Multi-view and multi-task training of RST discourse parsers

Chloé Braud
CoAStaL
Dep. of Computer Science
University of Copenhagen
braud@di.ku.dk

Barbara Plank
Computational Linguistics
CLCG
University of Groningen
b.plank@rug.nl

Anders Søgaard
CoAStaL
Dep. of Computer Science
University of Copenhagen
soegaard@di.ku.dk

Abstract

We experiment with different ways of training LSTM networks to predict RST discourse trees. The main challenge for RST discourse parsing is the limited amounts of training data. We combat this by regularizing our models using task supervision from related tasks as well as alternative views on discourse structures. We show that a simple LSTM sequential discourse parser takes advantage of this multi-view and multi-task framework with 12-15% error reductions over our baseline (depending on the metric) and results that rival more complex state-of-the-art parsers.

1 Introduction

Documents are not just an arbitrary collection of text spans, but rather an ordered list of structures forming a discourse. Discourse structures describe the organization of documents in terms of discourse or rhetorical relations. For instance, the discourse relation CONDITION holds between the two discourse units (marked with square brackets) in example (1a) and a relation MANNER-MEANS holds between the segments in example (1b).[1]

(1) a. [The gain on the sale couldn't be estimated] [until the "tax treatment has been determined."]

b. [On Friday, Datuk Daim added spice to an otherwise unremarkable address on Malaysia's proposed budget for 1990] [by ordering the Kuala Lumpur Stock Exchange "to take appropriate action immediately" to cut its links with the Stock Exchange of Singapore.]

Different theories of discourse structure exist. For instance, Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) analyzes texts as constituency trees covering entire documents. This theory has led to the RST Discourse Treebank (RST-DT) (Carlson et al., 2001) for English and the development of text-level discourse parsers (Hernault et al., 2010; Joty et al., 2012; Feng and Hirst, 2014; Ji and Eisenstein, 2014b). Such parsers have proven to be useful for several downstream applications (Taboada and Mann, 2006; Daumé III and Marcu, 2009; Thione et al., 2004; Sporleder and Lapata, 2005; Louis et al., 2010; Bhatia et al., 2015; Burstein et al., 2003; Higgins et al., 2004). Another corpus has been annotated for discourse phenomena in English, the Penn Discourse Treebank (Prasad et al., 2008) (PDTB). In contrast to RST-DT, PDTB does not encode discourse as tree structures and documents are not fully covered. In this study we focus on the RST-DT, but among other things, we consider the question of whether the information in PDTB can be used to improve RST discourse parsers.

Discourse parsing is known to be a hard task (Stede, 2011). It involves several complex and interacting factors, touching upon all layers of linguistic analysis, from syntax, semantics up to pragmatics. Consequently, also annotation is complex and time consuming, and hence available annotated corpora are sparse and limited in size. The aim of this paper is to address this training data sparsity by proposing to leverage different views of the same data as well as information from related auxiliary tasks. We aim at investigating which source of information are relevant for the discourse parsing task.

[1] The examples are taken from the RST Discourse Treebank, documents 1179 and 0613, respectively.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1903–1913, Osaka, Japan, December 11-17 2016.

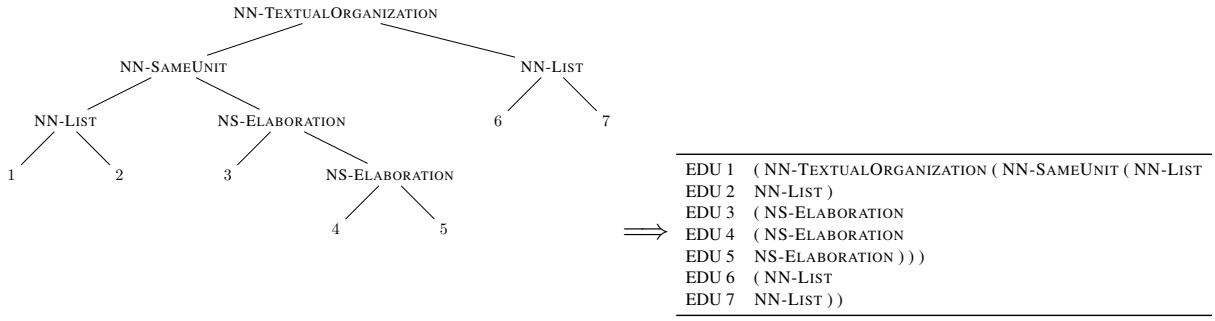

Figure 1: From RST-DT discourse trees to constituency sequence labels.

