the task of discourse parsing, the model inputs \( I \) are clause-like EDUs, representing sentence fragments consisting of one or more words. While the input and output encodings for word-level autoencoders
are naturally represented as the respective one-hot vectors of words in the vocabulary, this approach is not directly applicable for discourse parsing. Hence, we encode the EDUs as dense representations and execute the autoencoder objective directly on these embeddings [131]. Second, as discourse parsing considers complete documents, frequently containing a large number of sentences with oftentimes diverse content, we apply a commonly used approach in this area by separating within-sentence and between-sentence sub-trees [72]. In this setup, we apply the model described above for each sentence individually, trying to infer general patterns at sentence level and subsequently using the learned sentence encodings (orange in Figure 7.1) as the starting point of the document-level T-AE. Having two separate models on sentence- and document-level further aligns with previous work in discourse parsing, postulating different feature sets for different levels in the tree-generation process [72, 160].

7.5 Evaluation

7.5.1 Tasks

To fully evaluate the performance of our T-AE method, we conduct experiments on three distinct tasks, focusing on the two learning goals of our model: (i) Evaluating if the model is able to infer valuable and general discourse structures and (ii) Assessing the ability of the model to learn task-independent hidden states, capturing important relationships between instances. The three tasks are:

**Alignment with existing RST-style Discourse Structures:** Proposing a self-supervised approach to generate (discourse) tree structures allows us, in principle, to generate trees in any domain with sufficient raw text for training. However, due to the expensive and tedious annotation of gold-standard discourse trees, only very few datasets in some narrow domains are augmented with full RST-style trees, required to evaluate our generated structures. Despite their limited coverage, we believe that comparing the
discourse structures produced by our newly proposed model on the alignment with human-annotated discourse structures can be insightful.

**Ability to Predict Important Downstream Tasks:** Besides evaluating the overlap with existing, human-annotated discourse trees, we investigate the generality of the T-AE model by evaluating the performance when applied to an important downstream task in NLP – sentiment analysis. We, therefore, use the document-level hidden state of our model (orange in Figure 7.1), trained on the self-supervised autoencoder objective, and add a single feed-forward neural network layer on top, reducing the hidden state of our model to the number of sentiment classes required for the sentiment prediction task. Training this linear combination on top of the model’s document-level encoding gives further insight into the information contained in the hidden state and its alignment with the downstream task of sentiment analysis.

**General Representational Consistency:** In this third task, we further explore the information captured by the document-level hidden state by qualitatively comparing the dense encoding of a random (short) sample document with its most similar/most different documents, giving intuition about the relatedness of similarly encoded documents.

### 7.5.2 Datasets

**The RST-DT Treebank** published by Carlson et al. [18], as described in Section 2.2. In order to obtain a development set, we subdivide the training portion into 308 documents for training and 36 documents for a length-stratified development set. N-ary sub-trees are further converted into a sequence of right-branching constituents.

**The Yelp’13 Dataset** by Tang et al. [154] is a review dataset published as part of the 2013 Yelp Dataset Challenge. The corpus contains predominantly restaurant reviews alongside a 5-point star rating. Frequently used in previous
work, the dataset has been pre-segmented into EDUs by Angelidis and Lapata [4], using the discourse-segmenter proposed in Feng and Hirst [38]. The complete dataset contains 335,018 documents in an 80-10-10 data split, resulting in 268,014 training documents and 33,502 documents each in the development and test sets.

7.5.3 Baselines

For the Alignment with RST-style Discourse Structures, we evaluate three sets of related approaches: For supervised models, we compare against a diverse set of previously proposed, fully supervised discourse parsers, trained and evaluated on the RST-DT dataset. These include the CODRA model by Joty et al. [72], the Two-Stage approach by Wang et al. [160] and the neural topology by Guz et al. [51]. We further compare against our two distantly supervised models proposed in Chapter 3 (MEGA-DTBase and MEGA-DT), using sentiment analysis to inform the generation of discourse structures with distant supervision. Our last set of baselines for this task contains linguistically supervised approaches. We compare our model against fully left- and right-branching trees, as well as hierarchically left- and right-branching tree structures (separated on sentence level), encoding basic rhetorical strategies. Left-branching trees generally reflect a common sequential strategy while right-branching tree structures oftentimes accurately represent documents where the main objective is initially expressed and then further evaluated throughout the document (e.g. news)\(^3\). We further show the recently proposed model by Nishida and Nakayama [122] in our evaluation. In their best model setting, Nishida and Nakayama [122] heavily exploit basic rhetorical strategies of natural language by aggregating a document into right-branching trees on sentence- and paragraph-level and joining paragraphs using left-branching constituents. Starting from this linguistically inspired tree (already achieving remarkable performance on the well-structured news documents), they apply a Viterbi EM algorithm to

\(^3\)Right-branching trees are further artificially favoured in RST-style discourse parsing since most parsing models convert n-ary sub-trees into a sequence of right-branching constituents.
achieve further improvements. Despite the promising results on RST-DT, we believe that such high performance is mostly due to the well-structured nature of news documents and not generally applicable to other domains, the main objective of our presented approach.

Building on the intuition given in Chapters 3 and 9, we further evaluate our model regarding the **Ability to Predict Important Downstream Tasks**. More precisely, we evaluate the sentiment prediction performance of the document-level hidden-state of our model against the HAN model proposed by Yang et al. [177], the LSTM-GRNN approach by Tang et al. [154], as well as a document encoding, build from average random word encodings and the majority class baseline.

### 7.5.4 Hyper-Parameters Settings
We select our hyper-parameters based on the development set performance of the respective datasets. Despite the fact that we are training two self-supervised models (on RST-DT and Yelp’13), we use a single set of hyper-parameters, to be more general. We train all models using the Adam optimizer [80] with the standard learning rate of $10^{-3}$. As mentioned before, we are directly training on dense representations of input EDU embeddings, comparing them to the reconstructed representations of EDUs. This setup makes the Kullback-Leibler Divergence (KLD) or the Mean-Squared-Error (MSE) the natural choice as our loss function. In this work, we employ MSE due to its superior performance observed on the development set. Each EDU in the input document is represented as the average GloVe word-embedding [128] as in Choi et al. [21]. The loss is computed on the softmax of the respective inputs and outputs. We train our model on mini-batches of size 20, due to computational restrictions and apply regularization in form of 20% dropout on the input embeddings, the document-level hidden state and the output embeddings [21]. We clip gradients to a max norm of 2.0 to avoid exploding gradients. Documents are limited to 150 EDUs per document and a maximum of 50 words per EDU [59]. We restrict the vocabulary size

---

4 Trained on an Nvidia GTX 1080 Ti GPU with 11GB of memory.
to the most frequent 50,000 words with an additional minimal frequency requirement of 10. We train the sentence- and document-level model for 40 epochs and select the best-performing generation on the development set. The hidden dimension of our LSTM modules as well as the pointer component is set to 64, due to computational restrictions. To avoid our model to interfere with the input GloVe embeddings, we freeze the word representations. To promote consistency between the encoding and decoding, we tie the decoder tree decisions to the encoder predictions, enabling a more consistent tree embedding in the compression and reconstruction phase. Furthermore, to disentangle the optimization of structures and hidden states, we apply a phased approach, alternating the training of the two components in a conditional back-propagation loop with a single objective in each pass over the data (see footnote 2). This way, the hidden states are recalculated based on the last epoch’s structure prediction and vice-versa. To be able to explore diverse tree candidates in early epochs and further improve them during later epochs, we start with the diversity factor $\tau = 5$ and linearly reduce the parameter to $\tau = 1$ (see Choi et al. [21]) over 3 structure-learning epochs.

7.5.5 Experiments

In this section, we evaluate our novel T-AE model on the three tasks described in Section 7.5.1. Table 7.1 shows the results of the first task, evaluating our model on RST-style discourse structures from the RST-DT treebank. The first sub-table shows three top-performing, completely supervised models, reaching a structure-prediction performance of 86.47% using the neural approach by Guz et al. [51]. In comparison, the second sub-table contains our distantly supervised models, achieving a performance of 77.82% [61]. The third sub-table presents the linguistically supervised models, showing a clear advantage of the right-branching models over left-branching approaches, in line with our intuition given in Section 7.5.3. Furthermore, considering sentence boundaries and generating hierarchical baselines significantly improves the performance, reaching 74.37% with the hierarchical right branching
baseline and 70.58\% on the left-branching structures. The linguistically supervised Viterbi EM approach by Nishida and Nakayama \cite{122} reaches a performance of 84.30\% with their multi-level hierarchical approach. Our newly proposed, self-supervised and purely data-driven approach is shown in the fourth sub-table. In comparison to the aforementioned linguistically supervised models, this set of results makes no assumptions on the underlying data except the sentence/document split. When trained on the raw text of the small-scale RST-DT dataset, our T-AE approach reaches a performance of 69.68\%, slightly below the linguistically supervised hierarchical left-branching model. Even though the self-supervised training corpus is within the same domain as the test dataset, the very limited amount of data seems insufficient for the self-supervised objective. Training our model on the nearly three orders of magnitude larger Yelp'13 dataset, we reach a performance of 71.32\% evaluating the tree structures on RST-DT. This result shows that a larger training dataset, even though containing out-of-domain documents (reviews vs. news), can improve the performance over the within-domain model trained on a small-scale dataset and the hierarchical left-branching model.

To evaluate the ability of our model to capture valid information to represent input documents, we assess the document-level hidden state’s ability to capture useful information for the downstream task of sentiment analysis. The results of this experiment are provided in Table 7.2, showing the accuracy of our models when compared to commonly used approaches. The best system (the HAN model) reaches an accuracy of 66.2\%, while the random baseline reaches 37.30\%, and a simple majority class baseline achieves 35.63\%. Our models based on the T-AE hidden states obtain accuracy scores in-between those results, reaching 40.41\% and 42.69\% when trained on RST-DT and the much larger Yelp’13 respectively. While this performance is still far from the results of completely supervised models, the improvements over the simple baselines suggest the usefulness of our learned document-level encodings.

In our third and last experiment, we aim to further evaluate the quality of the document encodings in a qualitative manner. We, therefore, compare the
<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>88.30</td>
</tr>
</tbody>
</table>

### Supervised

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODRA</td>
<td>83.84</td>
</tr>
<tr>
<td>Two-Stage</td>
<td>86.00</td>
</tr>
<tr>
<td>Neural-SR</td>
<td><strong>86.47</strong></td>
</tr>
</tbody>
</table>

### Distantly Supervised

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEGA-DTBase</td>
<td>76.41</td>
</tr>
<tr>
<td>MEGA-DT</td>
<td><strong>77.82</strong></td>
</tr>
</tbody>
</table>

### Linguistically Supervised

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Branching</td>
<td>53.73</td>
</tr>
<tr>
<td>Right Branching</td>
<td>54.64</td>
</tr>
<tr>
<td>Hier. Left Branching</td>
<td>70.58</td>
</tr>
<tr>
<td>Hier. Right Branching</td>
<td>74.37</td>
</tr>
<tr>
<td>ViterbiEM</td>
<td><strong>84.30</strong></td>
</tr>
</tbody>
</table>

### Self-Supervised

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>OursRST-DT</td>
<td>69.68</td>
</tr>
<tr>
<td>OursYelp’13</td>
<td><strong>71.32</strong></td>
</tr>
</tbody>
</table>

**Table 7.1:** Results of the micro-average precision measure, evaluated on the RST-DT corpus. Subscripts identify training sets. Best model in each subset is **bold**.

hidden state of a random document from the Yelp’13 test set against all data points in the test portion and show the three most similar/most different documents according to the cosine similarity measure in Table 7.3. It can be observed that closely related documents have a similar argumentative structure as the core document (top row in Table 7.3), initially describing a positive aspect and subsequently evaluating negative components. The most different documents tend to have an inverse structure.
<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAN[177]</td>
<td>66.20</td>
</tr>
<tr>
<td>LSTM-GRNN[154]</td>
<td>65.10</td>
</tr>
<tr>
<td>OursYelp’13</td>
<td>42.69</td>
</tr>
<tr>
<td>OursRST-DT</td>
<td>40.41</td>
</tr>
<tr>
<td>Random Encoding</td>
<td>37.30</td>
</tr>
<tr>
<td>Majority Class</td>
<td>35.63</td>
</tr>
</tbody>
</table>

Table 7.2: Five-class sentiment accuracy scores trained and tested on the Yelp’13 dataset, subscripts in model-names indicate dataset for self-supervised training. Best model is **bold**.

7.6 Contributions

In this chapter, we propose a self-supervised and purely data-driven tree-style autoencoder to compress and reconstruct textual data. In our proposed modelling approach, we aggregate complete documents into a single dense document encoding $z$. However, in comparison to most previous work, we do not condition the autoencoder bottleneck ($z$) on sequential inputs (as for example done in seq2seq architectures) but infer a discrete, latent tree structure during the encoding and decoding process (similar to Choi et al. [21]). This way, we follow a more linguistically inspired aggregation approach, following the well-known, hierarchical structure of text, rather than interpreting a document as a flat sequence of words. In our experiments, we show the potential of our T-AE approach for the task of discourse parsing, which severely suffers from training data sparsity, due to the tedious and expensive annotation process of gold-standard (human-annotated) discourse trees. Our self-supervised model outperforms one of the commonly used, linguistically supervised approaches, without making any assumptions on the underlying data, except the sentence/document split. The superior performance compared to the hierarchical left-branching baseline plausibly indicates that our self-supervised structures are valuable when combined with supervised or distantly supervised models. Furthermore, the superior performance of the large out-of-domain model trained on the Yelp’13 dataset.
over the small-scale within-domain model trained on the raw text of RST-DT shows the robustness and scalability of our approach.

In conclusion, our Tree-AE methodology presents a new and more linguistically inspired approach to language modelling. While the performance in this chapter is below the results obtained in Chapter 6 using sequentially inspired, pre-trained language models, our model provides two valuable contributions: (i) With the additional inference of tree structures, our model is more interpretable than purely sequential models, allowing for a better understanding of the result, oftentimes critical for high-impact applications. (ii) Our Tree-AE model preserves sub-tree compositionality through the local TreeLSTM architecture, not relying on outside context. This is an interesting property for many downstream applications (e.g. aspect-based sentiment analysis) and a generally desired feature.

Regarding our central research questions in this thesis, we present a new generation approach to automatically annotate discourse trees from unlabelled data and show a detailed evaluation of our T-AE approach, analyzing: (i) The reasonable alignment of our generated structures with RST-style discourse trees (RQ1) as well as (ii) The ability of the T-AE model to learn valid representations for the downstream task of sentiment analysis (RQ2). By confirming the value of the T-AE learned structures and encodings, we substantiate our central hypothesis, showing that to properly learn the tree-style language modelling task, useful discourse structures are internally employed.

With our novel approach, we take the first step towards more interpretable and compositional autoencoder language models. As a result, despite the relatively low performance shown in our experiments, we believe that this initial line of work has the potential to pave the way for more elaborate architectures built around linguistic priors.
<table>
<thead>
<tr>
<th>Document</th>
<th>Prices were cheap, however food was served well after others who came in and they literally put brown gravy on the Mexican food, staff ignored simple requests. Only reason for 1 star was due to price.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similar-1</td>
<td>This establishment has a good 10$ lunch special with plenty of variety in the bento they offer, service is usually good polite and efficient the only thing that makes me crazy is the crappy usually too loud caned pop music they play.</td>
</tr>
<tr>
<td>Similar-2</td>
<td>The good: Awesome complimentary breakfasts, warm gooey chocolate chip cookie at check in, nice pool and and hot tub at the center, fairly large room with 2 TVs (flat screen) and a huge comfortable bed with down pillows. The bad: Not a bad fitness room but could be larger, it doesn’t have the feel look of a fancy hotel at first. More like a Motel (but the rooms are nice and the restaurant too). The ugly: No free internet</td>
</tr>
<tr>
<td>Similar-3</td>
<td>Just the facts: Great options for healthier eating, unique non-meat sandwich options at lunch (portabello, grilled zucchini, black bean, etc.), decent coffee, cute atmosphere and fun s&amp;p shakers at the table, kind of pricey. I want to go back to try breakfast.</td>
</tr>
<tr>
<td>Different-1</td>
<td>Forgot to mention the prices are great &amp; just had the baklava yummm to die for delicious!</td>
</tr>
<tr>
<td>Different-2</td>
<td>Bit pricey, but it’s always been our favorite place to go for treats.</td>
</tr>
<tr>
<td>Different-3</td>
<td>Decent place, but the drinks are too expensive unless its a buy 1 get 1 night.</td>
</tr>
</tbody>
</table>

**Table 7.3:** Representationally similar/different document-encodings based on the cosine similarity. For more examples of the representational similarity and additional tree structure comparisons see Appendix B.7 and B.8 respectively.
Part III

Discourse Application for Parsing, Downstream Tasks and Linguistic Extensions
Parts I and II present a variety of novel models and methodologies leveraging distantly supervised and self-supervised frameworks. Focusing on the challenging task of robust inter-domain discourse tree generation, we thereby answered research question (RQ1). In Part III, we focus on the potential of our automatically generated discourse trees to improve supervised discourse structures (RQ1), downstream applications (RQ2) and linguistic theories (RQ3). As such, we focus on three important indicators for the usefulness of our previously inferred discourse trees: (i) If the generated structures are general, robust and well-aligned with gold-standard discourse trees, they should be able to support supervised discourse parsers to better predict intra-domain structures. (ii) General and robust distantly supervised discourse structures should be a valuable source of information for real-world downstream tasks, especially when the distant supervision signal is similar to the target domain. (iii) The inferred discourse trees should be well aligned with the underlying discourse theory.

To this end, we present a novel approach using our MEGA-DT “silver-standard” discourse treebank presented in Chapter 3 to pre-train a neural discourse parser, before fine-tuning it with gold-standard discourse trees. This way, we explore if the distantly supervised discourse information is general and robust enough to learn valuable representations in the pre-training step, subsequently guiding the supervised fine-tuning (Chapter 8). Next, we confirm the generality and usefulness of our inferred discourse structures on the downstream task of sentiment analysis in Chapter 9. Finally, to close Part III, we show initial, yet promising results on using internal states of distantly supervised methodologies to replace crude human simplifications in the RST discourse theory (namely the binary nuclearity attribute) with more expressive, real-valued importance scores in Chapter 10.
Chapter 8

Discourse Parsing with Silver-Standard Pre-Training

This chapter has been previously published at COLING 2020 [51]. I have been the second author, with Grigorii Guz as the lead investigator. Throughout the project, I contributed to the process of defining project goals and key research questions, supported the implementation, and took stakes in the design of the experiments. I further contributed to the paper itself. Giuseppe Carenini was the supervisory author of the project.

8.1 Motivation

As throughout this thesis, this chapter focuses on discourse representations for the English language based on the Rhetorical Structure Theory (RST) proposed by Mann and Thompson [107].

In the literature, previous research has shown that the use of RST-style discourse parsing as a system component can enhance important downstream tasks and are complementary to learned contextual embeddings in the popular BERT model for tasks where linguistic information on complete documents is critical, such as argumentation analysis [19]. In line with the main objective of this work, to alleviate the lack of large-scale annotated datasets in the area

\footnote{For more information on the RST discourse theory, please refer to Section 2.1}
of discourse parsing, we propose a novel approach combining our large-scale “silver-standard” MEGA-DT discourse treebank (presented in Chapter 3) with a neural discourse parsing strategy. This way, we use the ≈ 250,000 discourse annotated documents in the MEGA-DT treebank to revisit the task of neural discourse parsing, which has been previously attempted by Yu et al. [180] and others with rather limited success. We believe that one of the main reasons why previous neural models could not yet consistently outperform more traditional approaches, heavily relying on feature engineering [160], is the lack of generalization when solely relying on the small-scale RST-DT dataset, containing only a minuscule 385 discourse annotated documents. As a result, we believe that using a more advanced neural discourse parser in combination with a nearly three orders of magnitude larger dataset (compared to RST-DT [18] and Instructional-DT [149]) can lead to significant performance gains.

Admittedly, even though MEGA-DT contains a large number of complete discourse trees to train on, the structures have been automatically annotated, potentially introducing noise and biases, which could negatively influence the discourse parsing performance. To explore this potential risk and confirm the robustness and generality of the MEGA-DT corpus, we use the “silver-standard” discourse dataset to pre-train a neural discourse parser and subsequently fine-tune the model on the small-scale RST-DT dataset as well as further human-annotated corpora. This way, we evaluate if general discourse structures are learned from the large-scale MEGA-DT treebank, enhancing the performance compared to training the same neural discourse parser exclusively on human-annotated datasets. With the results shown in this chapter suggesting that our proposed discourse parser effectively encodes discourse, we further confirm the status of MEGA-DT (and to some extent distantly supervised approaches in general) as a valuable resource to enhance neural discourse parsers by applying “silver-standard” discourse trees for general and robust pre-training. Our contributions in this chapter are

1. We propose a novel neural discourse parsing architecture which combines multiple lines of previous work in a single framework.

