One Vector is Not Enough: Entity-Augmented Distributed Semantics for Discourse Relations

Yangfeng Ji and **Jacob Eisenstein**
School of Interactive Computing
Georgia Institute of Technology
{jiyfeng, jacobe}@gatech.edu

Abstract

Discourse relations bind smaller linguistic units into coherent texts. Automatically identifying discourse relations is difficult, because it requires understanding the semantics of the linked arguments. A more subtle challenge is that it is not enough to represent the meaning of each argument of a discourse relation, because the relation may depend on links between lower-level components, such as entity mentions. Our solution computes distributed meaning representations for each discourse argument by composition up the syntactic parse tree. We also perform a downward compositional pass to capture the meaning of coreferent entity mentions. Implicit discourse relations are then predicted from these two representations, obtaining substantial improvements on the Penn Discourse Treebank.

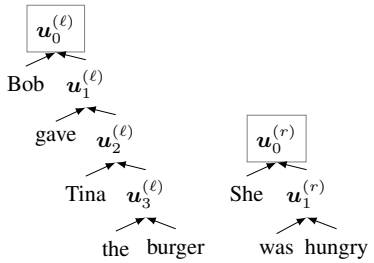
1 Introduction

The high-level organization of text can be characterized in terms of **discourse relations** between adjacent spans of text (Knott, 1996; Mann, 1984; Webber et al., 1999). Identifying these relations has been shown to be relevant to tasks such as summarization (Louis et al., 2010a; Yoshida et al., 2014), sentiment analysis (Somasundaran et al., 2009), coherence evaluation (Lin et al., 2011), and question answering (Jansen et al., 2014). While the Penn Discourse Treebank (PDTB) now provides a large dataset annotated for discourse relations (Prasad et al., 2008), the automatic identification of implicit relations is a difficult task, with state-of-the-art performance at roughly 40% (Lin et al., 2009).

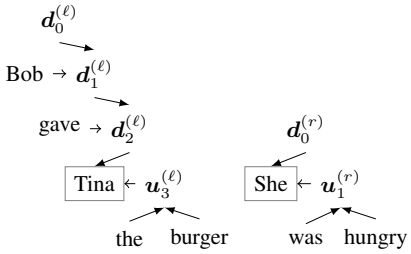
One reason for this poor performance is that discourse relations are rooted in semantics (Forbes-Riley et al., 2006), which can be difficult to recover from surface level features. Consider the implicit discourse relation between the following two sentences (also shown in Figure 1a):

(1) *Bob gave Tina the burger.*
She was hungry.

While a connector like *because* seems appropriate here, there is little surface information to signal this relationship, unless the model has managed to learn a bilexical relationship between *burger* and *hungry*. Learning all such relationships from annotated data — including the relationship of *hungry* to *knish*, *pierogie*, *pupusa* etc — would require far more data than can possibly be annotated.



(a) The distributed representations of *burger* and *hungry* are propagated up the parse tree, clarifying the implicit discourse relation between $\boldsymbol{u}_0^{(\ell)}$ and $\boldsymbol{u}_0^{(r)}$.



(b) Distributed representations for the coreferent mentions *Tina* and *she* are computed from the parent and sibling nodes.

Figure 1: Distributed representations are computed through composition over the parse.

Transactions of the Association for Computational Linguistics, vol. 3, pp. 329–344, 2015. Action Editor: Alexander Koller.
Submission batch: 12/2014; Revision batch 3/2015; Revision batch 4/2015; Published 6/2015.

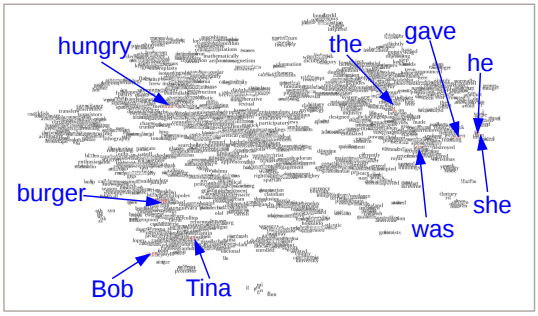


Figure 2: t-SNE visualization (van der Maaten and Hinton, 2008) of word representations in the PDTB corpus.