Specifically, we draw upon the recent success of deep learning methods and present a novel multi-view multi-task hierarchical deep learning model for discourse parsing. Our model, a bidirectional Long Short-Term Memory (bi-LSTM) model, learns joint text segment representations for predicting RST-DT discourse trees and learns from several auxiliary tasks. We encode RST-DT trees as sequences of bracket and label n-grams, and exploit multiple views of the data (such as RST-DT structures as dependencies) as well as multi-task learning through auxiliary tasks (such as modality information from TimeBank, or discourse relations as annotated in the PDTB). Our multi-view learning is different from the standard notion of multi-view learning. Jin et al. (2013), for example, combine multi-view and multi-task learning, but here, multiple views refer to multiple, independent feature sets describing the datapoints. We are, to the best of our knowledge, the first to use multiple views *on the output structures* to effectively regularize learning.

Contributions We present a hierarchical multi-task bi-LSTM architecture for multi-task learning, enabling better learning of discourse parsers with other views of the data and related tasks. Our approach achieves competitive performance compared to previous state-of-the-art models by making use of auxiliary tasks. We make the code and preprocessing scripts available for download at `http://bitbucket.org/chloebt/discourse`.

2 Baseline RST parser

Discourse parsing is a prediction problem where the input is a document, i.e., a sequence of elementary discourse units (EDUs) consisting of text fragments. The output of the task is a binary tree[2] with EDUs at the leaf nodes. The non-terminal nodes are labeled with two sets of information: (a) discourse relations and (b) an indication of whether the daughters are nucleus or satellite. A nucleus is being considered as the most important part of the text whereas a satellite presents secondary information. A discourse relation may involve a nucleus and a satellite (mononuclear relation) or two nuclei (multinuclear relation).

2.1 From sequences to trees

Our approach is to learn sequential models with transfer from models from related tasks, but the output structures are trees. For this purpose, we encode trees as sequences in a very simple way that preserves all the information from the original trees: Every EDU is labeled with its local surrounding discourse structure. More precisely, the first EDU is labeled by the entire path of the root of the tree to itself. Then, the following EDUs are labeled as beginning a new relation (using the opening bracket and the relation name) or ending one or more relations (using the relation name and closing brackets). For example, the EDU 1 in the tree in Figure 1 will be labeled with: "NS-TEXTUALORGANIZATION (NS-SAMEUNIT (NN-LIST".[3] Then, the EDU 2 ends a LIST relation, and the EDU 3 begins an ELABORATION relation. See Figure 1 for a complete conversion.

[2] As in all the previous studies on the RST-DT, we binarize the trees using right-branching.

[3] 'N' means nucleus and 'S' satellite, a relation is thus labeled 'NS' if its first argument is the nucleus of the relation and its second argument, the satellite.

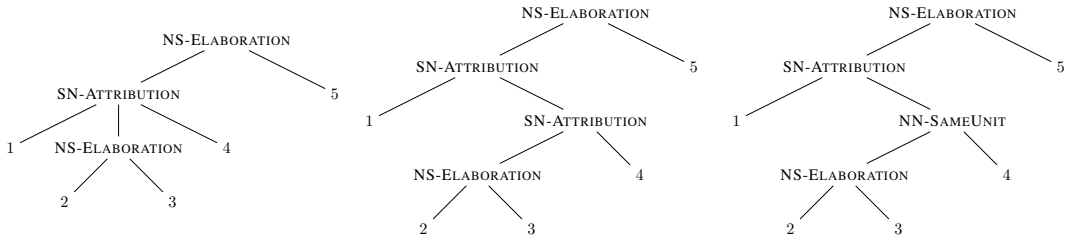

Figure 2: RST predicted, corrected and gold trees for document 1129 from the RST-DT.

Our output will be labeled trees, but below, in our multi-task learning models, we will consider unlabeled parsing and labeled parsing with only nuclearity or relations as auxiliary tasks. Note, however, that in a sequence prediction model we have no guarantee that our output structures form well-formed discourse trees. Therefore we use the following heuristics to guarantee well-formed output structures:

Heuristics The first three heuristics are enough to guarantee well-formed trees in practice: 1) If the first predicted label only contains closing parenthesis, we replace them by opening ones. 2) We remove any right hand side bracket that ends the tree too early, i.e., leads to a well-formed tree only covering a left subsequence of the sequence of EDUs. 3) We add right hand side brackets at the end if there are unclosed brackets after processing the sequence of labels.

However, we not only need to produce well-formed trees, we need to produce well-formed binary trees. Hence, we add the following two heuristics: 4) We first transform them to Chomsky Normal Form.[4] 5) We then remove unary nodes as follows:

- If the unary node is the root, an internal node whose child is a relation node, or a pre-terminal node (its child is a leaf and an EDU node), we replace it by its child.
- If the unary node is an internal node and its child an EDU node (but not a leaf node), then the EDU node becomes its left daughter and the daughter of the EDU node becomes its right daughter.

