resulting system strongly outperforms the prior state-of-the-art at labeled F-measure, obtaining raw improvements of roughly 6% on relation labels and 2.5% on nuclearity. In addition, we show that the latent representation coheres well with the characterization of discourse connectives in the Penn Discourse Treebank (Prasad et al., 2008).

2 Model

The core idea of this paper is to project lexical features into a latent space that facilitates discourse parsing. In this way, we can capture the meaning of each discourse unit, without suffering from the very high dimensionality of a lexical representation. While such feature learning approaches have proven to increase robustness for parsing, POS tagging, and NER (Miller et al., 2004; Koo et al., 2008; Turian et al., 2010), they would seem to have an especially promising role for discourse, where training data is relatively sparse and ambiguity is considerable. Prasad et al. (2010) show that there is a long tail of alternative lexicalizations for discourse relations in the Penn Discourse Treebank, posing obvious challenges for approaches based on directly matching lexical features observed in the training data.

Based on this observation, our goal is to learn a function that transforms lexical features into a much lower-dimensional latent representation, while simultaneously learning to predict discourse structure based on this latent representation. In this paper, we consider a simple transformation function, linear projection. Thus, we name the approach DPLP: Discourse Parsing from Linear Projection. We apply transition-based (incremental) structured prediction to obtain a discourse parse, training a predictor to make the correct incremental moves to match the annotations of training data in the RST Treebank. This supervision signal is then used to learn both the weights and the projection matrix in a large-margin framework.

2.1 Shift-reduce discourse parsing

We construct RST Trees using shift-reduce parsing, as first proposed by Marcu (1999). At each point in the parsing process, we maintain a stack and a queue; initially the stack is empty and the first elementary discourse unit (EDU) in the document is at the front of the queue.[1] The parser can then choose either to *shift* the front of the queue onto the top of the stack, or to *reduce* the top two elements on the stack in a discourse relation. The reduction operation must choose both the type of relation and which element will be the nucleus. So, overall there are multiple reduce operations with specific relation types and nucleus positions. Shift-reduce parsing can be learned as a classification task, where the classifier uses features of the elements in the stack and queue to decide what move to take. Previous work has employed decision trees (Marcu, 1999) and the averaged perceptron (Collins and Roark, 2004; Sagae, 2009) for this purpose. Instead, we employ a large-margin classifier, because we can compute derivatives of the margin-based objective function with respect to both the classifier weights as well as the projection matrix.

Notation	Explanation
$\mathcal{V}$	Vocabulary for surface features
V	Size of $\mathcal{V}$
K	Dimension of latent space
$\mathbf{w}_m$	Classification weights for class m
C	Total number of classes, which correspond to possible shift-reduce operations
$\mathbf{A}$	Parameter of the representation function (also the projection matrix in the linear representation function)
$\mathbf{v}_i$	Word count vector of discourse unit i
$\mathbf{v}$	Vertical concatenation of word count vectors for the three discourse units currently being considered by the parser
λ	Regularization for classification weights
τ	Regularization for projection matrix
ξ_i	Slack variable for sample i
$\eta_{i,m}$	Dual variable for sample i and class m
α_t	Learning rate at iteration t

Table 1: Summary of mathematical notation

2.2 Discourse parsing with projected features

More formally, we denote the surface feature vocabulary $\mathcal{V}$, and represent each EDU as the numeric vector $\mathbf{v} \in \mathbb{N}^V$, where $V = \#|\mathcal{V}|$ and the n-th element of $\mathbf{v}$ is the count of the n-th surface feature in this EDU (see Table 1 for a summary of notation). During shift-reduce parsing, we consider features of three EDUs:[2] the top two elements on

[1] We do not address segmentation of text into elementary discourse units in this paper. Standard classification-based approaches can achieve a segmentation F-measure of 94% (Hernault et al., 2010); a more complex reranking model does slightly better, at 95% F-Measure with automatically-generated parse trees, and 96.6% with gold annotated trees (Xuan Bach et al., 2012). Human agreement reaches 98% F-Measure.

[2] After applying a reduce operation, the stack will include a span that contains multiple EDUs. We follow the *strong*