We address this issue by applying a discriminatively-trained model of compositional distributed semantics to discourse relation classification (Socher et al., 2013; Baroni et al., 2014a). The meaning of each discourse argument is represented as a vector (Turney and Pantel, 2010), which is computed through a series of bottom-up compositional operations over the syntactic parse tree. The discourse relation can then be predicted as a bilinear combination of these vector representations. Both the prediction matrix and the compositional operator are trained in a supervised large-margin framework (Socher et al., 2011), ensuring that the learned compositional operation produces semantic representations that are useful for discourse. We show that when combined with a small number of surface features, this approach outperforms prior work on the classification of implicit discourse relations in the PDTB.

Despite these positive results, we argue that bottom-up vector-based representations of discourse arguments are insufficient to capture their relations. To see why, consider what happens if we make a tiny change to example (1):

(2) *Bob gave Tina the burger.*
***He** was hungry.*

After changing the subject of the second sentence to Bob, the connective "*because*" no longer seems appropriate; a contrastive connector like *although* is preferred. But despite the radical difference in meaning, the bottom-up distributed representation of the second sentence will be almost unchanged: the syntactic structure remains identical, and the words *he* and *she* have very similar word representations (see Figure 2). If we reduce each discourse argument span to a single vector, built from the elements in the argument itself, we cannot possibly capture the ways that discourse relations are signaled by entities and their roles (Cristea et al., 1998; Louis et al., 2010b). As Mooney (2014) puts it, "you can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!"

We address this issue by computing vector representations not only for each discourse argument, but also for each coreferent entity mention. These representations are meant to capture the **role** played by the entity in the text, and so they must take the entire span of text into account. We compute entity-role representations using a feed-forward compositional model, which combines "upward" and "downward" passes through the syntactic structure, shown in Figure 1b. In the example, the downward representations for *Tina* and *she* are computed from a combination of the parent and sibling nodes in the binarized parse tree. Representations for these coreferent mentions are then combined in a bilinear product, and help to predict the implicit discourse relation. In example (2), we resolve *he* to *Bob*, and combine their vector representations instead, yielding a different prediction about the discourse relation.

Our overall approach combines surface features, distributed representations of discourse arguments, and distributed representations of entity mentions. It achieves a 4% improvement in accuracy over the best previous work (Lin et al., 2009) on multiclass discourse relation classification, and also outperforms more recent work on binary classification. The novel entity-augmented distributed representation improves accuracy over the "upward" compositional model, showing the importance of representing the meaning of coreferent entity mentions.

2 Entity augmented distributed semantics

We now formally define our approach to entity-augmented distributed semantics, using the notation shown in Table 1. For clarity of exposition, we focus on discourse relations between pairs of sentences. The extension to non-sentence arguments is discussed in Section 5.

Notation	Explanation
$\ell(i), r(i)$	left and right children of i
$\rho(i), s(i)$	parent and sibling of i
$\mathcal{A}(m, n)$	set of aligned entities between arguments m and n
$\mathcal{Y}$	set of discourse relations
y^*	gold discourse relation
$\psi(y)$	decision function
$\boldsymbol{u}$	upward vector
$\boldsymbol{d}$	downward vector
$\mathbf{A}_y$	classification parameter associated with upward vectors
$\mathbf{B}_y$	classification parameter associated with downward vectors
$\mathbf{U}$	composition operator in upward composition procedure
$\mathbf{V}$	composition operator in downward composition procedure
$\mathcal{L}(\boldsymbol{\theta})$	objective function

Table 1: Table of notation

2.1 Upward pass: argument semantics

Distributed representations for discourse arguments are computed in a feed-forward "upward" pass: each non-terminal in the binarized syntactic parse tree has a K-dimensional vector representation that is computed from the representations of its children, bottoming out in pre-trained representations of individual words.