2. We combine the large-scale “silver-standard” MEGA-DT treebank
with small, domain-specific gold-standard treebanks in a pre-training, fine-tuning framework.

3. We show significant performance improvements of our model over previous state-of-the-art approaches.

8.2 Related Work

The field of discourse parsing has been mainly dominated by traditional machine learning models, frequently outperforming initial attempts to apply deep learning approaches to the task. Independent of the specific approach used, three general methodologies have been followed to learn discourse trees from small datasets, such as RST-DT [18] or Instructional-DT [149]: (i) Top-down discourse parsers, (ii) bottom-up parsers and (iii) the more locally inspired linear shift-reduce framework, adopted from previous work in syntactic parsing. In this chapter, we follow (iii), the bottom-up shift-reduce strategy\(^2\).

Besides the active research area of discourse parsing, the second line of related work generates large-scale discourse treebanks through automated annotations from downstream tasks, such as sentiment analysis [61] (see Chapter 3), text classification [97], summarization [102, 171] (see Chapter 5) and fake news detection [75]. With our work shown in Chapter 3 reaching the best inter-domain performance compared to RST-DT and the Instructional-DT, we believe that the treebank does not only learn sentiment-related information but can also be used to infer general discourse structures on a large scale.

8.3 Our Methodology

We follow the well-established bottom-up shift-reduce aggregation principle, as previously shown effective for traditional discourse parsers [68, 160] and neural approaches [16, 180]. We first introduce the general principle of shift-

\(^2\)For additional details on the different tree-aggregation methods (particularly (i) and (ii)), please refer to Section 2.3.1.
reduce parsing and define the necessary data structures and actions available to our system in Section 8.3.1. For a small scale example of the shift-reduce tree generation approach, we refer readers back to Figure 2.3 in Chapter 2. Based on the general description, we subsequently show our detailed architecture to execute a single step in the linear-time model (Section 8.3.2) and describe the training procedure (Section 8.3.3).

8.3.1 The General Shift-Reduce Architecture

The transition-based shift-reduce parsing architecture traditionally consists of two data structures (a queue and a stack), interacting through a set of possible actions (shift and reduce), as illustrated in Figure 8.1 and further described below.

**The Queue** initially contains the EDUs of a complete document in the natural, sequential order, obtained from manual annotation or discourse segmenters (e.g. Feng and Hirst [39], Li et al. [91]). Depending on the action performed by the parser, the top element on the queue either remains on the queue or is moved to the top of the stack. At the end of the parsing process, the queue must be empty.

**The Stack** contains all previously processed parts of the document (also in natural order). At the beginning of the shift-reduce procedure, the stack is empty and is subsequently filled and aggregated according to the system’s actions. After the parsing process is completed, the stack contains the complete, aggregated document as a single discourse tree.

**The Shift operation** delays the aggregation of sub-trees on the stack by moving the top input node (EDU) from the queue to the top of the stack. To avoid invalid shift-reduce actions, hard constraints are commonly added to ensure that shift operations are only executed when the queue still contains unprocessed nodes.
Figure 8.1: (a) Example Shift action – Top element of the queue gets moved to the top of the stack. (b) Example Reduce action – Top 2 elements of a stack are assembled into a sub-tree. (c) Example of input to our classifier, consisting of the RoBERTa string-encoding and structural features. Note that since EDUs 1 and 2 form a sub-tree, their spans are concatenated.
The Reduce-X operation is used to aggregate the top two partial trees \((S_1, S_2)\) on the stack into a single, joint representation \((S_{1,2})\). For complete RST-style discourse trees, each reduce action further defines a nuclearity assignment \(X_N\) and a relation \(X_R\) to the sub-tree covering \(S_{1,2}\) defined as:

\[
\begin{align*}
X_N &\in \{\text{NN}, \text{NS}, \text{SN}\} \\
X_R &\in \{\text{Elaboration}, \text{Contrast}, \ldots\}
\end{align*}
\]

In this chapter, we limit the scope of the reduce action to solely predict the nuclearity assignment \(X_N\), as the MEGA-DT treebank currently only provides partial discourse trees, not incorporating the relation assignment.

While the specific implementation of the stack and queue components are mostly fixed, the selection of shift- and reduce-actions can be realized with rule-based approaches [110], Support Vector Machines (SVMs) [68] or neural classifiers [180].

8.3.2 The Shift-Reduce Action Classifier

To realize the shift and reduce actions, we take advantage of the recent success of BERT-inspired models, specifically, employing the distilled version [141] of the large-scale RoBERTa [101] language model as the base for our action classifier. At each step in the shift-reduce framework, we predict the next action by encoding the top two elements of the stack \((S_1\) and \(S_2\)) and the top element of the queue \((Q_1)\) using the RoBERTa encoding combined with structural features.

**RoBERTa-based features:** To align with the input format requirements of the RoBERTa model, we encode a joint representation of the three components under consideration (namely \(S_1\), \(S_2\) and \(Q_1\)) into a single string

\[s = [CLS]\|S_2\| [SEP]\|S_1\| [SEP]\|Q_1\| [SEP]\]

with \(\|\) denoting the concatenation of the respective raw text and \([CLS]\) and
representing the sequence-classification and end-of-sequence tokens$^3$.

Since the input sequence length of the RoBERTa language model is by default bound by 512 tokens and the elements on the stack \((S_1, S_2)\) represent increasingly large sub-trees (and therefore text-spans), we restrict the length of \(S_1\) and \(S_2\) to a maximum of 240 words each. Specifically, if one of the constituents exceeds 240 tokens, the concatenation of the 120 leading and trailing words is chosen as the span’s representation, following Prasad et al. \cite{130}. The top element on the queue \((Q_1)\), only containing a single EDU, is truncated to the leftover capacity of 28 tokens. The RoBERTa embedding \(c\) is computed as follows:

\[
v = \text{RoBERTa}(s) \\
c = v[0] \in \mathbb{R}^{768}
\]

with \(c\) encoding the complete span through the [CLS]-token.

**Structural features:** Previously successful approaches to traditional discourse parsing \cite{68,72} have shown that the structural organization of a document into sentences and paragraphs plays a crucial role when predicting discourse, with Joty et al. \cite{72} giving strong intuition for their usefulness by showing that less than 5% of discourse sub-trees violate sentence boundaries in the RST-DT corpus. To explicitly model these structural constraints, we use the same organizational features used in Wang et al. \cite{160} to determine where a span is positioned within the document as well as relative to adjacent constituents. More specifically, for all three spans \(S_1\), \(S_2\) and \(Q_1\), we extract two sets of features: (i) Whether the span is at the beginning/end of a sentence/paragraph/document and (ii) For each pair of adjacent spans \(((S_2, S_1), (S_1, Q_1))\) we compute the features indicating whether the pair is within a single sentence/paragraph. For the three constituents under consideration, this results in an ordered sequence \(O\) of 28 values. Adding a distinct embedding layer with 10 neurons \(u_i = \text{emb}(o_i)\) on top of each value \(o_i \in O\)

\(^3\)Please note that \(S_1\), \(S_2\) and \(Q_1\) are not the typically used dense representations of sub-trees or EDUs, but solely contain the textual representation of a sub-tree or EDU as a flat sequence of words.
results in a concatenated dense representation \( u = u_1, ..., u_{28} \in \mathbb{R}^{280} \) for the structural features. Whenever a feature cannot be computed (for example when the queue is empty or the stack contains a single element), we use a zero-vector.

**Action classification:** To predict the next shift-reduce action during the tree-generation process, the features extracted from the RoBERTa model are concatenated with the structural representations (see Figure 8.1(c)) and fed into a two-layer MLP with an intermediate GeLU [54] activation and a final softmax layer (see Equation 8.1, 8.2).

\[
l = \text{MLP}(c \parallel u) \quad (8.1) \\
p = \text{SoftMax}(l) \quad (8.2)
\]

### 8.3.3 The Neural Shift-Reduce Training Procedure

During training, we minimize the weighted cross-entropy loss in every parsing step individually, allowing for more fine-grained optimization and more parallelizable training.

The training loss is computed between the prediction \( l \) (see Equation 8.1) and the respective gold-label \( y \in \{ \text{Shift}, \text{Reduce}_N, \text{Reduce}_S, \text{Reduce}_{SN} \} \). Since there is only a single shift but three reduce actions, we weight the four output classes by factors \([\frac{3}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}]\) to equally penalize an incorrect shift/reduce action. At test time, the document tree structure is constructed greedily by selecting the action with the highest probability (see Equation 8.2) at each parsing step.

### 8.4 Evaluation

In this section, we introduce the three datasets we used to train and evaluate our model against strong baselines (Section 8.4.1). Our model hyperparameters and the respective search spaces on the development set are
presented in Section 8.4.2, followed by the baseline models in Section 8.4.3. We then introduce the metrics used in this chapter (Section 8.4.4), discuss insights gathered in preliminary evaluations (Section 8.4.5) and finally present results aggregated in Section 8.4.6.

8.4.1 Datasets

We train and evaluate our model on the RST-DT and Instructional Dataset (short: Instr-DT), described in Section 2.2. For the pre-training stage, we use MEGA-DT, the first successfully generated “silver-standard” discourse treebank obtained by applying distant supervision on the large-scale Yelp’13 sentiment dataset [154]. As described in Chapter 3, the treebank contains \( \approx 250,000 \) documents with full RST-style discourse trees encompassing structure and nuclearity attributes. The MEGA-DT corpus has been shown to achieve superior performance when compared to human-annotated datasets (including RST-DT) on the discourse domain transfer task. Due to computational limitations, we pre-train our model on a 52.5k subset of MEGA-DT, with 50k trees used for training and 2.5k data points left for validation.

8.4.2 Hyper-parameters and Training Setup

The hyper-parameters in our model are heavily influenced by previous findings. For the RoBERTa model [101], we use the distilled version proposed in Sanh et al. [141] with 6 layers containing 12 attention heads and a hidden size of 768, as implemented by Wolf et al. [168]. The structural features used as inputs for the classification module are encoded as 10-dimensional embeddings for each of the 28 organizational features. During training, we use the AdamW optimizer [103] with a learning rate of \( 1e^{-3} \) and a weight decay value of \( 1e^{-2} \) for both pre-training and fine-tuning. We further apply gradient norm clipping at 0.2 [126]. The learning rate was scheduled as in Vaswani et al. [156], using 4,000 warm-up steps. Due to the variable size trees in the training data, we aggregate documents with an identical number of EDUs into batches of size 20 during pre-training and 5 for fine-tuning. All model configurations use early stopping if the performance of neither structure...
nor nuclearity improves over 3 consecutive epochs on the development set.
Our models are trained using PyTorch [127] on a GTX 1080 Ti GPU with
11GB of memory. Our code and model checkpoints are publicly available at
https://github.com/grig-guz/rst-large-scale

8.4.3 Baselines
To evaluate the performance of our model in the context of RST-style
discourse parsing, we compare it against a variety of competitive baselines:
The DPLP parser [68], a traditional discourse parser utilizing an SVM-
classifier within the shift-reduce framework, based on linear projections of
lexical features.
The CODRA model [72], using an optimal CKY-based chart parser in
combination with Dynamic Conditional Random Fields (CRF), is separated
on sentence level.
The gCRF model [39], follows the same approach as CODRA, however, uses
a greedy strategy.
The Two-Stage parser, proposed by Wang et al. [160], presents a top-
performing system on the RST-DT structure prediction using two separate
linear SVM classifiers. We use the public codebase provided by Wang et al.
[160] and remove the relation classification module for our experiments.
Transition-Syntax: A neural shift-reduce parser utilizing LSTMs to gener-
ate EDU embeddings, additionally applying a neural dependency parser for
extracting syntactic features [180].
Cross-Lingual, a neural shift-reduce approach by Braud et al. [16], utilizing
multilingual RST treebanks.
The Top-Down-Generative parser by Mabona et al. [104], a recent top-
down transition-based neural generative parser employing Tree-LSTMs to
encode sub-trees on the stack.

4https://github.com/yizhongw/StageDP/
### Table 8.1: Micro-averaged F1-scores for structure and nuclearity prediction using the original Parseval measure as proposed in Morey et al. [115], evaluated on the RST-DT and Instr-DT corpora. Best performance per column is **bold**. Subscripts on results indicate standard deviation, – non-published values.

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure RST-DT</th>
<th>Structure Instr-DT</th>
<th>Nuclearity RST-DT</th>
<th>Nuclearity Instr-DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPLP [68]</td>
<td>64.10</td>
<td>–</td>
<td>54.20</td>
<td>–</td>
</tr>
<tr>
<td>gCRF [39]</td>
<td>68.60</td>
<td>–</td>
<td>55.90</td>
<td>–</td>
</tr>
<tr>
<td>CODRA [72]</td>
<td>65.10</td>
<td>–</td>
<td>55.50</td>
<td>–</td>
</tr>
<tr>
<td>Cross-Lingual [16]</td>
<td>62.70</td>
<td>–</td>
<td>54.50</td>
<td>–</td>
</tr>
<tr>
<td>Two-Stage [160]</td>
<td>70.97</td>
<td>58.86</td>
<td>57.97</td>
<td>40.00</td>
</tr>
<tr>
<td>Top-Down-Generative [104]</td>
<td>67.10</td>
<td>–</td>
<td>57.40</td>
<td>–</td>
</tr>
<tr>
<td><strong>Our Parser</strong></td>
<td>±0.27 72.43</td>
<td>±1.12 64.55</td>
<td>±0.56 61.38</td>
<td>±1.37 44.41</td>
</tr>
<tr>
<td>Our Parser + Pretraining</td>
<td>±0.95 72.94</td>
<td>±0.61 65.41</td>
<td>±0.90 61.86</td>
<td>±1.11 46.59</td>
</tr>
<tr>
<td>Our Parser (- RoBERTa)</td>
<td>±0.18 59.89</td>
<td>±0.30 54.68</td>
<td>±0.55 33.28</td>
<td>±1.83 28.36</td>
</tr>
<tr>
<td>Our Parser (- Features)</td>
<td>±0.72 70.61</td>
<td>±1.74 63.32</td>
<td>±0.65 58.37</td>
<td>±1.83 44.41</td>
</tr>
<tr>
<td>Our Parser (- LM Pretraining)</td>
<td>±0.42 65.78</td>
<td>±2.97 53.50</td>
<td>±0.58 48.93</td>
<td>±2.69 32.82</td>
</tr>
<tr>
<td><strong>Human</strong> [115]</td>
<td>78.7</td>
<td>–</td>
<td>66.8</td>
<td>–</td>
</tr>
</tbody>
</table>

8.4.4 Metrics

As suggested in a recent literature analysis by Morey et al. [115], we use the original Parseval measure to compare micro-average F1 scores. To allow comparisons with previous work, we also report the results with respect to the RST Parseval score, still more commonly used in the recent literature.

8.4.5 Preliminary Evaluation

In our preliminary experiments, we evaluate a set of modelling decisions on the held-out development set, influencing the design of our final model. We obtained three useful insights during this phase: (i) Adding padding to the classifier inputs of the RoBERTa model enhances the performance of the component. We believe this is likely to be the case because RoBERTa internally uses absolute positional embeddings. (ii) Following the intuition that an incorrect shift- and reduce-action should be penalized similarly (independent of the nuclearity label), we found that weighting the loss
function boosts the model’s performance. (iii) In preliminary experiments with different ways for summarize RoBERTa outputs (see Figure 8.1 (c)) into a single vector, we found that simple and attention-based averaging of the outputs produces slightly worse results compared to the [CLS] token.

8.4.6 Main Results

The final results of our evaluation are presented in Tables 8.1 and 8.2. The two tables contain slightly different subsets of competitive discourse parsers from previous work, depending on the metric on which the original authors evaluate their models. The reported scores were either taken from the original paper or the literature survey by Morey et al. [115]5.

The leftmost column in the first sub-table contains the previously proposed models along with two versions of our new discourse parser, with and without pre-training. The center column contains the evaluation results

\begin{table}[h]
\centering
\begin{tabular}{|l|cc|cc|}
\hline
 & \multicolumn{2}{c|}{Structure} & \multicolumn{2}{c|}{Nuclearity} \\
 & RST-DT & Instr-DT & RST-DT & Instr-DT \\
\hline
DPLP\cite{68} & 82.00 & - & 68.20 & - \\
gCRF\cite{39} & 84.30 & - & 69.40 & - \\
CODRA\cite{72} & 82.60 & \textbf{82.88} & 68.30 & 64.13 \\
Cross-Lingual \cite{16} & 81.30 & - & 68.10 & - \\
Transition-Syntax\cite{180} & 85.50 & - & 73.10 & - \\
Two-Stage\cite{160} & 85.98 & 79.43 & 72.40 & 62.39 \\
Our Parser & $\pm 0.13$ & 86.22 & $\pm 0.56$ & 82.27 \\
 & $\pm 0.48$ & \textbf{86.47} & $\pm 0.70$ & \textbf{73.53} \\
Our Parser + Pretraining & $\pm 0.05$ \hspace{0.5cm} 79.95 & $\pm 0.15$ \hspace{0.5cm} 77.34 & $\pm 0.27$ \hspace{0.5cm} 74.87 & $\pm 1.11$ \hspace{0.5cm} 65.82 \\
 & $\pm 0.30$ \hspace{0.5cm} 85.30 & $\pm 0.87$ \hspace{0.5cm} 81.66 & $\pm 0.46$ \hspace{0.5cm} 71.04 & $\pm 0.56$ \hspace{0.5cm} 65.98 \\
Our Parser (- RoBERTa) & $\pm 0.21$ \hspace{0.5cm} 82.89 & $\pm 1.49$ \hspace{0.5cm} 76.75 & $\pm 0.24$ \hspace{0.5cm} 63.22 & $\pm 0.85$ \hspace{0.5cm} 59.18 \\
Our Parser (- Features) & $\pm 0.36$ \hspace{0.5cm} 85.30 & $\pm 1.49$ \hspace{0.5cm} 76.75 & $\pm 0.24$ \hspace{0.5cm} 63.22 & $\pm 0.85$ \hspace{0.5cm} 59.18 \\
Our Parser (- LM Pretraining) & $\pm 0.36$ \hspace{0.5cm} 85.30 & $\pm 1.49$ \hspace{0.5cm} 76.75 & $\pm 0.24$ \hspace{0.5cm} 63.22 & $\pm 0.85$ \hspace{0.5cm} 59.18 \\
Human\cite{115} & 88.30 & - & 77.30 & - \\
\hline
\end{tabular}
\caption{Micro-averaged F1-scores for structure and nuclearity prediction using RST Parseval, evaluated on the RST-DT and Instr-DT corpora. Best performance per column is \textbf{bold}. Subscripts on results indicate standard deviation, – non-published values.}
\end{table}

\footnote{Even though the authors of the Two-Stage parser only report RST-Parseval scores on RST-DT, we also evaluate their approach on Instr-DT and with respect to the original Parseval metric.}
on the structure prediction task for the two test datasets (RST-DT and Instr-DT). The right-most column shows the performance for each of the models on the nuclearity prediction task, again subdivided into the two evaluation datasets. The second sub-table contains the results for various ablations of our model and the bottom sub-table shows human results on the tasks. For all of our models, we report the average performance as well as the standard deviation on each metric over five independent runs. We compare the two versions of our neural discourse parser against the best-performing, previously proposed model for each of the four prediction tasks, meaning we compare the performance of our model, for example on the RST-DT dataset, against the current SOTA model on RST-DT structure-prediction by Wang et al. [160]. For the nuclearity measure, we compare against Yu et al. [180]. The best-performing baseline on the Instr-DT dataset is the CODRA model [72]°

When examining our final evaluation shown in Tables 8.1 and 8.2, it becomes clear that our newly proposed neural discourse parser reaches the highest performance on all measures except the structure prediction on the Instr-DT dataset. We observe that our model strongly outperforms the SOTA approach on the RST-DT structure prediction by Wang et al. [160]. Furthermore, pre-training on the MEGA-DT treebank leads to further improvement with respect to the mean scores over independent runs.