For example, the document 1129 in the RST-DT is predicted by our baseline model as the sequence in (2a), corrected first using the steps from 1) to 3). Here, we only need to remove a closing parenthesis after EDU 4. We obtain the sequence (2b) corresponding to the first tree in Figure 2. This tree only needs to be binarized, we thus end with the second tree in Figure 2 that can then be compared to the gold tree (third tree in Figure 2).

(2) a. (NS-ELABORATION (SN-ATTRIBUTION (1) (NS-ELABORATION (2)(3)) (4))) (5))

b. (NS-ELABORATION (SN-ATTRIBUTION (1) (NS-ELABORATION (2)(3)) (4)) (5))

3 Auxiliary tasks

We consider two types of auxiliary tasks: first, tasks derived from the RST-DT (multi-view), that is dependency encoding of the trees and additional auxiliary tasks derived from the main one; second, we consider tasks derived from additional data, namely, the Penn Discourse Treebank (Prasad et al., 2008), Timebank (Pustejovsky et al., 2003; Pustejovsky et al., 2005), Factbank (Saurí and Pustejovsky, 2009), Ontonotes (Hovy et al., 2006) and the Santa Barbara corpus of spoken American English (Du Bois, 2000).

All the auxiliary tasks are, as the main one, document-level sequence prediction tasks. In Table 1 we report the number of documents and single labels for each task. We hypothesize that such auxiliary information is useful to address data sparsity for RST discourse parsing.

[4] Using the implementation available in NLTK.

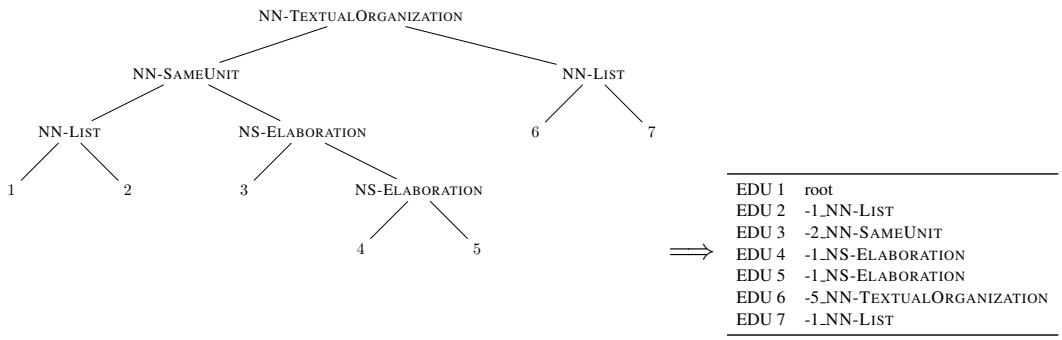

Figure 3: From RST-DT discourse trees to dependency sequence labels. The numbers indicate the position of the head of the EDU, e.g. EDU 2 and EDU 3 have the root EDU 1 as head.

3.1 Building other views of the RST-DT trees

Binary dependencies We first use a representation of the RST-DT trees as binary dependencies (RST-Dep). We roughly do the same transformation as (Muller et al., 2012; Li et al., 2014) but contrary to the latter, we choose as root the nucleus of the root node of the tree rather than the first EDU of the document. More precisely, we associate each node with its saliency set as defined in (Marcu, 1997): The nucleus is the salient EDU of a relation, and the nuclei can go up in the tree with possibly several nuclei in the saliency set of a node. Like Li et al. (2014), we replace all multi-nuclear relations (NN) by mono-nuclear ones choosing the left DU as the nucleus (NS). We thus have only one nucleus in each saliency set. Figure 3 illustrates the conversion of an RST tree into dependency sequence labels.

Nuclearity and relations We further add two alternative views that simply correspond to the main task with one label information removed, keeping either only nuclearity labels (Nuc) or discourse relations (Lab). The idea here is to break up the labeling task, since with the set of 18 discourse relations traditionally used, adding the nuclearity information leads to a large number of 41 labels.

Fine-grained labels Finally, we also use the main task with the original 78 fine-grained relations as an auxiliary task, the idea being of helping the model to learn finer distinctions between the relations.

3.2 Using additional annotations

As we already discussed, discourse relation identification is a hard task that requires access to high-level information. Previous work has shown that and indication about the *events* involved in the discourse units aids identification or constrains the set of inferable relations (Asher and Lascarides, 2003; Danlos and Rambow, 2011; Taboada and Das, 2013). Consider our examples given in the introduction. For instance, modals can indicate conditional relations as in example (1a). Similarly, in example (1b) two asynchronous successive events can be an indication for a causal relation, and besides marking temporal relations, the presence of a present participle may trigger a causal or a manner relation.