We follow the Recursive Neural Network (RNN) model of Socher et al. (2011). For a given parent node i, we denote the left child as $\ell(i)$, and the right child as $r(i)$; we compose their representations to obtain,

$$\boldsymbol{u}_i = \tanh\left(\mathbf{U}[\boldsymbol{u}_{\ell(i)}; \boldsymbol{u}_{r(i)}]\right), \quad (1)$$

where $\tanh(\cdot)$ is the element-wise hyperbolic tangent function (Pascanu et al., 2012), and $\mathbf{U} \in \mathbb{R}^{K \times 2K}$ is the upward composition matrix. We apply this compositional procedure from the bottom up, ultimately obtaining the argument-level representation $\boldsymbol{u}_0$. The base case is found at the leaves of the tree, which are set equal to pre-trained word vector representations. For example, in the second sentence of Figure 1, we combine the word representations of *was* and *hungry* to obtain $\boldsymbol{u}_1^{(r)}$, and then combine $\boldsymbol{u}_1^{(r)}$ with the word representation of *she* to obtain $\boldsymbol{u}_0^{(r)}$. Note that the upward pass is feedforward, meaning that there are no cycles and all nodes can be computed in linear time.

2.2 Downward pass: entity semantics

As seen in the contrast between Examples 1 and 2, a model that uses a bottom-up vector representation for each discourse argument would find little to distinguish between *she was hungry* and *he was hungry*. It would therefore almost certainly fail to identify the correct discourse relation for at least one of these cases, which requires tracking the roles played by the entities that are coreferent in each pair of sentences. To address this issue, we augment the representation of each argument with additional vectors, representing the semantics of the role played by each coreferent entity in each argument. For example, in (1a), Tina got the burger, and in (1b), she was hungry. Rather than represent this information in a logical form — which would require robust parsing to a logical representation — we represent it through additional distributed vectors.

The role of a constituent i can be viewed as a combination of information from two neighboring nodes in the parse tree: its parent $\rho(i)$, and its sibling $s(i)$. We can make a downward pass, computing the downward vector $\boldsymbol{d}_i$ from the downward vector of the parent $\boldsymbol{d}_{\rho(i)}$, and the **upward** vector of the sibling $\boldsymbol{u}_{s(i)}$:

$$\boldsymbol{d}_i = \tanh\left(\mathbf{V}[\boldsymbol{d}_{\rho(i)}; \boldsymbol{u}_{s(i)}]\right), \quad (2)$$

where $\mathbf{V} \in \mathbb{R}^{K \times 2K}$ is the downward composition matrix. The base case of this recursive procedure occurs at the root of the parse tree, which is set equal to the upward representation, $\boldsymbol{d}_0 \triangleq \boldsymbol{u}_0$. This procedure is illustrated in Figure 1b: for *Tina*, the parent node is $\boldsymbol{d}_2^{(\ell)}$, and the sibling is $\boldsymbol{u}_3^{(\ell)}$.

This up-down compositional algorithm propagates sentence-level distributed semantics back to entity mentions. The representation of each mention's role in the sentence is based on the corresponding role of the parent node in the parse tree, and on the internal meaning representation of the sibling node, which is computed by upward composition. Note that this algorithm is designed to maintain the *feedforward* nature of the neural network, so that we can efficiently compute all nodes without iterating. Each downward node $\boldsymbol{d}_i$ influences only other downward nodes $\boldsymbol{d}_j$ where $j > i$, meaning that the downward pass is feedforward. The upward node is also feedforward: each upward node $\boldsymbol{u}_i$ influences only other upward nodes $\boldsymbol{u}_j$ where $j < i$. Since the upward and downward passes are each feedforward, and the downward nodes do not influence any upward nodes, the combined up-down network is also feedforward. This ensures that we can efficiently com-

pute all $\boldsymbol{u}_i$ and $\boldsymbol{d}_i$ in time that is linear in the length of the input. In Section 7.2, we compare our approach with recent related work on alternative two-pass distributed compositional models.