On the Instr-DT dataset, our parser achieves a result similar to the model of Joty et al. [72] on the structure prediction task and substantially outperforms the SOTA baseline on the nuclearity measure when pre-training is applied. Nonetheless, a particularly important result is that our system produces consistently strong performance across multiple domains, which neither of the top-performing traditional systems [72, 160] manages to demonstrate. This serves as an indication that employing large-scale language models alleviates the need for extensive manual feature engineering employed by these systems for RST discourse parsing.

°Please note that the evaluation procedure on Instr-DT for CODRA [72] uses different dataset splits. For a more direct comparison, please see our Instr-DT results against the Two-Stage parser [160], which utilizes the same split as ours.
In addition, we perform an ablation study of our system to analyze the importance of each individual component of our parser. The first row in the second sub-tables illustrates the results when only organizational features are used, while the second row shows the impact of removing the features and only using RoBERTa for the action classification. Finally, the third row contains the performance of our system with organizational features and a randomly initialized RoBERTa model component.

We observe that removing the organizational features results in a noticeable drop in performance, implying the importance of encoding the document structure explicitly. Unsurprisingly, removing the RoBERTa feature extractor leads to a large performance drop, far below the competitive baselines. Finally, we demonstrate the importance of LM pre-training in the last row of the ablation sub-table. While this system performs on par with traditional systems in respect to structure prediction, most likely because of the organizational features, it demonstrates inferior performance on the nuclearity prediction task, which reasonably requires knowledge of more high-level concepts, such as sentence coordination, subordination, and, in more advanced cases, depends on the author’s communicative goal. The overall difficulty of this task is reflected in the relatively low human evaluation scores shown in the last row. Our results can be summarized as follows: (i) Our proposed approach achieves state-of-the-art performance on both the RST-DT and the Instr-DT datasets. (ii) Applying large-scale language models leads to stronger results and higher domain adaptivity in RST discourse parsing. (iii) Pre-training the discourse parser on the large-scale “silver-standard” MEGA-DT treebank enhances the performance and supports the ability of the neural parser to generalize across multiple datasets and domains.

8.5 Contributions

In this chapter, we propose a rather simple, yet highly effective neural shift-reduce discourse parser, utilizing the RoBERTa language model in combination with structural features. With the neural architecture allowing for the common pre-training/fine-tuning paradigm, we make use of our large-
scale MEGA-DT discourse datasets for pre-training. Using this previously unseen number of “silver-standard” discourse trees, we reach state-of-the-art (SOTA) performance on the RST-DT and Instr-DT treebanks.

In conclusion, we make two distinct contributions in this chapter: (i) We propose and evaluate a novel neural discourse parsing architecture, combining multiple lines of previous work in a single framework. (ii) We confirm the usefulness of distantly supervised and automatically generated discourse structures from sentiment analysis for the task of model pre-training. With this result, we further strengthen our previous contribution in Chapter 3, supporting the value of distantly supervised discourse structures by showing their positive impact on supervised discourse parsers.

Connecting this chapter to the central research questions, we further answer (RQ1) on how well the automatically generated discourse annotations can predict discourse. While this question has already been partially answered in Parts I and II, we show further evidence for the value of distantly supervised discourse structures (specifically from the auxiliary task of sentiment analysis) by showing synergistic effects of the MEGA-DT discourse treebank with gold-standard trees in the pre-training/fine-tuning framework. Similarly, we further solidify our central hypothesis on the existence of discourse-related information in auxiliary tasks, specifically for sentiment analysis.

To complement the evidence shown in this chapter, we next evaluate our inferred trees on downstream tasks (Chapter 9) and regarding their ability to extend linguistic theories (Chapter 10).
Chapter 9

Sentiment Prediction Using Silver-Standard Discourse Annotations

This chapter has been published at COLING 2020 [60]. I was the lead investigator, responsible for all major areas of concept formation, statement of research questions, data collection, implementation as well as the paper composition of the project. Giuseppe Carenini was the supervisory author, involved throughout the project in concept formation, discussions, and paper composition.

9.1 Motivation

Predicting whether a given word, sentence or document expresses a positive, neutral or negative sentiment is a fundamental task in Natural Language Processing (NLP). For instance, a survey of text mining papers from 1992-2017 found that out of 4,346 papers, 467 had a sentiment analysis component [96]. While early “bag-of-word” sentiment prediction models [152] and their extensions [166] already show promising results on the task, they all share one inherent limitation: Due to the absence of temporal information, they are not able to fully capture the semantic and pragmatic structure (and therefore the
Figure 9.1: Sentiment annotated discourse tree example for non-trivial document containing 13 clauses with positive and negative constituents, taken from MEGA-DT [61]. The gold-label sentiment is negative. Dashed lines indicate supplementary information, solid lines indicate primary importance. Full text: [I've been a member for a month now.]_1, [and I guess]_2, [I'm able to get my workout done.]_3, [I do find myself annoyed]_4, [how cramped it is at the weights.]_5, [The equipment is older.]_6, [but it suffices.]_7, [I worked out at another studio on 3rd]_8, [and it was amazing!]_9, [It was so clean, nice, and new -]_10, [TV’s on every cardio machine.]_11, [When i came back to this location.]_12, [I felt bad.]_13.

sentiment) of long texts, where different meanings oftentimes directly emerge from the word order, underlying syntax and discourse structure. Recent models for sentiment analysis address this limitation by leveraging sequential paradigms [3, 30, 79, 153], hierarchical information [177], syntactic structures [145] or discourse information of multi-sentential text [69].

In this chapter, we follow the last line of the aforementioned research, by developing a framework to exploit automatically generated, large-scale, domain-related discourse structures for sentiment prediction. Arguably, such framework can be especially beneficial for long documents that examine positive and negative aspects of a subject matter in complex rhetorical structures, as shown in Figure 9.1

More specifically, to incorporate RST-style discourse structures into the task of sentiment analysis, we employ a hybrid approach inspired by Bowman
et al. [15] and Choi et al. [21], integrating a TreeLSTM [153] with the well-established Hierarchical Attention Network (HAN) model [177]. We further adopt a non-competitive tree attention mechanism [69], previously shown to be more appropriate in this context.

Aiming to enhance the task of sentiment analysis by using discourse, it seems intuitive to employ domain-related discourse structures. Therefore, instead of using the standard RST-DT discourse treebank in the news domain [18], we decide to utilize our MEGA-DT discourse treebank, automatically generated from sentiment annotations [59, 61] (see Chapter 3). This way, our framework goes from sentiment to sentiment, in the sense that the discourse structures used to improve the sentiment predictions are generated through distant supervision from sentiment itself. Our hypothesis is that a parser trained on a large “silver-standard” discourse treebank, automatically generated from sentiment, will generate more useful discourse trees for sentiment prediction than one trained on a small and generic treebank, even if such treebank is human-annotated for RST discourse structures.

In a series of experiments, we show that while our novel approach to discourse-based sentiment prediction is statistically equivalent to the performance of sequential models, it delivers substantial performance gains for long documents, where discourse plays a crucial role. Furthermore, our experiments indicate that the performance of discourse-based sentiment prediction is significantly improved when using discourse trees generated by distant supervision on sentiment, compared to traditional, human-annotated RST discourse corpora. Using an ensemble method, we further improve the performance and, even if only by a small margin, significantly outperform individual models.

9.2 Related Work
This chapter is located at the intersection of recent approaches to discourse parsing and sentiment analysis, most influenced by the following research:

---

1 We did not apply tree transformers to the task, as in spite of recent proposals [121, 142], no standard method has been widely agreed upon yet and results are still rather preliminary.
RST-style discourse parsing is a valuable upstream task for many downstream models, including sentiment analysis. Empirically, the discourse parser by Wang et al. [160] shows that a horizontal separation of the task with the shift-reduce parsing framework achieves competitive performance. In this chapter, we demonstrate the potential of this parser trained on our large-scale sentiment-dependent treebank (MEGA-DT, see Chapter 3) to generate discourse trees for sentiment prediction, enhancing the sentiment analysis performance on long and diverse documents.

Neural sentiment analysis is a common sub-task in many real-world systems, as motivated in Section 2.4.1. Here, our goal is to investigate the influence of discourse information on the task of sentiment analysis. We, therefore, decided to build our framework on the HAN model [177], which is the most established, yet recent approach in the field, previously re-implemented and tested in many studies. We inject discourse information using TreeLSTMs [153], which are well-established to encode tree-style information compared to tree transformers, for which architectural variants and results are still preliminary (e.g. Shiv and Quirk [142], Nguyen et al. [121]).

Combining discourse parsing and sentiment analysis has been previously explored in multiple lines of work [14, 58, 69, 119]. Architecture-wise, the most closely related approach to our new model has been proposed by Ji and Smith [69]. In their evaluation, the authors show slight improvements compared to the sequential HAN model. These initial positive results are a key motivation for our work, in which we aim to further improve the performance, especially on long documents, by not only training the discourse parser on a larger and more appropriate treebank (i.e. MEGA-DT) but also by improving the sentiment prediction, replacing recursive neural networks with superior TreeLSTMs, tightly integrated with HAN.
9.3 The Sentiment-to-Sentiment Framework

Our sentiment-to-sentiment framework involves three phases: A phase of discourse augmentation (Figure 9.2 (a)), in which we follow our previous approach described in Chapter 3. For each document in a corpus containing document-level sentiment annotation, we generate corresponding, task-dependent discourse trees. Then, this discourse augmented sentiment treebank is used to train a discourse parser. In the second phase (Figure 9.2 (b)), the trained discourse parser is applied to the original corpus, using the predicted trees to train our new discourse-based sentiment predictor. Finally, in the third phase (Figure 9.2 (c)), the trained framework is applied to any new document. First, the trained discourse parser generates the discourse tree for the document. Subsequently, this tree (along with the document itself) is fed to our sentiment predictor, which returns the most likely sentiment. In essence, we go from sentiment annotations to sentiment predictions through discourse augmentation.

Regarding the first phase, we briefly describe the discourse augmentation step, previously presented in full detail in Chapter 3, in Section 9.3.1. For phase two, we focus on our novel sentiment predictor in Section 9.3.2. The inference phase is straightforward and will be limited to the description in

---

For more information on the RST discourse theory, downstream tasks and top-performing discourse parsers, please refer to Chapter 2.
9.3.1 Sentiment Inspired Discourse Trees

The approach to generate “silver-standard” discourse trees (incorporating structure and nuclearity) from distant sentiment supervision \([59, 61]\) comprises two major components. First, documents are annotated for sentiment and importance at the EDU level using a neural Multiple Instance Learning (MIL) method \([4]\), solely utilizing document-level supervision signals given in the original corpus. In particular, MIL infers a sentiment polarity label \(p_x\) within the interval of \([-1, 1]\) for each EDU \(x\), depending on the distribution of words/EDUs within and between documents. Using the neural model by Angelidis and Lapata \([4]\), an additional attention mechanism is internally used to weigh the importance of EDUs for the overall document sentiment. The attention weight \(a_x\) in the interval \([0, 1]\) of EDU \(x\) is also extracted from the model and subsequently used as an importance score when aggregating sub-trees. Next, the tuples \((p_x, a_x)\) are combined in a binary, bottom-up approach using dynamic programming, inspired by CKY \([74]\). With a multitude of possible discourse trees generated in this way, the tree structure minimizing the divergence between the document sentiment gold-label and the predicted sentiment, obtained by combining the tuples \((p_x, a_x)\) according to Equation \((9.1)\) is deemed to represent the document discourse structure.

\[
p = \frac{p_{c_l} * a_{c_l} + p_{c_r} * a_{c_r}}{a_{c_l} + a_{c_r}} \quad a = \frac{a_{c_l} + a_{c_r}}{2}
\] (9.1)

\(p_{c_l}\) and \(p_{c_r}\) represent the sentiment polarity labels of the left and right sub-tree respectively. \(a_{c_l}\) and \(a_{c_r}\) represent the importance scores, retrieved from the internal MIL attentions. \(p\) and \(a\) are the respective labels for the parent sentiment polarity and importance score \([59]\).

As described in Chapter \([3]\), the unconstrained CKY approach is not directly applicable for long documents (considered especially important in this work), since the spatial complexity of the CKY approach grows according to the Catalan number, with respect to the number of EDUs in a document. Hence, we apply the additional augmentations proposed in Section \([3.4]\).
reducing the spatial complexity through the application of a beam-search approach and improving the diversity in low-level trees through a stochastic extension. Further, we compute the additional nuclearity attribute, which has previously shown to be an important cue for a variety of downstream tasks \cite{69, 110, 142}. With these extensions, the discourse tree generation process can be effectively applied to documents of arbitrary length.

### 9.3.2 From Discourse to Sentiment

Discourse structure can be beneficial and complementary to sequential information for sentiment prediction, especially for long, complicated and nuanced documents (see Figure 9.1). We, therefore, take a balanced approach in this work, combining a sequential and tree-structured component to predict sentiment. Following the intuition of Bowman et al. \cite{15} and Choi et al. \cite{21}, we encode low-level representations in a sequential manner and use the inferred trees on higher levels to guide the prediction of the document-level sentiment.

**The Sequential Model Component**  With the HAN model presenting a strong baseline for many tasks, we take advantage of this contextualization for individual EDUs and for the document-level contextualization (see the bottom in Figure 9.3). In the standard HAN model, the first-level outputs (originally being sentence representations) are used as inputs to a document-level LSTM, augmented with an attention module, to generate the final hidden representation of a document. (see Equations 9.2 to 9.4).

\[
\begin{align*}
  u_i &= \tanh(W_h i + b) \\
  \alpha_i &= \frac{\exp(u_i^T c)}{\sum_{j \in d} \exp(u_j^T c)} \\
  h_d &= \sum_{i \in d} \alpha_i h_i
\end{align*}
\]

With \( h_i \) as the hidden state of EDU \( i \), obtained from the sentence-/EDU-level LSTM, \( c \) as the attention context vector and \( d \) representing the set
of all sentences/EDUs in the document. We inject discourse information by replacing the computation of the attention-weighted sum of the EDU embeddings (Equation 9.4) with a hierarchical TreeLSTM aggregation of the attention-weighted hidden states.

\[ h_d = \text{TreeLSTM}(\forall_{i \in d} \alpha_i h_i) \]  

(9.5)

We omit the description of the sentence/EDU level computations for brevity, as they are unchanged from the original HAN model.

**The Hierarchical Model Component** Using a tree-guided, hierarchical aggregation of EDU level hidden states to generate a discourse level hidden
representation of the document, we allow more important information according to the discourse tree to be more influential in the computation of the final document representation, as motivated by the example in Figure 9.1. The two crucial decisions on how to incorporate the discourse-guided tree aggregation are thereby:

1. The tree representation: Although discourse parsing typically processes constituency tree structures, most successful downstream applications of discourse parsing benefit from dependency discourse trees (e.g., Marcu [110], Ji and Smith [69], Shiv and Quirk [142]). Even though both tree representations are conveying the same information and
near isomorphic conversions are available \[116\], we believe that this is because of the different role that nuclearity plays in the tree representations. In particular, while in constituency trees nuclearity is an attribute of internal tree nodes, head-dependent relations in the dependency tree are fundamentally shaped by the nuclearity attribution. This more explicit encoding of nuclearity can benefit downstream applications. For this reason, we convert the RST constituency trees into dependency representations (see left of Figure 9.4).

2. The aggregation approach: The aggregation approach has a significant impact on the performance of the model. In this chapter, we choose the TreeLSTM model by Tai et al. \[153\], an evolution of the recursive neural network used in Ji and Smith \[69\]. Following the intuition for tree attention given by Ji and Smith \[69\], we add a conditional, non-competitive attention module to the child-sum TreeLSTM, augmenting the aggregation of text spans according to their position in the dependency discourse tree (see Equations 9.6 to 9.7). This extension has not been proposed as part of the TreeLSTM by Tai et al. \[153\], however, showed improved performance when used in combination with a recursive neural network for the task of discourse parsing \[69\], which lets us believe it can also enhance the TreeLSTM for our problem at hand.

The two core computations underlying the aggregation approach described above are thereby:

\[
\alpha_i = \sigma(h_{\text{head}}^T \times C \times h_{c_i}) \tag{9.6}
\]

\[
h_{\text{head}} = \text{LSTM}_{\text{cell}}(\sum_{i \in \text{dep}(h_{\text{head}})} \alpha_i h_{c_i}) \tag{9.7}
\]

with \(C\) as the attention matrix of dimension \(|h_{\text{head}}| \times |h_{c_i}|\), \(h_{\text{head}}\) representing the hidden state of the head node and \(\text{dep}(h_{\text{head}})\) returning the indices of the dependent child nodes of \(h_{\text{head}}\). Please note that the hidden
representation of every node in the dependency discourse tree is initialized
with the attention-weighted EDU representation obtained from the sequential
component and updated by the TreeLSTM function shown in Equation 9.7.
We combine the head node EDU representation with the dependants’ sub-tree
encoding during the bottom-up tree aggregation process (see top of Figure
9.3 and bottom of Figure 9.4). We name our new model DAH (Discourse
Augmented HAN).

9.4 Evaluation

In this section, we define the experimental setup and show empirical results of
our novel approach, predicting sentiment using sentiment-inspired discourse
parsing in the context of previous work. We present the datasets in Section
9.4.1, the evaluation metrics and their intuitive justifications are mentioned
in Section 9.4.2, followed by a short description of the baselines (Section
9.4.3). We finish the evaluation section by giving insights into our preliminary
evaluations, determining the system’s hyper-parameters in Section 9.4.4 and
describing the final experiments and results in Section 9.4.5.

9.4.1 Datasets

As shown in Figure 9.2, our proposed methodology requires two sets of
corpora. In the first step, as described in Section 9.3.1, we train a top-
performing discourse parser [160] on a discourse corpus containing RST-style
trees. In this step, we use two treebanks: (i) RST-DT, as described in Section
2.2 and (ii) MEGA-DT, our recently proposed “silver-standard” discourse
corpus [61], generated in an effort to provide an automatically annotated,
large-scale discourse treebank, presented in Chapter 3.

To evaluate the potential of the discourse treebanks to predict sentiment
in combination with our novel model architecture, we annotate a large-scale
sentiment dataset with discourse trees generated by the discourse parser
[160], trained on the corpora described above. The publicly available dataset
used in this work is the Yelp’13 dataset, published by Tang et al. [154] and
further described in Section 5.4.2. For models incorporating discourse, the
previously discourse segmented dataset published by Angelidis and Lapata [4] is used with an 80%/10%/10% train/dev/test split.

Please note that since we use the same base corpus for training the discourse parser (MEGA-DT) and predicting sentiment for the final evaluation (Yelp’13), we restrict the data used to train the discourse parser to the training portion of the corpus. This way we ensure that development and test documents are unseen during the training process.

9.4.2 Metrics

Previous models tackle the task of sentiment analysis by interpreting it as a classification problem. While this problem definition is valid for many text categorization tasks, we believe that sentiment analysis should be additionally evaluated as a regression task, taking the ordinal nature of the output into account. To more rigorously evaluate the models in our evaluation, we show four metrics for each system, including the commonly used accuracy and F1-score, as well as the Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics.

9.4.3 Baselines

We compare our new methodology against two closely related models, namely the Hierarchical Attention Network (HAN) by Yang et al. [177] and the MILNet model [4], which is used as part of the discourse augmentation process itself in Chapter 3. With those two closely related baselines we ensure that possible confounding factors in the comparison are minimized, allowing for a clear picture of the effectiveness of incorporating discourse structures into the task of sentiment analysis.

9.4.4 Encodings and Hyper-Parameters

To support a fair comparison, we use the same encodings and model hyperparameters in all systems. We replace the pre-trained word2vec encodings [114] used in the original HAN model, with standard GloVe embeddings [128]. We add MSE and MAE evaluation metrics to the publicly available
open-source deep-learning toolkit for the original HAN model. For the MILNet baseline, we align with our previous approach in Chapter 3 which is also consistent with the adapted HAN model. Regarding our novel approach, we convert the constituency tree output of the discourse parser into a dependency tree according to Hayashi et al. We run preliminary evaluations on the development set, comparing a set of loss function (namely Cross Entropy, MSE, MAE) and interpret the task as either a classification or a regression problem. We find that, without any further fine-tuning and adaptations, using a regression-based loss is not advisable. In accordance with the intuition described above, we execute a further hyper-parameter search on the main properties of the model itself, exploring a set of 5 learning rates (\{0.1, 0.05, 0.01, 0.05, 0.001\}) along with three optimization strategies (Adam [80], AdaGrad [33], SGD [138]). We follow the original HAN implementation using 100 neurons per layer for the bidirectional word and sentence/EDU encodings. The TreeLSTM module contains 512 neurons. The mini-batch size used in all models is set to 64, as suggested in Yang et al. [177]. Dropout is set to 50% for all models.