In this work we consider time and factuality auxiliary tasks in a multi-task setup. We use two resources for this, Factbank and Timebank, described next. We also include information concerning co-reference using Ontonotes annotations, and use the Santa Barbara corpus that contains conversations split into speaking turns. Finally, we incorporate some annotations from the PDTB, another corpus for discourse that however follows a different annotation scheme than the RST-DT. We describe below the different resources used, as well as how we convert the annotations into sequence labeling tasks in order to use them into the multi-task framework. See Table 1 for dataset characteristics.

Factbank and Timebank FactBank (Saurí and Pustejovsky, 2009) is a corpus of news reports that links events to their degree of factuality. The factuality corresponds to four modality values ('certain', 'probable', 'possible', 'unknown') combined to a polarity value ('positive', 'negative', 'unknown'). Factbank has been annotated on top of TimeBank and a part of AQUAINT TimeML, corpora that provide an annotation of the events according to the TimeML specifications (Pustejovsky et al., 2005). Each event is

annotated with several types of information, among which, of particular interest for discourse, are tense ('infinitive', 'pastpart', 'past', 'future', 'prespart', 'present', 'none'), aspect ('perfective', 'progressive', 'perfective_prog', 'none'), polarity ('positive', 'negative', 'none') and modality (e.g. 'have_to', 'would have to', 'should have to', 'possible', 'must', 'could', ...).

In order to build a sequence prediction task upon FactBank and TimeBank annotations, we choose to use sentences as minimal units. We then simply label each sentence in a document with its most frequent tag for each dimension (tense, aspect, modality and factuality). A more fine grained approach would be to retrieve the clause for each event.

Ontonotes OntoNotes (Hovy et al., 2006) contains, among other layers, the annotation of coreference links between entities in documents. Coreference and rhetorical relations are linked, as shown in (Ji and Eisenstein, 2014a). We only keep the English texts. We use sentences as minimal units. The first sentence of the document is annotated as root. We then label each sentence as coreferent to the immediately previous one, to one preceding sentence or as no coreferent.

Santa Barbara corpus We use the Santa Barbara corpus of spoken American English (Du Bois, 2000) to get a sequence labeling task corresponding to turns in a conversation, the idea being that rhetorical structure could share similarities with the structure of conversations. Specifically, we segment the dialogues by pauses and label the first turn-taking utterance as beginning a new turn. All other utterances are labeled as inside the turn of the current speaker. We randomly split the data into documents containing 100 turns.

Penn Discourse Treebank The PDTB (Prasad et al., 2008) is another corpus annotated at the discourse level for English. Contrary to the RST-DT, the annotation is theory neutral: the spans of text are not necessarily all connected, there is no specific structure representing a document. However, the PDTB contains much more data than the RST-DT, with more than two thousands documents annotated against around four hundreds in the RST-DT, making it interesting to try to take advantage of this relatively large amount of discourse annotated data. Since the PDTB and the RST-DT follow different annotation guidelines (i.e. different definitions of the minimal discourse units, of the relations, of the structures involved), multi-task learning is a relevant framework to try to combine them.

In PDTB, EDUs are the arguments of connectives and adjacent sentences inside paragraphs. The EDUs are mainly clauses, but the annotators are free to choose a span not covering an entire clause, or covering more than one sentence. In this paper, we use sentences as EDUs rather than the manually identified segments: if a relation links more than two sentences, we keep the relation between the last sentence of the first argument and the first sentence of the second argument; if a relation links two fragments belonging to two different sentences, we expand the text of each argument to cover the entire sentences. We ignore intra-sentential explicit relations.[5]

We use a BIO annotation scheme for relations between adjacent sentences. More precisely, a sentence is labeled with a BIO label and a discourse relation R_i among the 16 corresponding to the second level in the PDTB hierarchy of sense[6] and the pseudo relation EntRel corresponding to a link between entities. A sentence labeled with "B-R_a" is the first argument of a relation R_a whose second argument is the following sentence. If this following sentence is also the first argument of a relation R_b, it is labeled as "B-R_b", else, it is labeled as ending the current relation, thus "I-R_a". A sentence that is not linked to the previous or following sentence is labeled with "O".

4 Hierarchical bi-LSTMs and baselines

Our main technical contribution is a hierarchical bi-LSTM that composes embeddings for a sequence of words from lower-level word bi-LSTMs, and uses these to predict sequences of labels, encoding especially discourse tree structures.

[5]Preliminary experiments including non overlapping intra-sentential relations did not show improvements against only keeping inter-sentential ones. However, including intra-sentential instances requires more pre-processing and it makes necessary to decide which intra-sentential relations to keep to avoid overlaps.

[6]We only keep the first relation annotated for a pair of arguments.