Connection to the inside-outside algorithm In the inside-outside algorithm for computing marginal probabilities in a probabilistic context-free grammar (Lari and Young, 1990), the inside scores are constructed in a bottom-up fashion, like our upward nodes; the outside score for node i is constructed from a product of the outside score of the parent $\rho(i)$ and the inside score of the sibling $s(i)$, like our downward nodes. The standard inside-outside algorithm sums over all possible parse trees, but since the parse tree is observed in our case, a closer analogy would be to the constrained version of the inside-outside algorithm for latent variable grammars (Petrov et al., 2006). Cohen et al. (2014) describe a tensor formulation of the constrained inside-outside algorithm; similarly, we could compute the downward vectors by a tensor contraction of the parent and sibling vectors (Smolensky, 1990; Socher et al., 2014). However, this would involve K^3 parameters, rather than the K^2 parameters in our matrix-vector composition.

3 Predicting discourse relations

To predict the discourse relation between an argument pair (m, n), the decision function is a sum of bilinear products,

$$\psi(y) = (\boldsymbol{u}_0^{(m)})^\top \mathbf{A}_y \boldsymbol{u}_0^{(n)} + \sum_{i,j \in \mathcal{A}(m,n)} (\boldsymbol{d}_i^{(m)})^\top \mathbf{B}_y \boldsymbol{d}_j^{(n)} + b_y, \quad (3)$$

where $\mathbf{A}_y \in \mathbb{R}^{K \times K}$ and $\mathbf{B}_y \in \mathbb{R}^{K \times K}$ are the classification parameters for relation y. A scalar b_y is used as the bias term for relation y, and $\mathcal{A}(m, n)$ is the set of coreferent entity mentions shared by the argument pair (m, n). The decision value $\psi(y)$ of relation y is therefore based on the upward vectors at the root, $\boldsymbol{u}_0^{(m)}$ and $\boldsymbol{u}_0^{(n)}$, as well as on the downward vectors for each pair of aligned entity mentions. For the cases where there are no coreferent entity mentions between two sentences, $\mathcal{A}(m, n) = \varnothing$, the classification model considers only the upward vectors at the root.

To avoid overfitting, we apply a low-dimensional approximation to each $\mathbf{A}_y$,

$$\mathbf{A}_y = \boldsymbol{a}_{y,1} \boldsymbol{a}_{y,2}^\top + \text{diag}(\boldsymbol{a}_{y,3}). \quad (4)$$

The same approximation is also applied to each $\mathbf{B}_y$, reducing the number of classification parameters from $2 \times \#|\mathcal{Y}| \times K^2$ to $2 \times \#|\mathcal{Y}| \times 3K$.

Surface features Prior work has identified a number of useful surface-level features (Lin et al., 2009), and the classification model can easily be extended to include them. Defining $\boldsymbol{\phi}_{(m,n)}$ as the vector of surface features extracted from the argument pair (m, n), the corresponding decision function is modified as,

$$\psi(y) = (\boldsymbol{u}_0^{(m)})^\top \mathbf{A}_y \boldsymbol{u}_0^{(n)} + \sum_{i,j \in \mathcal{A}(m,n)} (\boldsymbol{d}_i^{(m)})^\top \mathbf{B}_y \boldsymbol{d}_j^{(n)} + \boldsymbol{\beta}_y^\top \boldsymbol{\phi}_{(m,n)} + b_y, \quad (5)$$

where $\boldsymbol{\beta}_y$ is the classification weight on surface features for relation y. We describe these features in Section 5.