9.4.5 Experiments and Results

We compare our novel model obtained from sentiment-inspired discourse structures and standard treebanks against discourse agnostic systems, solely based on sequential representations on word- and sentence-level. As motivated in Figure 9.1, we believe that discourse information is especially useful for long documents, where sentiment is generally expressed in a more diverse and subtle way, as compared to short reviews with a clear positive or negative sentiment. We align our evaluation with this intuition by comparing the overall performance in Table 9.1 and further showing insights into the performance based on the document length in Figure 9.5.

The final comparison in Table 9.1 reports the performance of two baseline systems, not taking discourse information into account, along with two versions of our novel approach, incorporating discourse, and an ensemble

---

3[https://github.com/castorini/hedwig]

4Selected hyper-parameter is **bold**
Table 9.1: Final evaluation on the Yelp’13 dataset. Subscripts in model names indicate discourse augmentation treebanks used to generate discourse trees. Best model for each metric is bold. ≃ Performance statistically equivalent to HAN model, †MEGA-DT treebank significantly better than RST-DT with p-value 0.05. ‡MEGA-DT treebank marginally significantly better than RST-DT with p-value 0.05-0.1. *Statistically significant to best model on the metric. All significance computations are Bonferroni adjusted.

method. The performance of all models is averaged over 5 independent runs with different random initializations. All models using discourse (DAH_{RST-DT}, DAH_{MEGA-DT} and the ensemble of HAN and DAH_{MEGA-DT}) are trained with the top-performing discourse parser by Wang et al. [160]. All discourse-inspired models further employ an identical neural network architecture, allowing us to directly evaluate the impact of different types of discourse trees on the task of sentiment analysis.

The best average performance of an individual model (not using the ensemble method) is achieved by the sequential HAN model shown in the first row in Table 9.1. Even though the average result over 5 independent runs for the DAH_{MEGA-DT} system is below the HAN performance, they are statistically equivalent. When compared to the discourse-inspired DAH_{RST-DT} model, the performance increase of DAH_{MEGA-DT} is statistically significant on the accuracy and F1-score measures and marginally significant for the MSE and MAE. Interestingly, the MILNet model, which is used as an early part of the pipeline to generate the MEGA-DT discourse treebank, does perform substantially worse than the DAH_{MEGA-DT} model, which leads us
to believe that the combination of the CKY tree aggregation and the DAH sentiment neural network is able to extract valid and important sentiment cues to improve the performance, despite the potential propagation of error from the early stage MILNet component. Besides the individual models, we execute additional experiments with a model ensemble combining the two top-performing models (HAN and DAHMEGA-DT), taking their respective strengths in different document length ranges (as revealed in Figure 9.5) into account. The model will be explained in more detail below.

The results shown in the final evaluation (Table 9.1) indicate an equal performance of our new DAHMEGA-DT methodology when compared to the original HAN model. However, discourse should arguably be more useful for long documents. Therefore, we investigate the document length-dependent performance of the models by splitting the test set into 5 length-dependent bins to show the performance across different document sizes (measures by the number of words). We exclude the MILNet baseline in this evaluation due to its clearly inferior performance shown in Table 9.1.

The results shown in Figure 9.5 confirm our initial intuition on the usefulness of discourse structures for long documents in the two rightmost
bins. While the performance generally drops for longer documents, the performance decrease is more severe for the sequential HAN model. Generally, we believe that the task of sentiment prediction is harder on longer and more diverse documents, however, we also partially account the performance decrease to the small number of long documents in the Yelp’13 corpus, as shown in the support for each of the bins on the horizontal axis of Figure 9.5. While the support shown here is on the test portion, the general length distribution on the training and development sets are similarly skewed towards short documents. It can be further observed that the significant performance increase on the overall dataset achieved by the DAH\textsubscript{MEGA-DT} over the DAH\textsubscript{RST-DT} can be mostly attributed to the performance increase in the two right-most bins, containing documents with more than 632 words.

With this confirmation of our initial intuition, we generate a document length-dependent ensemble of the two top-performing models (HAN and DAH\textsubscript{MEGA-DT}) as mentioned above, to take advantage of the strength of both systems. We select the appropriate classifier with a simple threshold – the document length. To determine the threshold, we evaluate both models on the development set and select the average of the optimal threshold over 3 runs independently for each metric of interest. We then combine the results of the two top-performing models on the test set according to the determined threshold. As shown in Table 9.1, our ensemble approach significantly outperforms all the individual models, but admittedly only by a narrow margin. Nevertheless, the result indicates potential for further improvements in discourse-inspired sentiment analysis for long documents.

### 9.5 Contributions

In this chapter, we combine sequential methods with tree-structured approaches to effectively leverage “silver-standard” discourse annotations for the downstream task of sentiment analysis. Going from sentiment annotations to sentiment prediction through discourse augmentation, we enhance the performance for long documents, where (i) the influence of discourse becomes more prominent and (ii) predicting sentiment is rendered increasingly
difficult. In our evaluation, we show that the integration of modern, distantly supervised discourse parsing approaches (here: MEGA-DT) into existing, sequential sentiment analysis frameworks has the potential to enhance the sentiment prediction performance.

In conclusion, we extend our previous findings in Chapter 3 by validating the potential of the MEGA-DT “silver-standard” discourse treebank for the downstream task of sentiment analysis, showing that it can not only improve the task of domain-independent discourse parsing itself but also shows encouraging results for real-world downstream tasks.

As a result, we directly tackle research question (RQ2), showing the benefit of distantly supervised discourse structures for the downstream task of sentiment analysis. With all previous work presented in this thesis confirming the alignment of our generated structures with human-annotated gold-standard discourse trees, this chapter shows the potential of our inferred structures to predict downstream tasks in a purely data-driven approach, well aligned with our central hypothesis.

At this point, the MEGA-DT treebank and the underlying, distantly supervised discourse annotation approach have been proven to (i) align well with gold-standard discourse trees (Chapter 3), (ii) support human-annotated discourse structures in the pre-training/fine-tuning framework (Chapter 8) and (iii) improve the task of long-form sentiment analysis (Chapter 9). As a result, the last remaining question in regard to the usefulness of the MEGA-DT treebank is its potential to extend discourse theories, which we explore in Chapter 10.
Chapter 10

Linguistic Generalizations
From Silver-Standard Discourse Annotations

This chapter is based on our ACL 2021 publication [64], for which I was the lead investigator, working with fellow Ph.D. student Wen Xiao. I was partially responsible for the concept formation, formulation of research questions, data collection, and implementation of discourse-related components and evaluations. I further guided the paper composition and wrote substantial parts of the publication. Giuseppe Carenini supervised and supported the project.

10.1 Motivation

Ideally, research in Natural Language Processing (NLP) should balance and integrate findings from machine learning with insights and theories from linguistics. With the enormous success of data-driven approaches over the last decades, this balance has arguably and excessively shifted, with linguistic theories playing a less and less critical role. Even more importantly, only a few attempts have been made to improve such theories in light of recent empirical results. In this chapter, we take the growing divide between

173
linguistics and NLP as an occasion to explore whether the RST framework can be refined in a data-driven manner.

Specifically, one central requirement of the RST discourse theory, as for all linguistic theories, is that a trained human should be able to specify and interpret the discourse representations. While this is a clear advantage when trying to generate explainable outcomes, it also introduces problematic, human-centred simplifications; the most radical of which is arguably the nuclearity attribute, indicating the importance among siblings.

Intuitively, such a coarse (binary) importance assessment does not allow nuanced differences regarding sub-tree importance, which can potentially be critical for downstream tasks. For instance, the importance of two nucleus siblings is rather intuitive to interpret. However, having siblings annotated as “nucleus-satellite” or “satellite-nucleus” leaves the question of how much more important the nucleus sub-tree is compared to the satellite, as shown in Figure 10.1. In general, it is unclear (and unlikely) that the actual importance distributions between siblings with the same nuclearity attribution are consistent.

Based on this observation, we investigate the potential of replacing the binary nuclearity assessment postulated by RST with automatically generated, real-valued importance scores in a new, Weighted-RST framework (W-RST). In contrast to previous work developing computational models of RST-style discourse by simply applying machine learning methods to RST annotated treebanks [39, 68, 72, 92, 160, 180], we rely on our empirical studies showing that weighted “silver-standard” discourse trees can be inferred from auxiliary tasks such as sentiment analysis (Chapter 3) and summarization (Chapter 5).

In our evaluation, we assess both, computational benefits and linguistic insights. In particular, we find that automatically generated, weighted discourse trees can benefit key NLP downstream tasks. We further show that real-valued importance scores (at least partially) align with human annotations and can interestingly also capture uncertainty in human annotators, implying some alignment of the importance distributions with linguistic ambiguity.
Figure 10.1: Document *wsj_0639* from the RST-DT corpus with inconsistent importance differences between N-S attributions. (The top-level satellite is clearly more central to the overall context than the lower-level satellite. However, both are similarly assigned the satellite attribution by at least one annotator). Top relation: Annotator 1: N-S, Annotator 2: N-N.

10.2 Related Work

First introduced by Mann and Thompson [107], the Rhetorical Structure Theory (RST) has been one of the primary guiding theories for discourse analysis [18, 45, 100, 149, 183], discourse parsing [39, 68, 72, 92, 160, 180], and text planning [42, 50, 155]. Arguably, the weakest aspect of an RST representation is the nuclearity assessment, which makes an overly coarse differentiation between the primary and secondary importance of sub-trees. However, despite its binary assignment of importance and even though the nuclearity attribute is only one of three components of an RST tree, it has major implications for many downstream tasks, as already shown early on by Marcu [109], using the nuclearity attribute as the key signal in extractive summarization. Further work in sentiment analysis [14] also showed the importance of nuclearity for the task by first converting the constituency tree into a dependency tree (more aligned with the nuclearity attribute) and then using that tree to predict sentiment more accurately. Both of these
results indicate that nuclearity, even in the coarse RST version, already contains valuable information. Hence, we believe that this coarse-grained classification is reasonable when manually annotating discourse, but see it as a major point of improvement if a more fine-grained assessment could be correctly assigned. We, therefore, explore the potential of assigning a weighted nuclearity attribute in this chapter.

While plenty of studies have highlighted the important role of discourse for real-world downstream tasks, including summarization, [43, 171, 176], sentiment analysis [14, 58, 119] and text classification [69], more critical is our previous work exploring such connection in the opposite direction. In Chapter 3, we exploit sentiment-related information to generate “silver-standard” nuclearity annotated discourse trees, showing their potential for the domain-transfer discourse parsing task. Crucially for our purposes here, the approach described in Chapter 3 internally generates real-valued importance weights for trees.

For the task of extractive summarization, we follow our intuition given in Chapter 5, exploiting the connection between summarization and discourse. In particular, we demonstrate that the self-attention matrix learned during
The training of a transformer-based summarizer captures valid aspects of constituency and dependency discourse trees.

To summarize, building on our previous work to create discourse trees through distant supervision, we go a step further and generate weighted discourse trees from the sentiment analysis and summarization tasks in this chapter.

10.3 The W-RST Treebank Generation

Given the intuition above, we combine information retrieved from machine learning methods with insights from linguistics, replacing the human-centred nuclearity assignment with real-valued weights obtained from the sentiment analysis and summarization tasks\(^1\). An overview of the process to generate weighted RST-style discourse trees is shown in Figure 10.2, containing the training phase (left) and the W-RST discourse inference phase (center) described here. The W-RST discourse evaluation (right), is covered in Section 10.4.

\(^1\)Please note that both tasks use binarized discourse trees, as commonly used in computational models of RST.
10.3.1 Weighted Trees from Sentiment

To generate weighted discourse trees from sentiment, we modify our approach presented in Chapter 3, by removing the nuclearity discretization component. 

An overview of our method is shown in Figure 10.2 (top) and a detailed view is presented in the left and center parts of Figure 10.3. First (on the left), we train the Multiple Instance Learning (MIL) model proposed by Angelidis and Lapata on a corpus with document-level sentiment gold labels, internally annotating each input unit (in our case EDUs) with a sentiment- and attention-score. After the MIL model is trained (center), a tuple \((s_i, a_i)\) containing a sentiment score \(s_i\) and attention \(a_i\) is extracted for each EDU \(i\). Based on these tuples representing leaf nodes, the CKY algorithm is applied to find the tree structure to best align with the overall document sentiment, through a bottom-up aggregation approach defined as:

\[
s_p = \frac{s_l \ast a_l + s_r \ast a_r}{a_l + a_r} \quad a_p = \frac{a_l + a_r}{2}
\]

with nodes \(l\) and \(r\) as the left and right child nodes of \(p\) respectively. The attention scores \((a_l, a_r)\) are interpreted as the importance weights for the respective sub-trees \((w_l = a_l/(a_l + a_r)\) and \(w_r = a_r/(a_l + a_r)\)), resulting in a complete, normalized and weighted discourse structure as required for W-RST. We call the discourse treebank generated with this approach W-RST-Sent.

10.3.2 Weighted Trees from Summarization

In order to derive weighted discourse trees from a summarization model we follow our approach in Chapter 5, generating weighted discourse trees from the self-attention matrices of a transformer-based summarization model. An overview of our method is shown in Figure 10.2 (bottom) and a detailed

---

2Code of Chapter 3 available at https://github.com/nlpat/MEGA-DT

3Equations taken from Chapter 3, as presented in Huber and Carenini

4Code of Chapter 5 available at https://github.com/Wendy-Xiao/summ_guided_disco
Figure 10.4: Three phases of our approach. Left/Center: Detailed view into the generation of weighted RST-style discourse trees using the summarization downstream task. Right: Summarization discourse application evaluation.

We start by training a transformer-based extractive summarization model (left), containing three components: (i) A pre-trained BERT EDU Encoder generating EDU embeddings, (ii) a standard transformer architecture as proposed in Vaswani et al. [156] and (iii) a final classifier, mapping the outputs of the transformer to a probability score for each EDU, indicating whether the EDU should be part of the extractive summary.

With the trained transformer model, we then extract the self-attention matrix $A$ and build a discourse tree in bottom-up fashion (as shown in the center of Figure 10.4). Specifically, the self-attention matrix $A$ reflects the relationships between units in the document, where entry $A_{ij}$ measures how much the $i$-th EDU relies on the $j$-th EDU. Given this information, we generate an unlabeled constituency tree using the CKY algorithm [74], optimizing the overall tree score, as done in Chapter 5 and presented as Xiao et al. [173].

In terms of weight assignment, given a sub-tree spanning EDUs $i$ to $j$, split into child-constituents at EDU $k$, then $\max(A_{i,k,(k+1):j})$, representing the maximal attention value that any EDU in the left constituent is paying to an EDU in the right child-constituent, reflects how much the left sub-tree relies on the right sub-tree, while $\max(A_{(k+1):j,i,k})$ defines how much the right sub-tree depends on the left. We define the importance weights of the left ($w_l$) and right ($w_r$) sub-trees as:
\[ w_l = \max(A_{(k+1)j,i:k})/(w_l + w_r) \]
\[ w_r = \max(A_{i:k,(k+1)j})/(w_l + w_r) \]

In this way, the importance scores of the two sub-trees represent a real-valued distribution. In combination with the unlabeled structure computation, we generate a weighted discourse tree for each document. We call the discourse treebank generated with the summarization downstream information **W-RST-Summ**.

### 10.4 The W-RST Weight Integration

To assess the potential of W-RST, we consider two evaluation scenarios (Figure 10.2, right): (i) Apply weighted discourse trees to the tasks of sentiment analysis and summarization and (ii) analyze the weight alignment with human annotations.

#### 10.4.1 Weight-based Discourse Applications

In this evaluation scenario, we address the question of whether W-RST trees can support downstream tasks better than traditional RST trees with nuclearity. Specifically, we leverage the discourse trees learned from sentiment for the sentiment analysis task itself and, similarly, rely on the discourse trees learned from summarization to benefit the summarization task.

**Sentiment Analysis**

In order to predict the sentiment of a document in **W-RST-Sent** based on its weighted discourse tree, we need to introduce an additional source of information to be aggregated according to such tree. Here, we choose word embeddings, as commonly used as an initial transformation in many models tackling the sentiment prediction task [3, 60, 79, 153, 177]. To avoid introducing additional confounding factors through sophisticated tree aggregation approaches (e.g. TreeLSTMs [153]), we select a simple method, aiming to
directly compare the inferred tree structures and allowing us to better assess
the performance differences originating from the weight/nuclearity attribution (see right step in Figure [10.3]). More specifically, we start by computing
the average word embedding for each leaf node \( \text{leaf}_i \) (here containing a single
EDU) in the discourse tree,

\[
\text{leaf}_i = \frac{\sum_{j=0}^{j<|\text{leaf}_i|} \text{Emb}(\text{word}_j^i)}{|\text{leaf}_i|}
\]

With \(|\text{leaf}_i|\) as the number of words in leaf \( i \), \( \text{Emb}(\cdot) \) being the embedding
lookup and \( \text{word}_j^i \) representing word \( j \) within leaf \( i \). Subsequently, we
aggregate constituents, starting from the leaf nodes (with \( \text{leaf}_i \) as embedding
c constituent \( c_i \)), according to the weights of the discourse tree. For any two
sibling constituents \( c_l \) and \( c_r \) of the parent sub-tree \( c_p \) in the binary tree, we
compute

\[
c_p = c_l \ast w_l + c_r \ast w_r
\]

with \( w_l \) and \( w_r \) as the real-valued weight-distribution extracted from the
inferred discourse tree and \( c_p, c_l \) and \( c_r \) as dense encodings. We aggregate the
complete document in bottom-up fashion, eventually reaching a root node
embedding containing a tree-weighted average of the leaf nodes. Given the
root-node embedding representing a complete document, a simple Multilayer
Perceptron (MLP) trained on the original training portion of the MIL model
is used to predict the sentiment of the document.

**Summarization**

In the evaluation step of the summarization model (right of Figure [10.4]), we
use the weighted discourse tree of a document in \( W-RST\text{-Summ} \) to predict its
extractive summary by applying an adaptation of the unsupervised method
by Marcu [109].

We choose this straightforward algorithm over more elaborate and hyper-
parameter-heavy approaches to avoid confounding factors since our aim is to
evaluate the potential of the weighted discourse trees compared to standard
RST-style annotations. In the original algorithm by Marcu [109], a summary is computed based on the nuclearity attribute by recursively computing the importance scores for all units as

\[
S_n(u, N) = \begin{cases} 
  d_N, & u \in \text{Prom}(N) \\
  S(u, C(N)) \quad \text{s.t.} \\
  S(u, C(N)) + w_N, & u \in C(N) \\
  \end{cases}
\]

where \( C(N) \) represents the child of \( N \), and \( \text{Prom}(N) \) is the promotion set of node \( N \), which is defined in bottom-up fashion as follows: (i) \( \text{Prom} \) of a leaf node is the leaf node itself. (ii) \( \text{Prom} \) of an internal node is the union of the promotion sets of its nucleus children. Furthermore, \( d_N \) represents the level of a node \( N \), computed as the distance from the level of the lowest leaf node. This way, units in the promotion set originating from nodes that are higher up in the discourse tree are amplified in their importance compared to those from lower levels.

As for the \( W-RST\)-Summ discourse trees with real-valued importance weights, we extend the algorithm by Marcu [109], replacing the promotion set with real-valued importance scores as follows:

\[
S_w(u, N) = \begin{cases} 
  d + w_N, & N \text{ is leaf} \\
  S_w(u, C(N)) + w_N, & u \in C(N) \\
  \end{cases}
\]

Once \( S_n \) or \( S_w \) are computed, the top-k units of the highest promotion set or with the highest importance scores respectively are selected into the final summary.