Task	# Doc	# Labels
Constituent	322	1955
Nuclearity	322	284
Relation	322	1159
Dependency	322	708
Fine grained	322	2,700
Aspect	208	4
Factuality	208	7
Modality	208	10
Polarity	208	3
Tense	208	7
Coreference	2,361	4
PDTB	2,065	35
Speech	446	2

Table 1: Number of documents (# Doc) and labels (# Labels) per task (training data). The main task corresponds to the first line (Constituent).

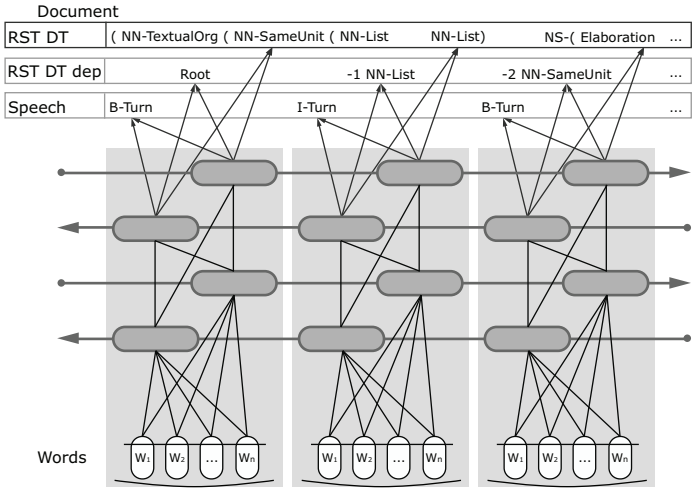

Figure 4: Multi-task learning, hierarchical bi-LSTM network architecture (with 2 layers).

In regular bi-directional recurrent neural networks (bi-RNNs), sequences are read in both regular and reversed order, enabling conditioning predictions on both left and right context. Below, in the forward pass, we run the input data through an embedding layer and compute the predictions of the forward and backward states, which are connected in one or more feed-forward layers, from which we compute the softmax predictions for the sequence based on a linear transformation. We then calculate the objective function derivative for the sequence using cross-entropy (logistic loss) and use backpropagation to calculate gradients and update the weights accordingly. LSTMs (Hochreiter and Schmidhuber, 1997) replace the cells of RNNs with LSTM cells, in which multiplicative gate units learn to open and close access to the error signal.

The overall architecture is shown in Figure 4: each input sequence in the document (i.e. a discourse unit, a speaking turn, a sentence, depending on the task) goes through the hierarchical bi-LSTM that outputs a sequence of labels for the entire document. In particular, an input sequence is represented as a sequence of word embeddings. This sequence goes first through the bi-directional LSTM at the lower level, and the final states (forward, backward) of the bi-LSTMs is taken as input representation for the document-level bi-LSTM at the upper level, which consists of two stacked layers.

For multi-task learning, each task is associated with a specific output layer, whereas the inner layers – the stacked LSTMs – are shared across the tasks. At training time, we randomly sample data points from target or auxiliary tasks and do forward predictions. In the backward pass, we modify the weights of the shared layers and the task-specific outer layer. Except for the outer layer, the target task model is thus regularized by the induction of auxiliary models.

Bi-LSTMs have already been used for syntactic chunking (Huang et al., 2015) and semantic role labeling (Zhou and Xu, 2015), as well as other tasks. Our model differs from most of these models in being a hierarchical model, composing word embeddings into sentence embeddings that are the inputs of a bigger bi-LSTM model. This means our model can also be initialized by pre-trained word embeddings. We implemented our recurrent network in CNN/pycnn,[7] fixing the random seed. We use standard SGD for learning our model parameters.

5 Experiments

Data The RST-DT contains 385 Wall Street Journal articles from the Penn Treebank (Marcus et al., 1993), with 347 documents for training and 38 for testing in the split used in previous studies. We

[7] https://github.com/yoavg/cnn/

follow previous works in using gold standard segmentation (Joty et al., 2012; Ji and Eisenstein, 2014b). Discourse segmentation on the RST-DT can be performed with performance above 95% in accuracy (Xuan Bach et al., 2012). The RST-DT contains newswire articles from the Wall Street Journal.

Baseline As baseline, we train a standard bi-LSTMs on the RST-DT corpus without any auxiliary task information.

Systems As our system, we use hierarchical bi-LSTMs with task supervision from other related tasks. We experiment with using pre-trained embeddings in both baselines and systems.

Competitive systems We compare our approach with the state-of-the-art text-level discourse parser DPLP (Ji and Eisenstein, 2014b). In our comparison, we reproduced the best results reported, including both proposed approaches for DPLP – DPLP concat (*concatenation form* for the projection matrix) and DPLP general (*general form*).