4 Large-margin learning framework

There are two sets of parameters to be learned: the classification parameters $\boldsymbol{\theta}_{class} = \{\mathbf{A}_y, \mathbf{B}_y, \boldsymbol{\beta}_y, b_y\}_{y \in \mathcal{Y}}$, and the composition parameters $\boldsymbol{\theta}_{comp} = \{\mathbf{U}, \mathbf{V}\}$. We use pre-trained word representations, and do not update them. While prior work shows that it can be advantageous to retrain word representations for discourse analysis (Ji and Eisenstein, 2014), our preliminary experiments found that updating the word representations led to serious overfitting in this model.

Following Socher et al. (2011), we define a large margin objective, and use backpropagation to learn all parameters of the network jointly (Goller and Kuchler, 1996). Learning is performed using stochastic gradient descent (Bottou, 1998), so we present the learning problem for a single argument pair (m, n) with the gold discourse relation y^*. The objective function for this training example is a regularized hinge loss,

$$\ell(\boldsymbol{\theta}) = \sum_{y' : y' \neq y^*} \max\Big(0, 1 - \psi(y^*) + \psi(y')\Big) + \lambda ||\boldsymbol{\theta}||_2^2 \quad (6)$$

where $\boldsymbol{\theta} = \boldsymbol{\theta}_{class} \cup \boldsymbol{\theta}_{comp}$ is the set of learning parameters. The regularization term $\lambda ||\boldsymbol{\theta}||_2^2$ indicates that the squared values of all parameters are penalized by λ; this corresponds to penalizing the

squared Frobenius norm for the matrix parameters, and the squared Euclidean norm for the vector parameters.

4.1 Learning the classification parameters

In Equation 6, $\mathcal{L}(\boldsymbol{\theta}) = 0$, if for every $y' \neq y^*$, $\psi(y^*) - \psi(y') \geq 1$ holds. Otherwise, the loss will be caused by any y', where $y' \neq y^*$ and $\psi(y^*) - \psi(y') < 1$. The gradient for the classification parameters therefore depends on the margin value between gold label and all other labels. Specifically, taking one component of $\mathbf{A}_y$, $\boldsymbol{a}_{y,1}$, as an example, the derivative of the objective for $y = y^*$ is

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{a}_{y^*,1}} = -\sum_{y':y'\neq y^*} \delta_{(\psi(y^*)-\psi(y')<1)} \cdot \boldsymbol{u}_0^{(m)}, \tag{7}$$

where $\delta_{(\cdot)}$ is the delta function. The derivative for $y' \neq y^*$ is

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{a}_{y',1}} = \delta_{(\psi(y^*)-\psi(y')<1)} \cdot \boldsymbol{u}_0^{(m)} \tag{8}$$

During learning, the updating rule for $\mathbf{A}_y$ is

$$\mathbf{A}_y \leftarrow \mathbf{A}_y - \eta(\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mathbf{A}_y} + \lambda \mathbf{A}_y) \tag{9}$$

where η is the learning rate.

Similarly, we can obtain the gradient information and updating rules for parameters $\{\mathbf{B}_y, \boldsymbol{\beta}_y, b_y\}_{y\in\mathcal{Y}}$.

4.2 Learning the composition parameters

There are two composition matrices $\mathbf{U}$ and $\mathbf{V}$, corresponding to the upward and downward composition procedures respectively. Taking the upward composition parameter $\mathbf{U}$ as an example, the derivative of $\mathcal{L}(\boldsymbol{\theta})$ with respect to $\mathbf{U}$ is

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mathbf{U}} = &\sum_{y':y'\neq y^*} \delta_{(\psi(y^*)-\psi(y')<1)} \\ &\cdot \Big(\frac{\partial \psi(y')}{\partial \mathbf{U}} - \frac{\partial \psi(y^*)}{\partial \mathbf{U}}\Big)\end{aligned} \tag{10}$$

As with the classification parameters, the derivative depends on the margin between y' and y^*.