**Nuclearity-attributed Baselines**

To test whether the W-RST trees are effectively predicting the downstream tasks, we generate traditional RST trees with nuclearity to compare against. However, moving from weighted discourse trees to coarse nuclearity requires the introduction of a threshold. More specifically, while “nucleus-satellite”
and “satellite-nucleus” assignments can be naturally generated depending on the distinct weights, in order to assign the third “nucleus-nucleus” class, frequently appearing in RST-style treebanks, we need to specify how close two weights have to be for such configuration to apply. Formally, we set a threshold \( t \) as follows:

\[
\text{If: } |w_l - w_r| < t \rightarrow \text{nucleus-nucleus}
\]

\[
\text{Else: If: } w_l > w_r \rightarrow \text{nucleus-satellite}
\]

\[
\text{Else: If: } w_l \leq w_r \rightarrow \text{satellite-nucleus}
\]

This way, RST-style treebanks with nuclearity attributions can be generated from \( W-RST-Sent \) and \( W-RST-Summ \) and used for the sentiment analysis and summarization downstream tasks. For the nuclearity-attributed baseline of the sentiment task, we use a similar approach as for the W-RST evaluation procedure but assign two distinct weights \( w_n \) and \( w_s \) to the nucleus and satellite child respectively. Since it is not clear how much more important a nucleus node is compared to a satellite using the traditional RST notation, we define the two weights based on the threshold \( t \) as

\[
w_n = 1 - (1 - 2t)/4 \quad w_s = (1 - 2t)/4
\]

The intuition behind this formulation is that for a high threshold \( t \) (e.g., 0.8), the nuclearity needs to be very prominent (i.e., the difference between the normalized weights needs to exceed 0.8), making the nucleus clearly more important than the satellite, while for a small threshold (e.g., 0.1), even relatively balanced weights (for example \( w_l = 0.56, w_r = 0.44 \)) will be assigned as “nucleus-satellite”, resulting in the potential difference in importance of the siblings to be less eminent.

For the nuclearity-attributed baseline for summarization, we directly apply the original algorithm by Marcu [109] as described in Section 10.4.1. However, when using the promotion set to determine which EDUs are added to the summarization, potential ties can occur. Since the discourse tree does not provide any information on how to prioritize those, we randomly select units from the candidates, whenever there is a tie. This avoids exploiting
any positional bias in the data (e.g., the lead bias), which would confound the results.

### 10.4.2 Weight Alignment with Human Annotation

As for our second W-RST discourse evaluation task, we investigate if the real-valued importance weights align with human annotations. To be able to explore this scenario, we generate weighted tree annotations for an existing discourse treebank (here: RST-DT [18]). For this evaluation task, we verify if: (i) The nucleus in a gold-annotation generally receives more weight than a satellite (i.e. if importance weights generally favour nuclei over satellites) and, similarly, if nucleus-nucleus relations receive more balanced weights. (ii) In accordance with Figure 10.1, we further explore how well the weights capture the extent to which a relation is dominated by the nucleus. Here, our intuition is that for inconsistent human nuclearity annotations the spread should generally be lower than for consistent annotations, assuming that human misalignment in the discourse annotation indicates ambivalence on the importance of sub-trees.

To test for these two properties, we use discourse documents individually annotated by two human annotators and analyze each sub-tree within the doubly-annotated documents with consistent inter-annotator structure assessment for their nuclearity assignment. For each of the 6 possible inter-annotator nuclearity assessments, consisting of 3 consistent annotation classes
(namely N-N/N-N, N-S/N-S and S-N/S-N) and 3 inconsistent annotation classes (namely N-N/N-S, N-N/S-N and N-S/S-N) we explore the respective weight distribution of the document annotated with the two W-RST tasks – sentiment analysis and summarization (see Figure 10.5). We compute an average spread $s_c$ for each of the 6 inter-annotator nuclearity assessments classes $c$ as

$$s_c = \frac{\sum_{j=0}^{j<|c|} (w^l_j - w^r_j)}{|c|}$$

With $w^l_j$ and $w^r_j$ as the weights of the left and right child node of sub-tree $j$ in class $c$, respectively.

10.5 Evaluation

10.5.1 Experimental Setup

**Sentiment Analysis:** We follow our previous approach in Chapter 3 for the model training and W-RST discourse inference steps (left and center in Figure 10.3) using the adapted MILNet model from Angelidis and Lapata [4]. Next, discourse trees are generated using the best-performing heuristic CKY method with the stochastic exploration-exploitation trade-off (beam size 10, linear decreasing $\tau$). As word-embeddings in the W-RST discourse evaluation (right in Figure 10.3), we use GloVe embeddings [128], which previous work [60, 153] indicates to be suitable for aggregation in discourse processing. For training and evaluation of the sentiment analysis task, we use the 5-class Yelp’13 review dataset [154]. To compare our approach against the traditional RST approach with nuclearity, we explore the impact of 11 distinct thresholds for the baseline described in Section 10.4.1, ranging from 0 to 1 in 0.1 intervals.

---

5We don’t take the order of annotators into consideration, mapping N-N/N-S and N-S/N-N both onto N-N/N-S.
**Summarization:** To be consistent with the RST theory, our summarizer extracts EDUs instead of sentences from a given document. The model is trained on the EDU-segmented CNNDM dataset containing EDU-level Oracle labels published by Xu et al. [176]. We further use a pre-trained BERT-base (“uncased”) model to generate the embeddings of EDUs. The transformer used is the standard model with 6 layers and 8 heads in each layer ($d = 512$). We train the extractive summarizer on the training set of the CNNDM corpus [117] and pick the best attention head using the RST-DT dataset [18] as the development set. We test the trees by running the summarization algorithm in Marcu [109] on the test set of the CNNDM dataset and select the top-6 EDUs based on the importance score to form a summary in natural order. Regarding the baseline model using thresholds, we apply the same 11 thresholds used for the sentiment analysis task.

**Weight Alignment with Human Annotation:** As discussed in Section 10.4.2, the human evaluation requires two parallel discourse trees for every document. Luckily, in the RST-DT corpus published by Carlson et al. [18], 53 of the 385 documents are doubly tagged by a second linguist. We use the 53 documents containing 1,354 consistent structure annotations between the two analysts to evaluate the linguistic alignment of our generated W-RST documents with human discourse interpretations. Out of the 1,354 structure-aligned sub-trees, in 1,139 cases both annotators agreed on the nuclearity attribute, while 215 times a nuclearity mismatch appeared, as shown in detail in Table 10.1.

**10.5.2 Results and Analysis**

The results of the experiments on the discourse applications for sentiment analysis and summarization are shown in Figure 10.6. The results for sentiment analysis (top) and summarization (bottom) thereby show a similar trend: With an increasing threshold and therefore a larger number of N-N relations (shown as grey bars in the Figure), the standard RST baseline (blue line) consistently improves for the respective performance measure of
both tasks. However, reaching the best performance at a threshold of 0.8 for sentiment analysis and 0.6 for summarization, the performance starts to deteriorate. This general trend seems reasonable, given that N-N relations represent a rather frequent nuclearity connection, however classifying every connection as N-N leads to a severe loss of information. Furthermore, the performance suggests that while the N-N class is important in both cases, the optimal threshold varies depending on the task (and potentially also the corpus), making further task-specific fine-tuning steps mandatory. The weighted discourse trees following our W-RST approach, on the other hand, do not require the definition of a threshold, resulting in a single, promising performance (red line) for both tasks in Figure 10.6. For comparison, we apply the generated trees of a standard RST-style discourse parser (here the Two-Stage parser by Wang et al. [160]) trained on the RST-DT dataset [18] on both downstream tasks. The fully-supervised parser reaches an

Figure 10.6: Top: Sentiment Analysis accuracy of the W-RST model compared to the standard RST framework with different thresholds. Bottom: Average ROUGE score (ROUGE-1, -2 and -L) of the W-RST summarization model compared to different thresholds. Full numerical results are shown in Appendix C.1.
Table 10.1: Statistics on consistently and inconsistently annotated samples of the 1,354 structure-aligned sub-trees generated by two distinct human annotators.

<table>
<thead>
<tr>
<th></th>
<th>N-N</th>
<th>N-S</th>
<th>S-N</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-N</td>
<td>273</td>
<td>99</td>
<td>41</td>
</tr>
<tr>
<td>N-S</td>
<td></td>
<td>694</td>
<td>75</td>
</tr>
<tr>
<td>S-N</td>
<td></td>
<td></td>
<td>172</td>
</tr>
</tbody>
</table>

Table 10.2: Confusion Matrices based on human annotation showing the absolute weight spread using the Sentiment (left) and Summarization (right) tasks on 620 and 791 sub-trees aligned with the human structure prediction, respectively. Cell upper value: Absolute weight spread for the respective combination of human-annotated nuclearities. Lower value (in brackets): Support for this configuration.

<table>
<thead>
<tr>
<th></th>
<th>Sent</th>
<th></th>
<th>Sum</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N-N</td>
<td>N-S</td>
<td>S-N</td>
<td>N-N</td>
</tr>
<tr>
<td>N-N</td>
<td>-0.228 (106)</td>
<td>-0.238 (33)</td>
<td>-0.240 (19)</td>
<td>0.572 (136)</td>
</tr>
<tr>
<td>N-S</td>
<td>-0.038 (325)</td>
<td>-0.044 (22)</td>
<td></td>
<td>0.713 (418)</td>
</tr>
<tr>
<td>S-N</td>
<td></td>
<td></td>
<td>-0.278 (115)</td>
<td></td>
</tr>
</tbody>
</table>

accuracy of 44.77% for sentiment analysis and an average ROUGE score of 26.28 for summarization. While the average ROUGE score of the fully-supervised parser is above the performance of our W-RST results for the summarization task, the accuracy on the sentiment analysis task is well below our approach. We believe that these results are a direct indication of the problematic domain adaptation of fully supervised discourse parsers, where the application on a similar domain (Wall Street Journal articles vs. CNN-Daily Mail articles) leads to superior performances compared to
Table 10.3: Confusion Matrices based on human annotation showing the weight spread relative to the task-average for Sentiment (left) and Summarization (right), aligned with the human structure prediction, respectively. Cell value: Relative weight spread as the divergence from the average spread across all cells in Table 10.2. ∅ = Average value of absolute scores.

<table>
<thead>
<tr>
<th></th>
<th>Sent</th>
<th>N-N</th>
<th>N-S</th>
<th>S-N</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-N</td>
<td>∅-0.36</td>
<td>∅-0.43</td>
<td>∅-0.45</td>
<td></td>
</tr>
<tr>
<td>N-S</td>
<td>-</td>
<td>∅+1.00</td>
<td>∅+0.96</td>
<td></td>
</tr>
<tr>
<td>S-N</td>
<td>-</td>
<td>-</td>
<td>∅-0.72</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Sum</th>
<th>N-N</th>
<th>N-S</th>
<th>S-N</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-N</td>
<td>∅-0.13</td>
<td>∅+0.13</td>
<td>∅-0.66</td>
<td></td>
</tr>
<tr>
<td>N-S</td>
<td>-</td>
<td>∅+1.00</td>
<td>∅-0.56</td>
<td></td>
</tr>
<tr>
<td>S-N</td>
<td>-</td>
<td>-</td>
<td>∅+0.22</td>
<td></td>
</tr>
</tbody>
</table>

our distantly supervised method, however, with larger domain shifts (Wall Street Journal articles vs. Yelp customer reviews), the performance drops significantly, allowing our distantly supervised model to outperform the supervised discourse trees for the downstream task. Arguably, this indicates that although our weighted approach is still not competitive with fully-supervised models in the same domain, it is the most promising solution available for inter-domain (sometimes also called cross-domain) discourse parsing.

With respect to exploring the weight alignment with human annotations, we show a set of confusion matrices based on human annotation for each W-RST discourse generation task on the absolute and relative weight spread in Tables 10.2 and 10.3, respectively. The results for the sentiment analysis task are shown on the left of both tables and the performance for the summarization task is shown on the right. For instance, the top right cell of the left confusion matrix in Table 10.2 shows that for 19 sub-trees in the doubly annotated subset of RST-DT one of the annotators labelled the sub-tree with a nucleus-nucleus nuclearity attribution, while the second annotator identified it as satellite-nucleus. The average weight spread (see Section 10.4.2) for these 19 sub-trees is −0.24. Regarding Table 10.3, the average spread across Table 10.2 is defined as
\[ \varnothing = \sum_{c_i \in C} (c_i) / |C| \]

with \( C = \{c_1, c_2, ..., c_6\} \) containing the cell values in the upper triangle matrix. \( \varnothing \) is then subtracted from each cell value \( c_i \) and normalized by:

\[ max = \max_{c_i \in C} (|c_i - \varnothing|) \]

With \( \varnothing = -0.177 \) and \( max = 0.1396 \) in the left table of Figure 10.2, we transform the value of \(-0.24\) in the top right cell into \((-0.24 - \text{avg})/max = -0.45\). Moving to the analysis of the results, we find the following trends:

1. As presented in Table 10.2, the sentiment analysis task tends to strongly over-predict S-N (i.e., \( w_l << w_r \)), leading to negative spreads in all cells. In contrast, the summarization task is heavily skewed towards N-S assignments (i.e., \( w_l >> w_r \)), leading to exclusively positive spreads. We believe both trends are consistent with the intrinsic properties of the tasks, given that the general structure of reviews tends to become more important towards the end of a review (leading to increased S-N assignments), while for summarization, the lead bias potentially produces the overall strong nucleus-satellite trend.

2. We investigate the relative weight spreads for different human annotations in Table 10.3 showing that the consistently annotated samples of the two linguists (along the diagonal of the confusion matrices) align reasonably. The most positive weight spread is consistently found in the agreed-upon nucleus-satellite case, while the nucleus-nucleus annotation has, as expected, the lowest divergence (i.e., closest to zero) along the diagonal in Table 10.3.

3. Regarding the inconsistently annotated samples (shown in the triangle matrix above the diagonal) it becomes clear that in the sentiment analysis model the values for the N-N/N-S and N-N/S-N annotated samples (top row in Table 10.3) are relatively close to the average value. This indicates that similar to the nucleus-nucleus case, the weights are
also ambivalent, with the N-N/N-S value (top center) slightly larger than the value for N-N/S-N (top right). The N-S/S-N case for the sentiment analysis model is less aligned with our intuition, showing a strongly negative weight spread (i.e. $w_l << w_r$) where we would have expected a more ambivalent result with $w_l \approx w_r$. For summarization, we see a very similar trend with the values for N-N/N-S and N-N/S-N annotated samples. Again, both values are close to the average, with the N-N/N-S cell showing a more positive spread than N-N/S-N. However, the consistent satellite-nucleus annotation (bottom right cell) seems misaligned with the rest of the table, following instead the general trend for summarization described in Table 10.2.

All in all, the results suggest that the values in most cells are well aligned with what we would expect regarding the relative spread. Interestingly, human uncertainty appears to be reasonably captured in the weights, describing the relative importance of sibling sub-trees.

10.6 Contributions

In this chapter, we propose W-RST as a new, weighted RST-style discourse framework, where the binary nuclearity assessment postulated by RST is replaced with more expressive weights, automatically generated from auxiliary tasks. We thereby expose the internal weights of two previously proposed, distantly supervised discourse annotation projects presented in Chapters 3 and 5 (sentiment analysis and summarization). In line with the internal weights of automated discourse prediction models, we interpret the scalar values as the relative importance of EDUs and internal sub-trees, leading to our proposal of the more fine-grained, weighted framework. In a series of experiments, we find that W-RST is beneficial to the key downstream tasks of sentiment analysis and summarization. Further, we show that W-RST trees interestingly align with the uncertainty of human annotations.

In conclusion, we provide evidence for the usefulness of our distantly supervised discourse structures from sentiment analysis and summarization beyond their applicability to discourse parsing and related downstream tasks.
Complementing our previous results, we show the alignment of internally used weights in the neural models with the RST nuclearity attribute, another strong indicator of the validity and usefulness of our inferred structures.

Regarding the central research questions in this thesis, we answer (RQ3) in this chapter by showing the reasonable alignment of our inferred discourse structures from the sentiment analysis and summarization auxiliary tasks with human-annotated discourse structures and their uncertainty. This important evaluation shows the value of our annotations, beyond just discourse and downstream applications, but for the underlying theory itself, arguably well aligned with our hypothesis presented in Chapter 1.
Conclusion
Chapter 11

Conclusion

Aiming to better understand, process and generate natural language, one central goal of the research field of Natural Language Processing (NLP) is to develop automated models and methods to generate meaningful insights from textual data. Insights can thereby take many forms, such as task-dependent labels (e.g., sentiment, summarization or topic labels), linguistic roles (e.g., coreference or part-of-speech tags), and structural information (e.g., syntactic, semantic, pragmatic structures). With early NLP research targeting more “local” problems, oftentimes inferring labels for individual textual units (e.g., word-level sentiment labels), an obvious and distinct trend toward more contextualized, multi-sentential models has been a main driving factor in the last decades. With researchers exploiting progressively larger models, utilizing information beyond the local context, systems take the inherently ambiguous and context-dependent nature of natural language into account. Given the increasing importance of contextualized models using global (document-level) features, robustly extracted insights at the document level constitute a central objective in the field of NLP.

In this thesis, we tackle one of the most popular approaches to encoding document-level structural information, namely, discourse structures. More specifically, we take the first steps towards overcoming one of the major limitations for the widespread application of discourse structures in the field of NLP to date, the lack of large-scale datasets. The main goal of this
thesis is to overcome the data sparsity issue and, hence, improve document-
level natural language understanding through the data-driven application
of discourse theories. In line with this goal, we pose three central research
questions in Chapter 1:

• RQ1 How can we effectively generate discourse annotations at scale,
without the current limitations inherent to the human annotation pro-
cess?

• RQ2 Can large-scale, automatically generated discourse annotations
enhance real-world auxiliary tasks?

• RQ3 Can large-scale, automatically generated discourse annotations
augment linguistic theories of discourse?

which we subsequently answer in the three parts of this thesis by proposing
novel approaches to infer and apply distantly and self-supervised computational
models following our central hypothesis, stating:

Context-sensitive NLP tasks internally encode some notion of discourse
information to prioritize, structure, and connect parts of the input.

In Part I we tackle the problem of how to infer discourse structures from
large-scale, naturally annotated auxiliary datasets in a distantly supervised
framework. We show three promising approaches to how different parts of
complete discourse trees can be successfully generated.

In Chapter 3, we show a discourse structure and nuclearity inference
method using sentiment annotated data (here: Yelp reviews). Combining a
neural Multiple Instance Learning approach with the traditional CKY chart
parsing method, we generate the new and large-scale MEGA-DT discourse
treebank. Our MEGA-DT “silver-standard” dataset thereby contains nearly
two orders of magnitude more data points than any existing, gold-standard
corpus (e.g., RST-DT, GUM, ...). In our evaluation, we show the value and
validity of our generated treebank by comparing its inter-domain discourse
parsing performance to gold-standard approaches (oftentimes also called zero-shot inference). Our obtained results paint a clear picture regarding the usefulness of MEGA-DT, reaching the best performance on the challenging inter-domain discourse parsing task, even compared to fully supervised methods.

In Chapter 4, we explicitly tackle the generation of high-level discourse structures, an arguably difficult, but important part of the complete discourse tree prediction. To robustly infer discourse structures at high levels of a document, the topical structure of the document plays a crucial role, defining how paragraphs and segments are related. Following this motivation, we exploit the state-of-the-art neural topic segmentation model to obtain a sequence of topic-break probabilities. From there, we generate discourse trees in a greedy manner, following the auxiliary topic segmentation annotation. In our evaluation, we take a detailed look at the performance of discourse generation methods on different levels of a document. To the best of our knowledge, this is the first time such a detailed evaluation has been proposed and executed for discourse-related problems. We find that while the sentiment auxiliary task, described in Chapter 3, predicts discourse structures on low- and mid-levels of the document well, the topic segmentation task provides a better supervision signal on high-level structures.

In Chapter 5, we close the first part of this thesis with a novel proposal on how to use information from the extractive summarization task to generate discourse structures with nuclearity attributes. To be able to exploit information from the summarization task, we first train a transformer-based neural summarizer on large-scale summarization datasets (here: CNN-DailyMail and New York Times) and subsequently extract the internal self-attention matrices to infer discourse trees following the CKY, Eisner and Chu-Liu-Edmonds parsing algorithms. In our evaluation, we find that the transformer self-attention matrices encode some notion of discourse, outperforming random baselines. We further find that more nuclearity-related than structural information is encoded in the model.

Overall, in Part I, we answer research question (RQ1) on how discourse structures can be generated at scale, without the current limitations inherent
to human annotations, by confirming our hypothesis posed in Chapter 1. Proposing and evaluating three novel approaches to overcome the current lack of available training data in size and domains using distant supervision, we take the first steps to enable future discourse-related research to scale to the requirements of modern deep learning models.

In Part II we take the general idea of Part I one step further, replacing the auxiliary task supervision (e.g., sentiment labels) with a self-supervised objective. Using a self-supervised approach thereby entirely removes the dependency on annotated datasets, enabling methods to access even larger data sources, plausibly inferring more general discourse structures.