Parameter tuning We used a development set of 25 documents randomly chosen among the training set. We optimized the number of passes p over the data ($p \in [10, 60]$), the value of the Gaussian noise ($\sigma \in \{0.0, 0.2\}$), the number of hidden dimensions ($d \in \{200, 400\}$), the number of stacked layers ($h \in \{1, 2, 3, 4, 5\}$), and the auxiliary tasks to be included and combined. In the end, we report results using 2 feed-forward layers with 128 dimensions, a Gaussian noise with sigma of 0.2, 200 hidden dimensions, 20 passes over the data, 2 layers and Polyglot embeddings (Al-Rfou et al., 2013)[8].

Metrics Following (Marcu, 2000b) and most subsequent work, output trees are evaluated against gold trees in terms of how similar they bracket the EDUs (Span), how often they agree about nuclei when predicting a true bracket (Nuclearity), and in terms of the relation label, i.e., the overlap between the shared brackets between predicted and gold trees (Relation).[9] These scores are analogous to labeled and unlabeled syntactic parser evaluation metrics. The exact definitions of the three metrics are:

- Span: This metric is the unlabeled F_1 over gold and predicted trees, and identical to the PARSEVAL metric in syntactic parsing. This metric reflects a correct bracketing and ignores nuclearity and relation labels.
- Nuclearity: This metric is the labeled F_1 over gold and predicted discourse trees, disregarding the discourse relations.
- Relation: This metric is the labeled F_1 over gold and predicted discourse trees, disregarding the nuclearity information.

6 Results

Our results are summarized in Table 2. We note that the bi-LSTM baseline that only receives task supervision from RST-DT discourse trees achieves scores comparable to the state-of-the-art for the unlabeled structure (Span), but lower scores for the other metrics.

More importantly, multi-task learning, i.e., combining different representations of the data, leads to substantial improvements over our baseline for 8 out of the 11 tasks tested. We found that it is much more beneficial to have multiple views, thus, interestingly using different views on the data, with all the tasks derived from the main one leading to improvements (RSTFin, RSTDep, Nuc+Lab). Especially, the model takes advantage of using the data from the main task but with fine grained relations, with 82.88% in unlabelled F_1 (Span), 67.46% in labelled F_1 considering nuclearity (Nuclearity), and 53.25% in labelled F_1 considering relations (Relation). This auxiliary view helps the model to discriminate between the relations.

Most of the tasks derived from additional annotations also lead to improvements. Especially, we found that the speech data (Speech) leads to good results: this confirms our assumption that the turns

[8] `https://sites.google.com/site/rmyeid/projects/polyglot`

[9] We use the evaluation script provided at `https://github.com/jiyfeng/DPLP`.

System	RSTFin	Fact	Speech	Asp	RSTDep	Nuc+lab	Mod	Pol	PDTB	Coref	Ten	Span	Nuclearity	Relation
Prior work														
DPLP concat	-	-	-	-	-	-	-	-	-	-	-	82.08	**71.13**	61.63
DPLP general	-	-	-	-	-	-	-	-	-	-	-	81.60	70.95	**61.75**
Our work														
Hier-LSTM	-	-	-	-	-	-	-	-	-	-	-	81.39	64.54	49.15
MTL-Hier-LSTM	✓	-	-	-	-	-	-	-	-	-	-	82.88	67.46	53.25
MTL-Hier-LSTM	-	✓	-	-	-	-	-	-	-	-	-	83.40	67.16	52.10
MTL-Hier-LSTM	-	-	✓	-	-	-	-	-	-	-	-	83.26	67.51	51.75
MTL-Hier-LSTM	-	-	-	✓	-	-	-	-	-	-	-	83.69	66.25	51.25
MTL-Hier-LSTM	-	-	-	-	✓	-	-	-	-	-	-	81.25	65.34	51.24
MTL-Hier-LSTM	-	-	-	-	-	✓	-	-	-	-	-	82.09	65.68	51.12
MTL-Hier-LSTM	-	-	-	-	-	-	✓	-	-	-	-	81.66	65.31	50.58
MTL-Hier-LSTM	-	-	-	-	-	-	-	✓	-	-	-	82.01	65.29	50.11
MTL-Hier-LSTM	-	-	-	-	-	-	-	-	✓	-	-	81.61	63.10	48.89
MTL-Hier-LSTM	-	-	-	-	-	-	-	-	-	✓	-	80.26	63.35	47.70
MTL-Hier-LSTM	-	-	-	-	-	-	-	-	-	-	✓	81.33	62.34	47.57
Best combination	-	-	-	-	✓	✓	✓	-	✓	-	-	**83.62**	69.77	55.11
Human annotation	-	-	-	-	-	-	-	-	-	-	-	88.70	77.72	65.75