For every $y \in \mathcal{Y}$, we have the unified derivative form,

$$\begin{aligned}\frac{\partial \psi(y)}{\partial \mathbf{U}} = &\frac{\partial \psi(y)}{\partial \boldsymbol{u}_0^{(m)}} \frac{\partial \boldsymbol{u}_0^{(m)}}{\partial \mathbf{U}} + \frac{\partial \psi(y)}{\partial \boldsymbol{u}_0^{(n)}} \frac{\partial \boldsymbol{u}_0^{(n)}}{\partial \mathbf{U}} \\ &+ \sum_{i,j\in\mathcal{A}(m,n)} \frac{\partial \psi(y)}{\partial \boldsymbol{d}_i^{(m)}} \frac{\partial \boldsymbol{d}_i^{(m)}}{\partial \mathbf{U}} \\ &+ \sum_{i,j\in\mathcal{A}(m,n)} \frac{\partial \psi(y)}{\partial \boldsymbol{d}_j^{(n)}} \frac{\partial \boldsymbol{d}_j^{(n)}}{\partial \mathbf{U}},\end{aligned} \tag{11}$$

The gradient of $\mathbf{U}$ also depends on the gradient of $\psi(y)$ with respect to every downward vector $\boldsymbol{d}$, as shown in the last two terms in Equation 11. This is because the computation of each downward vector $\boldsymbol{d}_i$ includes the upward vector of the sibling node, $\boldsymbol{u}_{s(i)}$, as shown in Equation 2. For an example, see the construction of the downward vectors for *Tina* and *she* in Figure 1b.

The partial derivatives of the decision function in Equation 11 are computed as,

$$\begin{aligned}&\frac{\partial \psi(y)}{\partial \boldsymbol{u}_0^{(m)}} = A_y \boldsymbol{u}_0^{(n)}, \frac{\partial \psi(y)}{\partial \boldsymbol{u}_0^{(n)}} = A_y^\top \boldsymbol{u}_0^{(m)}, \\ &\frac{\partial \psi(y)}{\partial \boldsymbol{d}_i^{(m)}} = B_y \boldsymbol{d}_j^{(n)}, \frac{\partial \psi(y)}{\partial \boldsymbol{d}_i^{(n)}} = B_y^\top \boldsymbol{d}_j^{(m)}, \langle i,j\rangle \in \mathcal{A}.\end{aligned} \tag{12}$$

The partial derivatives of the upward and downward vectors with respect to the upward compositional operator are computed as,

$$\frac{\partial \boldsymbol{u}_i^{(m)}}{\partial \mathbf{U}} = \sum_{\boldsymbol{u}_k^{(m)}\in\mathcal{T}(\boldsymbol{u}_i^{(m)})} \frac{\partial \boldsymbol{u}_i^{(m)}}{\partial \boldsymbol{u}_k^{(m)}} (\boldsymbol{u}_k^{(m)})^\top \tag{13}$$

and

$$\frac{\partial \boldsymbol{d}_i^{(m)}}{\partial \mathbf{U}} = \sum_{\boldsymbol{u}_k^{(m)}\in\mathcal{T}(\boldsymbol{d}_i^{(m)})} \frac{\partial \boldsymbol{d}_i^{(m)}}{\partial \boldsymbol{u}_k^{(m)}} (\boldsymbol{u}_k^{(m)})^\top, \tag{14}$$

where $\mathcal{T}(\boldsymbol{u}_m)$ is the set of all nodes in the upward composition model that help to generate $\boldsymbol{u}_m$. For example, in Figure 1a, the set $\mathcal{T}(\boldsymbol{u}_2^{(\ell)})$ includes $\boldsymbol{u}_3^{(\ell)}$ and the word representations for *Tina*, *the*, and *burger*. The set $\mathcal{T}(\boldsymbol{d}_{m,i})$ includes all the **upward** nodes involved in the downward composition model generating $\boldsymbol{d}_i^{(m)}$. For example, in Figure 1b, the set $\mathcal{T}(\boldsymbol{d}_{\text{she}}^{(r)})$ includes $\boldsymbol{u}_1^{(r)}$ and the word representations for *was* and *hungry*.