In Chapter 6, we follow the self-supervised discourse generation objective by exploiting existing, large-scale pre-trained language models (PLMs). Similar to Chapter 5, we extract the internal self-attention matrices of the transformer model, argued to contain the reciprocal dependence of tokens to each other. To assess the robustness and locality of discourse structures in PLMs (here: BERT and BART), we examine the captured information across self-attention heads and diverse fine-tuning tasks, finding that inferred discourse trees are indeed local and robust. In further evaluations, we show the structures to be competitive with distantly supervised approaches, without relying on explicit supervision signals.

In Chapter 7, we propose another self-supervised approach to generate discourse structures, interpreting the task in a more linguistic manner. Here, we argue that flat, sequential models do not take advantage of the well-known, hierarchical structure of text. As a result, we propose a tree-style autoencoder framework (T-AE), encoding documents into a fixed-size, latent dimension following the inferred tree structure. In this initial attempt to propose a more linguistically inspired approach to self-supervision, we find that the obtained trees are non-trivial, contain interesting discourse-related structures, and align with the downstream task of sentiment analysis.

In conclusion, Part II extends our previous answer to research question (RQ1) on the generation of discourse structures at scale. We show promising potential of modern self-supervised models (i.e., PLMs) and more linguisti-
cally inspired tree-style approaches for the task of self-supervised discourse inference. Similar to the methodologies in Part I, our solutions presented in this part allow for the direct generation of large-scale, “silver-standard” discourse structures, bypassing the human annotation process. Confirming the central hypothesis of this thesis, we show that self-supervised models internally encode some notion of discourse as well.

In Part III we show application scenarios for our generated discourse structures from the distant supervision signals of sentiment analysis and extractive summarization. This part plays a pivotal role in the overall contribution of this thesis, showing the value of our inferred trees beyond the alignment with gold-standard discourse trees, exploring their potential for the task of discourse parsing, the downstream task of sentiment analysis and the application to refine linguistic theories.

In Chapter 8, we show the potential of our MEGA-DT discourse treebank as a pre-training corpus for neural discourse parsers. Specifically, we propose a novel neural discourse parsing architecture based on the shift-reduce framework, combining modern pre-trained language models (PLMs) with structural features. We show that using the RoBERTa model in combination with simple, structural priors reaches on-par performance when compared with current state-of-the-art (SOTA) models on the RST discourse parsing task. Adding the additional pre-training step on our MEGA-DT treebank, presented in Chapter 3, the newly proposed supervised discourse parser outperforms previous systems and reaches SOTA performance. The results obtained in Chapter 8 hence, confirm the effectiveness of our distantly supervised discourse generation approach from sentiment analysis.

In Chapter 9, we further confirm the validity of our generated “silver-standard” discourse structures from Chapter 3. Here, we propose a hybrid model for the task of sentiment analysis, combining the hierarchical HAN model with the commonly used TreeLSTM aggregation approach. Combining sequential encodings at the EDU level with tree-style aggregations at the document level, our evaluation shows the potential of discourse trees from the MEGA-DT treebank to guide the task of sentiment prediction for long
documents, showing the alignment of our generated trees with real-world downstream tasks.

In Chapter 10 we show the potential of our inferred discourse structures from the two distantly supervised tasks of sentiment analysis and summarization to improve and extend the Rhetorical Structure Theory (RST). After showing their alignment with human-annotated trees in Chapters 3 and 5 we take the next step in this chapter by replacing the coarse-grained, binary nuclearity attribute with more expressive weights, extracted from the distantly supervised methodologies. In our evaluation regarding the potential of weighted discourse trees to support downstream tasks (here: sentiment analysis and extractive summarization), we find that weights are superior to the binary nuclearity attribute for most cases. Even more interestingly, we also show the alignment of our automatically inferred weights with human annotations and their uncertainty, finding that weights are surprisingly well aligned with the ambiguity in human annotators.

In conclusion, Part III shows a set of strong results further confirming the validity, robustness and value of our distantly supervised approaches presented in Part I. As such, we answer (RQ1), (RQ2) and (RQ3), on how well automatically generated discourse annotations can predict discourse, enhance real-world auxiliary tasks, and inform linguistic theories of discourse. We directly quantify the value of automatically generated discourse structures along all three dimensions, further confirming our central hypothesis.

Looking at the thesis as a whole, we propose the first line of research to successfully apply distantly and self-supervised approaches to generate large-scale “silver-standard” discourse structures in a data-driven manner. Our proposed answers to the two central research questions are based on the underlying hypothesis that the previously shown synergistic relationship between discourse parsing and downstream tasks is bidirectional for tasks such as text classification [69], summarization [43] and sentiment analysis [14, 58, 119]. More specifically, as stated in Chapter 1 we hypothesize that these tasks require some notion of discourse information being internally learned to prioritize, structure, and connect parts of the input text to successfully predict
sentiment, summaries, topic labels and language in general. Throughout this thesis, we verify our central hypothesis by successfully answering the three research questions. We show promising approaches to infer discourse structures without human annotations and quantify their usefulness in a set of experiments for discourse parsing, downstream tasks and the application to linguistic theories. Specifically, we show that all three auxiliary tasks to extract discourse structures in a distantly supervised manner do indeed encode some notion of discourse. We further show that self-supervised models, purely relying on the textual form, also capture some degree of discourse-related information.

In the light of previous contributions to this area, computationally inferring discourse structures through either “linguistic” supervision [122] or distant supervision [75, 102], this thesis provides the research community with a new set of thoroughly evaluated proposals on how to extract discourse structures from a variety of auxiliary tasks, as well as through self-supervised approaches, to address the data sparsity issue. We contribute clear and diverse evidence of the potentials, strengths and weaknesses of our methodologies, allowing researchers to make an informed decision on the most suitable discourse extraction method.

While we show promising initial results on the ability to capture valid discourse structures from distantly supervised auxiliary tasks and self-supervised systems, the performance of our proposed methods is still limited in two ways: (i) When comparing the performance of our proposed models and methodologies to fully supervised systems on the intra-domain discourse parsing task, our results are not comparable. However, if no human-annotated gold-standard data is available, distantly supervised methods outperform domain-transferred, fully supervised systems. (ii) Our solutions based on different auxiliary tasks capture distinct parts of complete discourse trees well, however, fall short on other properties. As shown in Part I, the auxiliary task of sentiment analysis performs well on short- and mid-range discourse structures, however, does not predict high-level structures and the nuclearity attribute well. The summarization auxiliary task, on the other hand, excels
at the prediction of the nuclearity attribute, but under-performs on the structure generation. With different auxiliary tasks performing well on specific parts of the complete discourse tree, a combination of these tasks according to their strengths and weaknesses is an open problem to be tackled in future work.

Given the fundamental nature of our proposals in this thesis, there is significant potential for downstream applications. The most prominent and impactful application is the automatic generation of out-of-domain discourse structures for any NLP task requiring document-level, structural representations of semantic and pragmatic relationships. As shown in Chapters 3, 8 and 9, our automated discourse inference approach from sentiment analysis (and the resulting MEGA-DT discourse treebank) provides superior discourse signals compared to supervised discourse parsers for domains where no gold-standard annotations are available. Since RST-style discourse structures in the English language are only accessible for a handful of domains, most notably the news and home-repair domains\footnote{The GUM corpus contains 12 different domains, however on average less than 16 samples per domain.}, this results in our novel approach being the best available resource to generate discourse structures for the vast majority of textual data. Further findings presented in Chapters 6 and 7 can potentially lead to more discourse-related approaches in language modelling, either through document-aware transformer models or generally more linguistically inspired methods to model language.

Surrounding our contributions in this thesis, there is a number of open research problems left to be explored in the future. Most significantly, we explore three distantly supervised and two self-supervised approaches to generate robust discourse structures in isolation. While this is a reasonable and necessary step to explore the strengths and weaknesses of individual approaches, the obvious next step is to combine them in a synergistic manner, tailored towards the strengths of competing methods. With the objective to effectively integrate task-dependent and self-supervised discourse trees into a single framework, different approaches could be explored, for example
by (i) simply joining the treebanks generated from different distantly and self-supervised models (i.e., a union of treebanks), or (ii) following a more integrated approach, by generating a single, joint treebank through comparison and rectification of parallel discourse annotations across models (i.e., combine the predictions of different models for a single document).

Orthogonal to this extension, none of the previously explored approaches covers the important relation attribute, describing the relationship between sub-trees (e.g., Contrast, Evidence, ...). Constituting the most difficult component of the complete discourse annotation to predict, the relation attribute is contingent on the structure attribute and tightly connected to the nuclearity explored in this thesis. As a result, integrating the relation attribute is a natural next step. To compute the relation attribute in future work, different sources of information and their potential combinations could be leveraged, such as supervised data, auxiliary-task-specific information, or external knowledge representations (e.g., large-scale language models, knowledge graphs).

Lastly, throughout this thesis, we show the performance of distantly and self-supervised models on “medium-scale” datasets in the English language, such as the Yelp’13 sentiment corpus (Chapters 3 and 7), a 20,000 document subset of Wikipedia (Chapter 4), the individual CNN-DailyMail and New York Times summarization datasets (Chapter 5), as well as multiple mid-size fine-tuning corpora (Chapter 6). As a result, the third direction of future research aims to (i) apply our distantly and self-supervised models to larger data sources, generating even more general, robust and large-scale discourse structures and (ii) extend the investigation to further languages, where distantly and self-supervised methods can be applied, even if no human-annotated discourse treebanks are available in the language.
Bibliography


[34] C. Dyer, A. Kuncoro, M. Ballesteros, and N. A. Smith. Recurrent


208


212


224


[171] W. Xiao, P. Huber, and G. Carenini. Do we really need that many parameters in transformer extractive summarization? discourse can


Appendix A

Discourse Inference from Distant Supervision

A.1 Qualitative Analysis of Generated Discourse Trees from Sentiment Annotations

The following examples are automatically generated trees from our MEGA-DT corpus. EDU leaf nodes are enumerated and can be referenced with the discourse units in the description. The colour-saturation and -hue values represent the sentiment of the nodes, with a dark red (high saturation) representing a strongly negative sub-tree, white (low saturation) representing a neutral sentiment sub-tree and a dark green (high saturation) representing a strongly positive sub-tree. The thickness of edges and the size of nodes represent the attention of the sub-tree, which is strongly correlated with the sub-tree nuclearity.
Figure A.1: Accurately predicted example, Gold-label polarity: 0.5, Predicted polarity: 0.345,
Discourse: [it has been a hell of a work week], [and friday could not have come any sooner.], [this week was rough especially with the announcement], [that we are officially in a recession], [and the ambiguity was hitting me from all sides.], [i need some distraction from all my worries.], [man, did i need some distraction from all my worries.], [so the bf (bacon) and i decided on thai], [but wanted to venture out from the norm], [and we are very glad], [we did.], [the thai hut exceeded our expectations.], [we were a little skeptical at first with so many lackluster reviews], [but we hit the jackpot on our night.], [this place has a great vibe!], [we were seated immediately], [and staff was beyond courteous and attentive.], [we were approached by a few staff members], [which gave us the feeling of true teamwork.], [our server was attentive], [and even sparked up some conversation throughout our meal.], [we started with a hot pot of chicken tom kha kai], [and this soup hit the spot.], [my worries were vanishing with every spoonful.], [they say], [chicken soup will cure a cold], [and that menudo will feed a hangover.], [well, i], [now believe], [tom kha kai is the cure for the blues], [because it sure made me happy!], [for our main dish we shared the red chicken curry], [a little heavy on the red and green peppers but very tasty and was the perfect match with the soup], [so we will definitely be back], [and will be sharing this place with some of our closest friends], [i’ve already made lunch plans for next wednesday].
Figure A.2: Accurately predicted example, Gold-label polarity: −0.5, Predicted polarity: −0.391,
Discourse: [stopped in here for a friday happy hour with co-workers.]1, [the beer was decently]2, [priced for happy hour.]3, [the appetizers were decently priced.]4, [which would be awesome]5, [if they were good.]6, [the chicken strips were terrible.]7, [i have never eaten something so greasy and yet dry all at once.]8, [they are beer battered ( like fish )]9, [which could be good.]10, [but the execution on this was terrible.]11, [the outside was really greasy]12, [which took away all of the crispy goodness]13, [that usually happens]14, [when things are battered and deep fried.]15, [the chicken itself was dry as a bone.]16, [we also got an order of fries]17, [that came out cold]18, [and were just below mediocre.]19, [the place was really warm.]20, [which could be attributed to the summer heat.]21, [but we were sitting inside.]22, [so there is a fair assumption]23, [that air conditioning would be involved.]24, [i’ll pass next time]25, [my coworkers are planning a trip here.]26, [i’d be better off]27, [eating at mcdonald’s.]28
Figure A.3: Inaccurately predicted example, Gold-label polarity: 1, Predicted polarity: −0.098.
Discourse: [upon first moving here 2 years ago.]_1, [i had the worse experience]_2, [attempting to get an airbrush spray tan at this salon.]_3, [they had only 2 people at specific times]_4, [that could spray you custom.]_5, [no problem showed up]_6, [and the tech could not figure out how to use the gun.]_7, [so awkward enough]_8, [him being a male]_9, [and standing there naked, i to get my money back]_10, [after waiting 20 min.]_11, [well a couple months back]_12, [they ran a deal for versa.]_13, [which is a booth spray tan.]_14, [i love this booth.]_15, [it is like airbrushing but private.]_16, [and this spray tan absolutely does not smell or stain your sheets!]_17, [i found this]_18, [upon leaving denver, co. and just]_19, [until i saw it online on living deals for amazon. :) ]_20, [one down two]_21, [to go.]_22, [all for $ 29 :) love !]_23, [as far as the gym goes,]_24, [never used it !]_25
Figure A.4: Inaccurately predicted example, Gold-label polarity: $-0.5$,
Predicted polarity: $0.085$,
Discourse: [i hate having]$_1$, [to write a poor review for this joint!]$_2$, [the owner is a really great guy]$_3$, [and the service was excellent]$_4$, [the place is decorated well]$_5$, [and has a clean finished look]$_6$, [i really wanted to love the pudding]$_7$, [but it really didn’t work out for my wife and i. from first glance]$_8$, [the pudding was all very soupy]$_9$, [and while it tasted]$_{10}$, [okay, was not anything to write home about.]$_{11}$, [the shop is trying too hard]$_{12}$, [to be an ice cream or gelato setup.]$_{13}$, [i think all the flavors and take away from their core business model.]$_{14}$, [i think]$_{15}$, [they should focus on making the rice pudding more solid]$_{16}$, [and have a couple]$_{17}$, [warm pudding options.]$_{18}$, [i can envision a warm rice pudding with some nuts and raisins with some brown sugar or cinnamon on top]$_{19}$, [yum!]$_{20}$, [shoot for rich, creamy and full of flavor.]$_{21}$, [my best wishes go out to them]$_{22}$, [and hope]$_{23}$, [that the masses will enjoy it more than we did.]$_{24}$, [they are good folks]$_{25}$, [and deserve to be successful.]$_{26}$
Figure A.5: Inaccurately predicted example, Gold-label polarity: \(-0.5\), Predicted polarity: \(-0.006\),
Discourse: \([i’ve been here a couple of times in the past.\)]\(^1\), \([usually at someone else’s suggestion.\)]\(^2\), \([i can’t say\)]\(^3\), \([that i recommend this place.\)]\(^4\), \([unless you like\)]\(^5\), \([your lunch served up with a lot of attitude.\)]\(^6\), \([the lady\)]\(^7\), \([that takes the orders at the counter\)]\(^8\), \([is usually abrasive and rude.\)]\(^9\), \([i am the type of person\)]\(^10\), \([who will kill the meanest person with kindness.\)]\(^11\), \([but there are places\)]\(^12\), \([where i draw the line.\)]\(^13\), \([so, i have drawn the line with rome’s pizza.\)]\(^14\), \([the funny part about it all is\)]\(^15\), \([that my line is often zig-zag and curvy.\)]\(^16\), \([so i still go here\)]\(^17\), \([when someone else wants to go.\)]\(^18\), \([hehe.\)]\(^19\), \([my friends like the abuse i guess.\)]\(^20\), \([one friend says\)]\(^21\), \([the lady is nice to him.\)]\(^22\), \([the plus side\)]\(^23\), \([of going here is the fact\)]\(^24\), \([that they serve an average pizza by the slice with your custom toppings.\)]\(^25\), \([they also make hoagies and some other dishes.\)]\(^26\), \([they have a nice lunch special\)]\(^27\), \([that includes soda for a few bucks.\)]\(^28\), \([they also serve some typical american favorites like hot wings.\)]\(^29\), \([i usually order the of pizza lunch special\)]\(^30\), \([and get the unsweetened tea.\)]\(^31\), \([i’m not sure\)]\(^32\), \([why i forget.\)]\(^33\), \([but their tea tastes horrible\)]\(^34\), \([because the water from the fountain tastes terrible !\)]\(^35\), \([but it never fails, i forget\)]\(^36\), \([that i need to of water with me.\)]\(^37\), \([all in all, this place is a dive.\)]\(^38\), \([give it a try.\)]\(^39\)
A.2 Complete Quantitative Results of Generated Discourse Trees from Topic Segmentation Annotations

We show the complete quantitative results, including the previously shown Sentence-to-Paragraph (S-P), Paragraph-to-Document (P-D) and Sentence-to-Document (S-D) performance, as well as the additional EDU-to-Sentence (E-S), EDU-to-Paragraph (E-P), EDU-to-Document (E-D) and EDU-to-Sentence-to-Document (E-S-D) sub-tree results here. Tables A.1 and A.2 contain the RST parseval scores on the RST-DT and GUM corpus, respectively. Table A.3 contains the full results of the original parseval score applied to RST-DT and Table A.4 shows the original parseval performance on GUM. Besides the additional performance evaluations also covering low-level structures for the baseline models (E-S, E-P, E-D and E-S-D), we show an additional model, combining the out-of-the-box discourse segmenter by Wang et al. [161] with our above-sentence tree structures generated from topic-segmentation in the last row of Tables A.1, A.2, A.3 and A.4. Since we combine strictly intra-sentence structures from the discourse segmenter with strictly above-sentence trees, this approach does not support leaky EDUs, and hence constitutes E-S-D structures (as compared to non-sentence limited E-D trees).
<table>
<thead>
<tr>
<th>Model</th>
<th>E-S</th>
<th>S-P</th>
<th>P-D</th>
<th>E-P</th>
<th>S-D</th>
<th>E-D</th>
<th>E-S-D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baselines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random*</td>
<td>70.08</td>
<td>77.11</td>
<td>63.90</td>
<td>64.05</td>
<td>60.20</td>
<td>58.03</td>
<td>70.10</td>
</tr>
<tr>
<td>Right-Branching</td>
<td>64.12</td>
<td>73.57</td>
<td>65.50</td>
<td>58.41</td>
<td>59.46</td>
<td>54.64</td>
<td>74.37</td>
</tr>
<tr>
<td>Left-Branching</td>
<td>63.85</td>
<td>72.41</td>
<td>64.07</td>
<td>57.73</td>
<td>58.07</td>
<td>53.73</td>
<td>70.58</td>
</tr>
<tr>
<td><strong>Supervised RST-style Parsers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top-Down RST-DT</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>86.10</td>
<td>86.40</td>
<td></td>
</tr>
<tr>
<td>Two-Stage GUM</td>
<td>92.57</td>
<td>88.82</td>
<td>65.63</td>
<td>88.62</td>
<td>69.58</td>
<td>81.24</td>
<td>×</td>
</tr>
<tr>
<td>Two-Stage RST-DT</td>
<td>94.48</td>
<td>90.64</td>
<td>68.09</td>
<td>91.06</td>
<td>72.11</td>
<td>83.90</td>
<td>×</td>
</tr>
<tr>
<td>SpanBERT RST-DT</td>
<td>96.27</td>
<td>90.75</td>
<td>76.03</td>
<td>92.87</td>
<td>77.19</td>
<td>87.69</td>
<td>×</td>
</tr>
<tr>
<td><strong>Distantly Supervised RST-style Parsers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xiao et al. [173] CNN/DM</td>
<td>89.95</td>
<td>74.23</td>
<td>66.15</td>
<td>78.19</td>
<td>59.10</td>
<td>60.68</td>
<td>75.13</td>
</tr>
<tr>
<td>Two-Stage MEGA-DT</td>
<td>89.57</td>
<td>85.00</td>
<td>65.50</td>
<td>84.73</td>
<td>66.99</td>
<td>77.90</td>
<td>×</td>
</tr>
<tr>
<td>TS RST-DT</td>
<td>×</td>
<td>84.34</td>
<td>62.52</td>
<td>×</td>
<td>65.96</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS Wiki</td>
<td>×</td>
<td>83.43</td>
<td>69.78</td>
<td>×</td>
<td>68.13</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS Wiki + RST-DT</td>
<td>×</td>
<td>83.84</td>
<td>66.54</td>
<td>×</td>
<td>65.84</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Ablation - TS Wiki</td>
<td>×</td>
<td>83.51</td>
<td>68.61</td>
<td>×</td>
<td>67.47</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS Wiki / Disc-Seg</td>
<td>87.10</td>
<td>83.43</td>
<td>69.78</td>
<td>81.67</td>
<td>68.13</td>
<td>75.56</td>
<td>×</td>
</tr>
</tbody>
</table>