Table 2: Parsing results of different models on the RST-DT test data. Prior work results are reprinted (DPLP) (Ji and Eisenstein, 2014b). The auxiliary tasks are: RST-DT sequences from trees but keeping only the relations (Lab) or the nuclearity information (Nuc), RST-DT dependency parsing (RSTDep), sequence labels from Factbank using modality information (Mod), and inter-sentential relation from the PDTB (PDTB).

of speech and the structures involved share some similarities with the rhetorical units and structures. Moreover, factuality (Fact), aspect (Asp), modality (Mod) and polarity (Pol) information prove to be useful for discourse parsing. On the other hand, the tasks derived from tense (Ten) and coreference (Coref) annotations do not lead to improvements. These information, crucial for the task, would probably benefit from a finer grained encoding at the sentence level. The task derived from the PDTB, taken alone, lowers slightly the results.

Finally, we experiment with task combinations. Our best system only uses the views based on nuclearity and label (Nuc+lab), the encoding of the tree as dependency (RSTDep), the modality information (Mod) and the task derived from the PDTB data. This combination leads to substantial improvements, with 83.62% in unlabelled F_1 (Span), 69.77% in labelled F_1 considering nuclearity (Nuclearity), and 55.11% in labelled F_1 considering relations (Relation). This closes 60,7% of the gap to human performance on unlabelled discourse parsing. It is slightly better than state-of-the-art in discourse parsing for Span. Feng and Hirst (2014) proposed a system with better scores for these metrics, but the comparison to their system is not entirely fair, since they add common-sense constraints that are not clearly explained and post-editing. Besides, there is no single approach that does best for all metrics.

Our results indicate that our architecture learns useful representations capturing some of the syntactic and contextual information needed for the task.

7 Related work

Some of the first text-level discourse parsers were based on hand-crafted rules and heuristics, making mainly use of the connectives as indication of the relations and using constraints to build the entire RST trees (Marcu, 2000a; Le Thanh et al., 2004).

More recent works proposed learning based approaches inspired by syntactic parsing. Hernault et al. (2010) (HILDA) proposed a greedy approach with SVM classifiers performing attachment and relation classification at each step of the tree building. Joty et al. (2012) (TSP) built a two-stage parsing system, training separate sequential models (CRF) for the intra and the inter-sentential levels. These models jointly learn the relation and the structure, and a CKY-like algorithm is used to find the optimal tree. Feng and Hirst (2014) noticed the inefficiency of TSP and proposed a greedy approach inspired by

HILDA but using CRF as local models for the inter- and intra-sententials levels, allowing to take into account sequential dependencies.

Last studies also focused on the issue of building a good representation of the data. Feng and Hirst (2012) introduced linguistic features, mostly syntactic and contextual ones. Ji and Eisenstein (2014b) (DPLP) proposed to learn jointly the representation and the task, more precisely a projection matrix that maps the bag-of-words representation of the discourse units into a new vector space. This idea is promising, but a drawback could be the limited amount of data available in the RST-DT, an issue even more crucial for other languages.

Discourse parsing has proven useful for many applications (Taboada and Mann, 2006), ranging from summarization (Daumé III and Marcu, 2009; Thione et al., 2004; Sporleder and Lapata, 2005; Louis et al., 2010), sentiment analysis (Bhatia et al., 2015) or essay scoring (Burstein et al., 2003; Higgins et al., 2004). However, the range of applications and the improvement allowed are for now limited by the low performance of the existing discourse parsers.

We are not aware of other studies trying to combine various encodings of the RST-DT trees or to leverage relevant information through multi-task learning to improve discourse parsing. To the best of our knowledge, multi-task learning has only been used for discourse relation classification (Lan et al., 2013) on the Penn Discourse Treebank to combine implicit and explicit data.

We are not the first to propose using bi-LSTMs for tree structure prediction problems. Zhou and Xu (2015), for example, use bi-LSTMs to produce semantic role labelling structures. Zhang et al. (2015) did the same for relation extraction. None of them considered multi-task learning architectures, however. Multi-task learning in deep networks was first introduced by Caruana (1993), who did multi-task learning by doing parameter sharing across several deep networks, letting them share hidden layers. The same technique was used by Collobert et al. (2011) for various NLP tasks, and for sentence compression in (Klerke et al., 2016). Hierarchical multi-task bi-LSTMs have been previously used for part-of-speech tagging (Plank et al., 2016).

8 Conclusion and future work

We presented the first experiments exploiting different views of the data and related tasks to improve text-level discourse parsing. We presented a hierarchical bi-LSTM model allowing to leverage information from various sequence prediction tasks (multi-task learning) that achieves a new state-of-the-art performance on unlabeled text-level discourse parsing, and competitive performance in predicting nuclearity and discourse relations.