**Table A.1:** Evaluation results using the RST Parseval micro-average precision measure on the RST-DT dataset. Subscripts indicate the training dataset. TS = Topic Segmentation Model. Disc-Seg = Discourse Segmentation Model by Wang et al. [161]. × = Not feasible combination. * = Average performance over 10 runs. Best performance per sub-table underlined, best performance per column bold.
<table>
<thead>
<tr>
<th>Model</th>
<th>E-S</th>
<th>S-P</th>
<th>P-D</th>
<th>E-P</th>
<th>S-D</th>
<th>E-D</th>
<th>E-S-D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baselines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random*</td>
<td>72.16</td>
<td>67.53</td>
<td>60.96</td>
<td>61.60</td>
<td>57.99</td>
<td>57.24</td>
<td>69.93</td>
</tr>
<tr>
<td>Right-Branching</td>
<td>66.30</td>
<td>64.15</td>
<td>72.71</td>
<td>56.23</td>
<td>59.39</td>
<td>54.71</td>
<td>71.58</td>
</tr>
<tr>
<td>Left-Branching</td>
<td>65.96</td>
<td>62.07</td>
<td>54.35</td>
<td>55.08</td>
<td>51.56</td>
<td>50.77</td>
<td>64.57</td>
</tr>
<tr>
<td><strong>Supervised RST-style Parsers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two-StageRST-DT</td>
<td>93.51</td>
<td>74.20</td>
<td>63.29</td>
<td>82.74</td>
<td>63.65</td>
<td>77.16</td>
<td>×</td>
</tr>
<tr>
<td>Two-StageGUM</td>
<td>93.25</td>
<td>76.70</td>
<td><strong>72.94</strong></td>
<td><strong>83.49</strong></td>
<td><strong>68.38</strong></td>
<td><strong>79.04</strong></td>
<td>×</td>
</tr>
<tr>
<td><strong>Distantly Supervised RST-style Parsers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xiao et al. [173]CNN-DM</td>
<td>88.09</td>
<td>67.89</td>
<td>57.80</td>
<td>76.52</td>
<td>53.82</td>
<td>59.28</td>
<td>72.39</td>
</tr>
<tr>
<td>Two-StageMEGA-DT</td>
<td>88.78</td>
<td>73.37</td>
<td>69.88</td>
<td>78.35</td>
<td>64.69</td>
<td><strong>73.89</strong></td>
<td>×</td>
</tr>
<tr>
<td>TS\textsubscript{GUM}</td>
<td>×</td>
<td>72.54</td>
<td>67.60</td>
<td>×</td>
<td>62.79</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS\textsubscript{Wiki}</td>
<td>×</td>
<td><strong>76.98</strong></td>
<td>63.53</td>
<td>×</td>
<td>65.84</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS\textsubscript{Wiki}+GUM</td>
<td>×</td>
<td>74.48</td>
<td>67.29</td>
<td>×</td>
<td>64.69</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Ablation – TS\textsubscript{Wiki}</td>
<td>×</td>
<td>75.94</td>
<td>64.71</td>
<td>×</td>
<td>65.38</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS\textsubscript{Wiki}/Disc-Seg</td>
<td>88.58</td>
<td>76.98</td>
<td>63.53</td>
<td>79.30</td>
<td>65.84</td>
<td>×</td>
<td><strong>73.94</strong></td>
</tr>
</tbody>
</table>

**Table A.2:** Evaluation results using the **RST Parseval** micro-average precision measure on the GUM dataset. Subscripts indicate the training dataset. TS=Topic Segmentation Model. Disc-Seg=Discourse Segmentation Model by Wang et al. [161]. ×=Not feasible combination. *=Average performance over 10 runs. Best performance per sub-table **underlined**, best performance per column **bold**
<table>
<thead>
<tr>
<th>Model</th>
<th>E-S</th>
<th>S-P</th>
<th>P-D</th>
<th>E-P</th>
<th>S-D</th>
<th>E-D</th>
<th>E-S-D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baselines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random*</td>
<td>19.35</td>
<td>22.55</td>
<td>30.61</td>
<td>17.13</td>
<td>21.83</td>
<td>16.06</td>
<td>41.07</td>
</tr>
<tr>
<td>Right-Branching</td>
<td>3.25</td>
<td>7.27</td>
<td>31.62</td>
<td>4.12</td>
<td>19.21</td>
<td>9.27</td>
<td>48.74</td>
</tr>
<tr>
<td>Left-Branching</td>
<td>2.53</td>
<td>3.20</td>
<td>28.79</td>
<td>2.56</td>
<td>16.45</td>
<td>7.45</td>
<td>41.16</td>
</tr>
<tr>
<td><strong>Supervised RST-style Parsers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDParse – TS&lt;sub&gt;RST-DT&lt;/sub&gt;</td>
<td>×</td>
<td>×</td>
<td>40.52</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Two-Stage&lt;sub&gt;GUM&lt;/sub&gt;</td>
<td>79.97</td>
<td>60.76</td>
<td>31.88</td>
<td>73.78</td>
<td>39.38</td>
<td>62.48</td>
<td>×</td>
</tr>
<tr>
<td>Two-Stage&lt;sub&gt;RST-DT&lt;/sub&gt;</td>
<td>85.10</td>
<td>67.44</td>
<td>36.76</td>
<td>79.40</td>
<td>44.54</td>
<td>70.97</td>
<td>×</td>
</tr>
<tr>
<td>SpanBERT&lt;sub&gt;RST-DT&lt;/sub&gt;</td>
<td><strong>90.16</strong></td>
<td><strong>68.69</strong></td>
<td><strong>52.75</strong></td>
<td><strong>83.83</strong></td>
<td><strong>54.37</strong></td>
<td><strong>75.39</strong></td>
<td>×</td>
</tr>
<tr>
<td><strong>Distantly Supervised RST-style Parsers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xiao et al. [173]&lt;sub&gt;CNN-DM&lt;/sub&gt;</td>
<td>62.11</td>
<td>9.59</td>
<td>32.09</td>
<td>49.72</td>
<td>18.49</td>
<td>21.36</td>
<td>49.21</td>
</tr>
<tr>
<td>Two-Stage&lt;sub&gt;MEGA-DT&lt;/sub&gt;</td>
<td>71.87</td>
<td>47.38</td>
<td>31.62</td>
<td>64.81</td>
<td>34.21</td>
<td>55.81</td>
<td>×</td>
</tr>
<tr>
<td>TS&lt;sub&gt;RST-DT&lt;/sub&gt;</td>
<td>×</td>
<td>45.06</td>
<td>28.02</td>
<td>×</td>
<td>32.17</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS&lt;sub&gt;Wiki&lt;/sub&gt;</td>
<td>×</td>
<td>41.86</td>
<td>41.90</td>
<td>×</td>
<td>36.49</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS&lt;sub&gt;Wiki&lt;/sub&gt;+&lt;sub&gt;RST-DT&lt;/sub&gt;</td>
<td>×</td>
<td>43.31</td>
<td>34.96</td>
<td>×</td>
<td>31.93</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Ablation – TS&lt;sub&gt;Wiki&lt;/sub&gt;</td>
<td>×</td>
<td>42.15</td>
<td>39.07</td>
<td>×</td>
<td>35.17</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS&lt;sub&gt;Wiki&lt;/sub&gt;/Disc-Seg</td>
<td>65.22</td>
<td>41.86</td>
<td>41.90</td>
<td>57.74</td>
<td>36.49</td>
<td>×</td>
<td><strong>50.95</strong></td>
</tr>
</tbody>
</table>

Table A.3: Evaluation results using the original Parseval micro-average precision measure on the RST-DT dataset. Subscripts indicate the training dataset. TS=Topic Segmentation Model. Disc-Seg=Discourse Segmentation Model by Wang et al. [161]. ×=Not feasible combination. *=Average performance over 10 runs. Best performance per sub-table underlined, best performance per column bold
<table>
<thead>
<tr>
<th>Model</th>
<th>E-S</th>
<th>S-P</th>
<th>P-D</th>
<th>E-P</th>
<th>S-D</th>
<th>E-D</th>
<th>E-S-D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baselines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random*</td>
<td>18.45</td>
<td>17.83</td>
<td>25.46</td>
<td>15.25</td>
<td>16.71</td>
<td>14.49</td>
<td>43.11</td>
</tr>
<tr>
<td>Right-Branching</td>
<td>1.28</td>
<td>7.35</td>
<td>46.30</td>
<td>3.39</td>
<td>18.78</td>
<td>9.42</td>
<td>43.15</td>
</tr>
<tr>
<td>Left-Branching</td>
<td>0.30</td>
<td>1.97</td>
<td>10.19</td>
<td>0.86</td>
<td>3.11</td>
<td>1.54</td>
<td>29.15</td>
</tr>
<tr>
<td><strong>Supervised RST-style Parsers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two-Stage_RST-DT</td>
<td>81.00</td>
<td>33.69</td>
<td>27.78</td>
<td>61.90</td>
<td>27.53</td>
<td>54.33</td>
<td>×</td>
</tr>
<tr>
<td>Two-Stage_GUM</td>
<td>80.22</td>
<td>39.78</td>
<td></td>
<td>46.76</td>
<td>63.56</td>
<td>36.75</td>
<td>58.08</td>
</tr>
<tr>
<td><strong>Distantly Supervised RST-style Parsers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xiao et al. [173]_CNN-DM</td>
<td>65.72</td>
<td>7.60</td>
<td>16.75</td>
<td>46.69</td>
<td>7.64</td>
<td>18.56</td>
<td>45.97</td>
</tr>
<tr>
<td>Two-Stage_MEGA-DT</td>
<td>67.13</td>
<td>31.36</td>
<td>40.74</td>
<td>52.22</td>
<td>29.49</td>
<td>47.79</td>
<td>×</td>
</tr>
<tr>
<td>TS_GUM</td>
<td>×</td>
<td>29.03</td>
<td>37.50</td>
<td>×</td>
<td>25.58</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS_Wiki</td>
<td>×</td>
<td>40.50</td>
<td>31.02</td>
<td>×</td>
<td>31.68</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS_Wiki + GUM</td>
<td>×</td>
<td>34.05</td>
<td>37.04</td>
<td>×</td>
<td>29.38</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Ablation - TS_Wiki</td>
<td>×</td>
<td>37.81</td>
<td>33.33</td>
<td>×</td>
<td>30.76</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TS_Wiki / Disc-Seg</td>
<td>66.54</td>
<td>40.50</td>
<td>31.02</td>
<td>54.32</td>
<td>31.68</td>
<td>×</td>
<td>47.81</td>
</tr>
</tbody>
</table>

Table A.4: Evaluation results using the original Parseval micro-average precision measure on the GUM dataset. Subscripts indicate the training dataset. TS=Topic Segmentation Model. Disc-Seg=Discourse Segmentation Model by Wang et al. [161]. ×=Not feasible combination. *=Average performance over 10 runs. Best performance per sub-table underlined, best performance per column bold
A.3 Qualitative Analysis of Generated Discourse Trees from Topic Segmentation Annotations on GUM

Figure A.6: Positive example (GUM_bio_jespersen) with 76.92% structural overlap between prediction (left) and GUM gold-label annotation (right).

Figure A.7: Random example (GUM_voyage_oakland) with 70.83% structural overlap between prediction (left) and GUM gold-label annotation (right).

Figure A.8: Negative example (GUM_bio_dvorak) with 57.14% structural overlap between prediction (left) and GUM gold-label annotation (right).
## A.4 Detailed Performance of the Discourse Augmented Summarization Task

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Layer</th>
<th>#Head</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNNDM</td>
<td>2</td>
<td>1</td>
<td>40.92</td>
<td>18.69</td>
<td>37.85</td>
</tr>
<tr>
<td>CNNDM</td>
<td>2</td>
<td>8</td>
<td>41.02</td>
<td>18.78</td>
<td>37.96</td>
</tr>
<tr>
<td>CNNDM</td>
<td>6</td>
<td>8</td>
<td>41.03</td>
<td>18.69</td>
<td>37.86</td>
</tr>
<tr>
<td>NYT</td>
<td>2</td>
<td>1</td>
<td>43.64</td>
<td>25.58</td>
<td>36.87</td>
</tr>
<tr>
<td>NYT</td>
<td>2</td>
<td>8</td>
<td>44.11</td>
<td>26.08</td>
<td>37.34</td>
</tr>
<tr>
<td>NYT</td>
<td>6</td>
<td>8</td>
<td>43.93</td>
<td>25.99</td>
<td>37.15</td>
</tr>
</tbody>
</table>

**Table A.5:** In-domain performance of the summarizers on the Rouge-1, Rouge-2 and Rouge-L scores.
A.5 Full Results on Discourse Inference from Summarization Annotations

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSTDT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNNDM-2-1</td>
<td>61.16</td>
<td>76.23</td>
<td>74.63</td>
<td>23.65</td>
<td>18.23</td>
<td>21.56</td>
<td>29.29</td>
<td>19.56</td>
<td></td>
</tr>
<tr>
<td>CNNDM-2-8</td>
<td>62.65</td>
<td>76.42</td>
<td>74.28</td>
<td>22.09</td>
<td>21.29</td>
<td>20.31</td>
<td>26.57</td>
<td>22.67</td>
<td></td>
</tr>
<tr>
<td>CNNDM-6-8</td>
<td>60.33</td>
<td>75.44</td>
<td>75.04</td>
<td>7.89</td>
<td>13.83</td>
<td>7.28</td>
<td>16.10</td>
<td>25.50</td>
<td></td>
</tr>
<tr>
<td>NYT-2-1</td>
<td>60.27</td>
<td>75.57</td>
<td>75.29</td>
<td>9.76</td>
<td>23.15</td>
<td>6.18</td>
<td>21.06</td>
<td>21.73</td>
<td></td>
</tr>
<tr>
<td>NYT-2-8</td>
<td><strong>63.20</strong></td>
<td>76.63</td>
<td>75.23</td>
<td>7.35</td>
<td>16.04</td>
<td>6.44</td>
<td>16.72</td>
<td>22.90</td>
<td></td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>62.42</td>
<td>75.58</td>
<td>75.58</td>
<td>15.74</td>
<td>18.90</td>
<td>13.71</td>
<td>25.04</td>
<td>21.14</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>58.60</td>
<td>74.10</td>
<td>74.10</td>
<td>11.16</td>
<td>20.28</td>
<td>1.67</td>
<td>18.72</td>
<td>11.14</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CNNDM-2-1</td>
<td>61.06</td>
<td>71.39</td>
<td>70.29</td>
<td><strong>31.07</strong></td>
<td>4.39</td>
<td><strong>29.33</strong></td>
<td>13.45</td>
<td>28.06</td>
<td>37.38</td>
</tr>
<tr>
<td>CNNDM-2-8</td>
<td>61.44</td>
<td>71.13</td>
<td>71.09</td>
<td>26.98</td>
<td>8.89</td>
<td>24.72</td>
<td>14.75</td>
<td>24.56</td>
<td>28.70</td>
</tr>
<tr>
<td>CNNDM-6-8</td>
<td>60.32</td>
<td>71.24</td>
<td>70.88</td>
<td>8.53</td>
<td>19.51</td>
<td>9.93</td>
<td>21.96</td>
<td>6.92</td>
<td>15.85</td>
</tr>
<tr>
<td>NYT-2-1</td>
<td>60.31</td>
<td>71.40</td>
<td><strong>71.43</strong></td>
<td>10.67</td>
<td>21.15</td>
<td>7.82</td>
<td>17.59</td>
<td>21.30</td>
<td>24.54</td>
</tr>
<tr>
<td>NYT-2-8</td>
<td>61.27</td>
<td>70.80</td>
<td>70.90</td>
<td>6.26</td>
<td>12.59</td>
<td>13.64</td>
<td>19.34</td>
<td>5.25</td>
<td>7.96</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td><strong>61.32</strong></td>
<td>71.30</td>
<td>70.03</td>
<td>16.22</td>
<td>12.14</td>
<td>22.79</td>
<td>16.37</td>
<td>14.81</td>
<td>9.81</td>
</tr>
<tr>
<td>Random</td>
<td>59.49</td>
<td>70.53</td>
<td>70.13</td>
<td>13.14</td>
<td>19.31</td>
<td>2.94</td>
<td>17.88</td>
<td>17.42</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GUM</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CNNDM-2-1</td>
<td>58.74</td>
<td>72.73</td>
<td>71.92</td>
<td><strong>21.28</strong></td>
<td>2.24</td>
<td><strong>27.26</strong></td>
<td>16.12</td>
<td>19.50</td>
<td>28.77</td>
</tr>
<tr>
<td>CNNDM-2-8</td>
<td>59.98</td>
<td>72.69</td>
<td>71.95</td>
<td>19.45</td>
<td>4.98</td>
<td>25.00</td>
<td>19.25</td>
<td>18.63</td>
<td>29.72</td>
</tr>
<tr>
<td>CNNDM-6-8</td>
<td>58.92</td>
<td>72.40</td>
<td>72.69</td>
<td>4.74</td>
<td>15.80</td>
<td>11.53</td>
<td>24.79</td>
<td>4.01</td>
<td>13.14</td>
</tr>
<tr>
<td>NYT-2-1</td>
<td>57.81</td>
<td>71.95</td>
<td>72.23</td>
<td>5.64</td>
<td>12.84</td>
<td>19.94</td>
<td>20.19</td>
<td>2.92</td>
<td>9.79</td>
</tr>
<tr>
<td>NYT-2-8</td>
<td><strong>60.17</strong></td>
<td>71.98</td>
<td>71.82</td>
<td>5.66</td>
<td>7.22</td>
<td>15.21</td>
<td>18.81</td>
<td>4.54</td>
<td>3.96</td>
</tr>
<tr>
<td>NYT-6-8</td>
<td>59.65</td>
<td>72.19</td>
<td>71.56</td>
<td>12.58</td>
<td>8.61</td>
<td>23.35</td>
<td>17.14</td>
<td>10.67</td>
<td>8.23</td>
</tr>
<tr>
<td>Random</td>
<td>57.47</td>
<td>71.50</td>
<td>71.50</td>
<td>10.37</td>
<td>9.26</td>
<td>19.15</td>
<td>0.92</td>
<td>0.05</td>
<td>17.01</td>
</tr>
</tbody>
</table>

Table A.6: The RST Parseval Scores of generated constituency trees and Unlabeled Attachment Score of generated dependency trees by the Eisner and CLE algorithms on the three datasets. The numbers in each cell are represented as the performance of (Layer 0 / Layer 1). Random results are obtained by applying the parser on randomly generated matrices 10 times, and are represented as “Average (Std)”.
A.6 Results on Sensitivity to Initialization for Discourse Inference from Summarization Annotations

To explore if the models with different random initialization have consistent performances, we train 5 models with 6 layers and 8 heads on the CNNDM dataset with different initialization. The results of each layer for constituency/dependency parsing are shown in Table A.7. Additional explorations on the performance of all heads are shown in Figure A.9. The dependency parsing task at different training checkpoints of the summarization model is shown in Figure A.10. Lastly, we show heatmaps for the average UAS across three datasets and all heads in Figure A.11.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RSTDT</td>
<td>61.13 (1.11)</td>
<td>75.81 (0.26)</td>
<td>10.32 (4.03)</td>
<td>9.40 (3.62)</td>
<td>18.89 (4.19)</td>
<td>18.89 / 28.33</td>
</tr>
<tr>
<td>Random</td>
<td>58.60 (0.1)</td>
<td>74.10 (0.20)</td>
<td>11.16 (2.80)</td>
<td>1.67 (1.37)</td>
<td>18.72 (1.59)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction</th>
<th>CNNDM-6-8</th>
<th>61.87 (1.17)</th>
<th>70.84 (0.51)</th>
<th>11.50 (5.71)</th>
<th>9.79 (5.31)</th>
<th>17.81 / 26.30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>59.49 (0.3)</td>
<td>70.53 (0.1)</td>
<td>13.14 (0.33)</td>
<td>19.31 (4.44)</td>
<td>2.94 (0.24)</td>
<td>17.88 (0.42)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GUM</th>
<th>CNNDM-6-8</th>
<th>58.19 (0.82)</th>
<th>72.28 (0.27)</th>
<th>7.62 (2.87)</th>
<th>6.77 (3.96)</th>
<th>17.32 / 25.13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>57.47 (0.1)</td>
<td>71.50 (0.2)</td>
<td>10.37 (0.23)</td>
<td>19.15 (0.26)</td>
<td>0.92 (0.05)</td>
<td>17.01 (0.2)</td>
</tr>
</tbody>
</table>

Table A.7: The average RST Parseval Scores of generated constituency trees and average Unlabeled Attachment Scores of generated dependency trees using the Eisner and CLE algorithms, on the three datasets with 5 random initialization. Numbers in parenthesis are the standard deviation across different runs.
Figure A.9: The heatmap of the average UAS across three discourse datasets for all attention heads in the models with different initialization.

Figure A.10: Max and Mean UAS for dependency trees generated by the CLE algorithm on all attention heads (48) after training for (0,1k,5k,10k,20k,23k) steps on the RST-DT (top), Instructional (middle) and GUM (bottom) datasets. The corresponding ROUGE scores are increasing.
Figure A.11: Heatmaps of the average UAS across the three discourse datasets for all heads during training of the summarization model.
Appendix B

Discourse Inference from Self-Supervision

B.1 Huggingface Models for Discourse Inference from Pre-Trained Language Models

We investigate 7 fine-tuned BERT and BART models from the Huggingface model library, as well as the two pre-trained models in Chapter 6. The model names and links are provided in Table B.1.

<table>
<thead>
<tr>
<th>Pre-Trained</th>
<th>Fine-Tuned</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-base IMDB</td>
<td><a href="https://huggingface.co/textattack/bert-base-uncased-imdb">https://huggingface.co/textattack/bert-base-uncased-imdb</a></td>
<td></td>
</tr>
<tr>
<td>BERT-base Yelp</td>
<td><a href="https://huggingface.co/briceyhc/bert-base-uncased-yelp_polarity">https://huggingface.co/briceyhc/bert-base-uncased-yelp_polarity</a></td>
<td></td>
</tr>
<tr>
<td>BERT-base MNLI</td>
<td><a href="https://huggingface.co/textattack/bert-base-uncased-MNLI">https://huggingface.co/textattack/bert-base-uncased-MNLI</a></td>
<td></td>
</tr>
<tr>
<td>BART-large CNN-DM</td>
<td><a href="https://huggingface.co/facebook/bart-large-cnn">https://huggingface.co/facebook/bart-large-cnn</a></td>
<td></td>
</tr>
<tr>
<td>BART-large XSUM</td>
<td><a href="https://huggingface.co/facebook/bart-large-xsum">https://huggingface.co/facebook/bart-large-xsum</a></td>
<td></td>
</tr>
<tr>
<td>BART-large SQuAD</td>
<td><a href="https://huggingface.co/valhalla/bart-large-finetuned-squadv1">https://huggingface.co/valhalla/bart-large-finetuned-squadv1</a></td>
<td></td>
</tr>
</tbody>
</table>

Table B.1: Huggingface pre-trained and fine-tuned model links.
B.2 PLM Discourse Inference Test Set Results on RST-DT and GUM

Figure B.1: Constituency (top) and dependency (bottom) discourse tree evaluation of BERT (a) and BART (b) models on RST-DT (test). Purple=high score, blue=low score. + indicates fine-tuning dataset.
(a) BERT: PLM, +IMDB, +Yelp, +MNLI, +SST-2

(b) BART: PLM, +CNN-DM, +XSUM, +SQuAD

**Figure B.2:** Constituency (top) and dependency (bottom) discourse tree evaluation of BERT (a) and BART (b) models on GUM (test). Purple=high score, blue=low score. + indicates fine-tuning dataset.
### B.3 Oracle-picked Self-Attention Head Compared to the Validation-Picked Matrix for Discourse Inference from PLMs

<table>
<thead>
<tr>
<th>Model</th>
<th>RST-DT Span</th>
<th>UAS</th>
<th>GUM Span</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>rand. init</td>
<td>25.5 (-0.0)</td>
<td>13.3 (-0.0)</td>
<td>23.2 (-0.0)</td>
<td>12.4 (-0.0)</td>
</tr>
<tr>
<td>PLM</td>
<td>35.7 (-1.6)</td>
<td>45.3 (-4.9)</td>
<td>33.0 (-0.4)</td>
<td>45.2 (-0.0)</td>
</tr>
<tr>
<td>+ IMDB</td>
<td>35.4 (-1.8)</td>
<td>42.8 (-2.4)</td>
<td>33.0 (-3.8)</td>
<td>43.3 (-0.1)</td>
</tr>
<tr>
<td>+ Yelp</td>
<td>34.7 (-1.0)</td>
<td>42.3 (-1.9)</td>
<td>32.6 (-3.6)</td>
<td>43.7 (-0.0)</td>
</tr>
<tr>
<td>+ SST-2</td>
<td>35.5 (-1.9)</td>
<td>42.9 (-2.5)</td>
<td>32.6 (-0.3)</td>
<td>43.5 (-0.9)</td>
</tr>
<tr>
<td>+ MNLI</td>
<td>34.8 (-1.7)</td>
<td>41.8 (-1.4)</td>
<td>32.4 (-0.3)</td>
<td>43.3 (-0.5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>RST-DT Span</th>
<th>UAS</th>
<th>GUM Span</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>rand. init</td>
<td>25.3 (-0.0)</td>
<td>12.5 (-0.0)</td>
<td>23.2 (-0.0)</td>
<td>12.2 (-0.0)</td>
</tr>
<tr>
<td>PLM</td>
<td>39.1 (-0.4)</td>
<td>41.7 (-2.7)</td>
<td>31.8 (-0.3)</td>
<td>41.8 (-0.0)</td>
</tr>
<tr>
<td>+ CNN-DM</td>
<td>40.9 (-0.0)</td>
<td>44.3 (-4.0)</td>
<td>32.7 (-0.3)</td>
<td>42.8 (-0.7)</td>
</tr>
<tr>
<td>+ XSUM</td>
<td>40.1 (-0.9)</td>
<td>41.9 (-3.4)</td>
<td>32.1 (-1.7)</td>
<td>39.9 (-0.0)</td>
</tr>
<tr>
<td>+ SQuAD</td>
<td>40.1 (-0.0)</td>
<td>43.2 (-4.6)</td>
<td>31.3 (-2.1)</td>
<td>40.7 (-0.1)</td>
</tr>
</tbody>
</table>

#### Baselines

<table>
<thead>
<tr>
<th>Model</th>
<th>RST-DT Span</th>
<th>UAS</th>
<th>GUM Span</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right-Branch/Chain</td>
<td>9.3</td>
<td>40.4</td>
<td>9.4</td>
<td>41.7</td>
</tr>
<tr>
<td>Left-Branch/Chain⁻¹</td>
<td>7.5</td>
<td>12.7</td>
<td>1.5</td>
<td>12.2</td>
</tr>
<tr>
<td>SumCNN-DM[173]</td>
<td>21.4</td>
<td>20.5</td>
<td>17.6</td>
<td>15.8</td>
</tr>
<tr>
<td>SumNYT[173]</td>
<td>24.0</td>
<td>15.7</td>
<td>18.2</td>
<td>12.6</td>
</tr>
<tr>
<td>Two-StageRST-DT[160]</td>
<td>72.0</td>
<td>71.2</td>
<td>54.0</td>
<td>54.5</td>
</tr>
<tr>
<td>Two-StageGUM</td>
<td>65.4</td>
<td>61.7</td>
<td>58.6</td>
<td>56.7</td>
</tr>
</tbody>
</table>

**Table B.2:** Original parseval (span) and Unlabelled Attachment Score (UAS) of the single best-performing oracle self-attention matrix and validation set picked head (in brackets) of the BERT and BART models compared with baselines and previous work. “rand. init” = Randomly initialized transformer model of similar architecture as the PLM.
### B.4 Detailed Self-Attention Statistics of Discourse Inference PLMs

<table>
<thead>
<tr>
<th>Model</th>
<th>Span</th>
<th>Eisner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Med</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RST-DT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rand. init</td>
<td>21.7</td>
<td>23.4</td>
</tr>
<tr>
<td>PLM</td>
<td>19.3</td>
<td>27.0</td>
</tr>
<tr>
<td>+ IMDB</td>
<td>19.7</td>
<td>26.9</td>
</tr>
<tr>
<td>+ YELP</td>
<td>20.2</td>
<td>26.6</td>
</tr>
<tr>
<td>+ SST-2</td>
<td>19.5</td>
<td>27.3</td>
</tr>
<tr>
<td>+ MNLI</td>
<td>18.5</td>
<td>26.9</td>
</tr>
<tr>
<td>GUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rand. init</td>
<td>18.6</td>
<td>21.0</td>
</tr>
<tr>
<td>PLM</td>
<td>17.8</td>
<td>24.2</td>
</tr>
<tr>
<td>+ IMDB</td>
<td>18.1</td>
<td>23.8</td>
</tr>
<tr>
<td>+ YELP</td>
<td>18.6</td>
<td>24.0</td>
</tr>
<tr>
<td>+ SST-2</td>
<td>18.2</td>
<td>24.6</td>
</tr>
<tr>
<td>+ MNLI</td>
<td>17.4</td>
<td>23.9</td>
</tr>
</tbody>
</table>

**Table B.3:** Minimum, median, mean and maximum performance of the self-attention matrices on RST-DT and GUM for the BERT model.
<table>
<thead>
<tr>
<th>Model</th>
<th>Span</th>
<th></th>
<th></th>
<th></th>
<th>Eisner</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Med</td>
<td>Mean</td>
<td>Max</td>
<td>Min</td>
<td>Med</td>
<td>Mean</td>
<td>Max</td>
</tr>
<tr>
<td>RST-DT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rand. init</td>
<td>20.3</td>
<td>23.3</td>
<td>23.3</td>
<td>25.3</td>
<td>8.5</td>
<td>10.6</td>
<td>10.6</td>
<td>12.5</td>
</tr>
<tr>
<td>PLM</td>
<td>20.3</td>
<td>28.3</td>
<td>28.5</td>
<td>39.1</td>
<td>4.1</td>
<td>15.8</td>
<td>19.2</td>
<td>41.7</td>
</tr>
<tr>
<td>+ CNN-DM</td>
<td>20.5</td>
<td>28.6</td>
<td>28.7</td>
<td>40.9</td>
<td>3.6</td>
<td>15.2</td>
<td>19.2</td>
<td>44.3</td>
</tr>
<tr>
<td>+ XSUM</td>
<td>20.2</td>
<td>27.6</td>
<td>28.3</td>
<td>40.1</td>
<td>4.8</td>
<td>14.8</td>
<td>18.7</td>
<td>41.9</td>
</tr>
<tr>
<td>+ SQuAD</td>
<td>20.5</td>
<td>27.6</td>
<td>28.2</td>
<td>40.1</td>
<td>2.8</td>
<td>14.8</td>
<td>18.8</td>
<td>43.2</td>
</tr>
<tr>
<td>GUM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rand. init</td>
<td>18.6</td>
<td>21.0</td>
<td>21.0</td>
<td>23.2</td>
<td>8.0</td>
<td>10.2</td>
<td>10.2</td>
<td>12.2</td>
</tr>
<tr>
<td>PLM</td>
<td>16.7</td>
<td>23.4</td>
<td>23.8</td>
<td>31.5</td>
<td>2.6</td>
<td>15.2</td>
<td>18.7</td>
<td>41.8</td>
</tr>
<tr>
<td>+ CNN-DM</td>
<td>15.9</td>
<td>23.7</td>
<td>24.1</td>
<td>32.4</td>
<td>3.7</td>
<td>14.7</td>
<td>18.9</td>
<td>42.8</td>
</tr>
<tr>
<td>+ XSUM</td>
<td>16.4</td>
<td>23.2</td>
<td>23.9</td>
<td>31.8</td>
<td>3.0</td>
<td>14.1</td>
<td>18.1</td>
<td>39.9</td>
</tr>
<tr>
<td>+ SQuAD</td>
<td>16.1</td>
<td>23.4</td>
<td>23.8</td>
<td>31.0</td>
<td>2.4</td>
<td>14.8</td>
<td>18.3</td>
<td>40.7</td>
</tr>
</tbody>
</table>

**Table B.4:** Minimum, median, mean and maximum performance of the self-attention matrices on RST-DT and GUM for the BART model.
B.5 Details of Structural Discourse Similarity of Discourse Structures from PLMs

Figure B.3: Detailed PLM discourse constituency (left) and dependency (right) structure overlap with baselines and gold trees according to the original parseval and UAS metrics.
Figure B.4: Detailed PLM discourse constituency (left) and dependency (right) structure performance of intersection with gold trees (e.g., BERT ∩ Gold Trees ↔ Two-Stage (RST-DT) ∩ Gold Trees) according to the original parseval and UAS metrics.
B.6 Intra-/Inter-Model Self-Attention for Discourse Structures from PLMs

![Heatmaps by heads (left) and models (right)](image)

(a) BERT constituency tree similarity on GUM

![Heatmaps by heads (left) and models (right)](image)

(b) BERT dependency tree similarity on GUM

![Heatmaps by heads (left) and models (right)](image)

(c) BART constituency tree similarity on GUM

![Heatmaps by heads (left) and models (right)](image)

(d) BART dependency tree similarity on GUM

**Figure B.5:** Top: Visual analysis of sorted heatmaps. Yellow=high score, purple=low score. Bottom: Aggregated similarity of same heads, same models, different heads and different models. *Head/**Model significantly better than **Head//Model performance with p-value < 0.05.
## B.7 Representational Consistency of Discourse Inference from Tree-Style AutoEncoders

<table>
<thead>
<tr>
<th>Document</th>
<th>Horrible service, they keep delaying the delivery of our order, they say they arrived and called with no answer to buy more time, when called to complain got hang up on, do not order from here. PS: I’m not an ex-employee.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similar-1</td>
<td>I used to love this place but it’s cash only now. So inconvenient that if I want bahama bucks I drive further to go to one that accepts credit/debit.</td>
</tr>
<tr>
<td>Similar-2</td>
<td>Vito’s is pretty good food, not the top on my list but my husband and I will go here for a special occasion from time to time when we cant get in elsewhere and it does the job.</td>
</tr>
<tr>
<td>Similar-3</td>
<td>Love this place! Great healthy choices and awesome desert. Way to go in healthy choices!</td>
</tr>
<tr>
<td>Different-1</td>
<td>Love the Happy Hour at Brio’s. Great food, drinks and people to meet. The staff is very attentive and want to ensure you enjoy your time there.</td>
</tr>
<tr>
<td>Different-2</td>
<td>I love the Gyro’s pita and I like the service, so I will be back to try some other things in the Menu. Good place!</td>
</tr>
<tr>
<td>Different-3</td>
<td>Great menu of food, you really can’t go wrong with any choice. Love how the chef can bring the spice. Definitely one of the better Asian restaurants in Scottsdale.</td>
</tr>
</tbody>
</table>

Table B.5: Representationally similar/different document-encodings based on the cosine similarity.
<table>
<thead>
<tr>
<th>Document</th>
<th>5 major health code violations. It’s a shame because I really liked this place and the food was pretty good, but I won’t be eating there again.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similar-1</td>
<td>Food is alright. Service is garbage. You would expect a lot more from a place with the kind of prices they have. If anything, the food is what gave it the 2 stars.</td>
</tr>
<tr>
<td>Similar-2</td>
<td>Had a not-so-great experience at US-egg. My veggie frittata was really soggy, like a veggie frittata soup. The frozen vegetables must have thawed in the dish.</td>
</tr>
<tr>
<td>Similar-3</td>
<td>Very cool, great music. Study room but beware of the smoke cloud surrounding the outside. Can’t sit outside unless you want to leave smelling like smoke.</td>
</tr>
<tr>
<td>Different-1</td>
<td>First time trying out this place and good thing I did. Good, helpful service and a quick turn around. There was a 50 cent debit charge but was waived off for being a first timer. Pretty awesome, not that many businesses would do that.</td>
</tr>
<tr>
<td>Different-2</td>
<td>Overall, it is a department store. You have helpful employees usually however when it comes to sale time, be ready to get in and get out. The customer-to-employee ratio is not the best and to achieve any assistance is nearly impossible.</td>
</tr>
<tr>
<td>Different-3</td>
<td>Beef ribs were perfectly spicy and sweet. The Indian fried bread and biscuits with whipped cinnamon butter are great. I loved the whipped butter served room temperature.</td>
</tr>
</tbody>
</table>

Table B.6: Representationally similar/different document-encodings based on the cosine similarity.
<table>
<thead>
<tr>
<th>Document</th>
<th>Ask them to wear a hair net when they make your sandwich. I used to eat here a lot till I got hair in 3 sandwiches. Two were on the same day. Hell no will I go back.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similar-1</td>
<td>Worst experience ever! Salsa was spoiled and my chicken taco was disgusting. Even the little I did eat I got food poisoning. Will never return to this restaurant.</td>
</tr>
<tr>
<td>Similar-2</td>
<td>Service was marginal and food was not very good. I remember this place being better, but I guess they changed their Menu around a bit ago. Let’s just say I like the old one better. I will pass on this place moving forward.</td>
</tr>
<tr>
<td>Similar-3</td>
<td>It’s a fast food joint, that being said everything here went off without a hitch. They even brought our order to us at the table. That’s something I’ve never seen done by this chain. I hope they keep up the good work.</td>
</tr>
<tr>
<td>Different-1</td>
<td>Finally a chop shop in Tempe! The set up here is adorable and they have great healthy food. They’ve added a few new juices to the menu which I’m excited to try. Will definitely be here all the time!</td>
</tr>
<tr>
<td>Different-2</td>
<td>Perfect early Thai dinner. Spicy mussels. Tad pad with shrimp and chicken, perfect fit for a early dinner. No rush crowd here if you arrive a week night early dinner good food, great price!</td>
</tr>
<tr>
<td>Different-3</td>
<td>Good wholesome BBQ, but they were out of collard greens: the cornbread muffins weren’t the best either, but the brisket and cowboy beans are worth coming for.</td>
</tr>
</tbody>
</table>

**Table B.7:** Representationally similar/different document-encodings based on the cosine similarity.
B.8 Tree-Style AutoEncoder Qualitative Discourse Tree Comparison

Figure B.6: Our generated tree (left) compared to the gold-standard tree (right) for document *wsj_1395*

Figure B.7: Our generated tree (left) compared to the gold-standard tree (right) for document *wsj_1198*

Figure B.8: Our generated tree (left) compared to the gold-standard tree (right) for document *wsj_1998*
Appendix C

Discourse Application

C.1 Numeric Results of the Weighted Discourse Approach in Comparison to Threshold Nuclearity

The numeric results of our W-RST approach for the sentiment analysis and summarization downstream tasks presented in Figure 10.6 are shown in Table C.1 below, along with the threshold-based approach, as well as the supervised parser.
Approach | Sentiment Accuracy | Summarization | R-1 | R-2 | R-L
---|---|---|---|---|---
Nuclearity with Threshold

| t | 53.76 | 28.22 | 8.58 | 26.45 |
| t = 0.1 | 53.93 | 28.41 | 8.69 | 26.61 |
| t = 0.2 | 54.13 | 28.64 | 8.85 | 26.83 |
| t = 0.3 | 54.33 | 28.96 | 9.08 | 27.14 |
| t = 0.4 | 54.44 | 29.36 | 9.34 | 27.51 |
| t = 0.5 | 54.79 | 29.55 | 9.50 | 27.68 |
| t = 0.6 | 54.99 | 29.78 | 9.65 | 27.90 |
| t = 0.7 | 55.07 | 29.57 | 9.45 | 27.74 |
| t = 0.8 | 55.32 | 29.18 | 9.08 | 27.32 |
| t = 0.9 | 54.90 | 28.11 | 8.29 | 26.35 |
| t = 1.0 | 54.15 | 26.94 | 7.60 | 25.27 |

Our Weighted RST Framework

| weighted | 54.76 | 29.70 | 9.58 | 27.85 |

Supervised Training on RST-DT

| supervised | 44.77 | 34.20 | 12.77 | 32.09 |

**Table C.1:** Results of the W-RST approach compared to threshold-based nuclearity assignments and supervised training on RST-DT.