For relation prediction, future work includes adding additional information at the sentence level, such as syntactic information used in most of the studies identifying discourse relation on the PDTB (Pitler et al., 2009; Lin et al., 2009; Rutherford and Xue, 2014), or better representation of the combination between the arguments (Ji and Eisenstein, 2014b).

Acknowledgements

We thank the three anonymous reviewers for their comments. Chloé Braud and Anders Søgaard were funded by the ERC Starting Grant LOWLANDS No. 313695.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of Conll*.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from rst discourse parsing. In *Proceedings of EMNLP*.

Jill Burstein, Daniel Marcu, and Kevin Knight. 2003. Finding the write stuff: automatic identification of discourse structure in student essays. *IEEE Intelligent Systems: Special Issue on Advances in Natural Language Processing*, 18.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Rich Caruana. 1993. Multitask learning: a knowledge-based source of inductive bias. In *ICML*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Laurence Danlos and Owen Rambow. 2011. Discourse Relations and Propositional Attitudes. In *CID 2011 - Fourth International workshop on Constraints in Discourse*.

Hal Daumé III and Daniel Marcu. 2009. A noisy-channel model for document compression. In *Proceedings of ACL*.

John W Du Bois. 2000. *Santa Barbara Corpus of Spoken American English*. University of California, Santa Barbara Center for the Study of Discourse.

Vanessa Wei Feng and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of ACL*.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of ACL*.

Hugo Hernault, Helmut Prendinger, David A. duVerle, and Mitsuru Ishizuka. 2010. Hilda: A discourse parser using support vector machine classification. *Dialogue and Discourse*, 1:1–33.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *Proceedings of HLT-NAACL*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of HLT-NAACL*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv:1508.01991*.

Yangfeng Ji and Jacob Eisenstein. 2014a. One vector is not enough: Entity-augmented distributional semantics for discourse relations. *TACL*.

Yangfeng Ji and Jacob Eisenstein. 2014b. Representation learning for text-level discourse parsing. In *Proceedings of ACL*.

Xin Jin, Fuzhen Zhuang, Shuhui Wang, Qing He, and Zhongzhi Shi. 2013. Shared structure learning for multiple tasks with multiple views. In *Machine Learning and Knowledge Discovery in Databases*, pages 353–368. Springer.

Shafiq R. Joty, Giuseppe Carenini, and Raymond T. Ng. 2012. A novel discriminative framework for sentence-level discourse analysis. In *Proceedings of EMNLP*.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proceedings of NAACL*.

Man Lan, Yu Xu, and Zhengyu Niu. 2013. Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In *Proceedings of ACL*.

Huong Le Thanh, Geetha Abeysinghe, and Christian Huyck. 2004. Generating discourse structures for written text. In *Proceedings of COLING*.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of ACL*.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of EMNLP*.

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of SIGDIAL*.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory : Toward a functional theory of text organization. *Text*, 8:243–281.

Daniel Marcu. 1997. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. Ph.D. thesis, University of Toronto.

Daniel Marcu. 2000a. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*.

Daniel Marcu. 2000b. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Philippe Muller, Stergos Afantenos, Pascal Denis, and Nicholas Asher. 2012. Constrained decoding for text-level discourse parsing. In *Proceedings of COLING*.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of ACL-IJCNLP*.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of ACL*.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of LREC*.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Rob Gaizauskas, Andrea Setzer, Dragomir R. Radev, Beth Sundheim, David S. Day, Lisa Ferro, and Marcia Lazo. 2003. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics*.

James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Saurí. 2005. Temporal and event information in natural language text. *Language Resources and Evaluation*, 39(2):123–164.

Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of EACL*.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. In *Proceedings of LREC*, volume 43, pages 227–268.

Caroline Sporleder and Mirella Lapata. 2005. Discourse chunking and its application to sentence compression. In *Proceedings of HLT/EMNLP*.

Manfred Stede. 2011. *Discourse Processing*. Morgan & Claypool.

Maite Taboada and Debopam Das. 2013. Annotation upon annotation: Adding signalling information to a corpus of discourse relations. *Dialogue and Discourse*, 4(2):249–281.

Maite Taboada and William C. Mann. 2006. Applications of rhetorical structure theory. *Discourse Studies*, 8:567–588.

Gian Lorenzo Thione, Martin Van den Berg, Livia Polanyi, and Chris Culy. 2004. Hybrid text summarization: Combining external relevance measures with structural analysis. In *Proceedings of the ACL Workshop Text Summarization Branches Out*.

Ngo Xuan Bach, Nguyen Le Minh, and Akira Shimazu. 2012. A reranking model for discourse segmentation using subtree features. In *Proceedings of Sigdial*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of PACLIC*.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of ACL*.