The derivative of the objective with respect to the downward compositional operator $\mathbf{V}$ is computed in a similar fashion, but it depends only on the downward nodes, $\boldsymbol{d}_i^{(m)}$.

5 Implementation

Our implementation is available online at `https://github.com/jiyfeng/updown`. Training on the PDTB takes roughly three hours to converge, on an Intel(R) Xeon(R) CPU 2.20GHz without parallel computing. Convergence is faster if the surface feature weights $\boldsymbol{\beta}$ are trained separately first. We now describe some additional details of our implementation.

Learning During learning, we used AdaGrad (Duchi et al., 2011) to tune the learning rate in each iteration. To avoid the exploding gradient problem (Bengio et al., 1994), we used the norm clipping trick proposed by Pascanu et al. (2012), fixing the norm threshold at $\tau = 5.0$.

Hyperparameters Our model includes three tunable hyperparameters: the latent dimension K for the distributed representation, the regularization parameter λ, and the initial learning rate η. All hyperparameters are tuned by randomly selecting a development set of 20% of the training data. We consider the values $K \in \{20, 30, 40, 50, 60\}$ for the latent dimensionality, $\lambda \in \{0.0002, 0.002, 0.02, 0.2\}$ for the regularization (on each training instance), and $\eta \in \{0.01, 0.03, 0.05, 0.09\}$ for the learning rate. We assign separate regularizers and learning rates to the upward composition model, downward composition model, feature model and the classification model with composition vectors.

Initialization All the classification parameters are initialized to $\mathbf{0}$. For the composition parameters, we follow Bengio (2012) and initialize $\mathbf{U}$ and $\mathbf{V}$ with uniform random values drawn from the range $[-\sqrt{6/2K}, \sqrt{6/2K}]$.

Word representations We trained a word2vec model (Mikolov et al., 2013) on the PDTB corpus, standardizing the induced representations to zero-mean, unit-variance (LeCun et al., 2012). Experiments with pre-trained GloVe word vector representations (Pennington et al., 2014) gave broadly similar results.

Syntactic structure Our model requires that the syntactic structure for each argument is represented as a binary tree. We run the Stanford parser (Klein and Manning, 2003) to obtain constituent parse trees of each sentence in the PDTB, and binarize all resulting parse trees. Argument spans in the Penn Discourse Treebank need not be sentences or syntactic constituents: they can include multiple sentences, non-constituent spans, and even discontinuous spans (Prasad et al., 2008). In all cases, we identify the syntactic subtrees within the argument span, and unify them in a right branching superstructure.

Coreference The impact of entity semantics on discourse relation detection is inherently limited by two factors: (1) the frequency with which the arguments of a discourse relation share coreferent entity mentions, and (2) the ability of automated coreference resolution systems to detect these coreferent mentions. To extract entities and their mentions from the PDTB, we ran the Berkeley coreference system (Durrett and Klein, 2013) on each document. For each argument pair, we simply ignore the non-corefential entity mentions. Line 1 in Table 2 shows the proportion of the instances with shared entities in the PDTB training and test data, as detected by the Berkeley system. As the system does not detect coreferent mentions in more than 70% of the cases, the performance improvements offered by distributed entity semantics are therefore limited. To determine whether this low rate of coreference is an intrinsic property of the data, or whether it is due to the quality of state-of-the-art coreference resolution, we also consider the gold coreference annotations in the OntoNotes corpus (Pradhan et al., 2007), a portion of which intersects with the PDTB (597 documents). Lines 2 and 3 of Table 2 give the statistics for automatic and gold coreference on this intersection. These results indicate that with perfect coreference resolution, the applicability of distributed entity semantics would reach 40% of the training set and nearly 50% of the test set. Thus, improvements in coreference resolution can be expected to yield further improvements in the effectiveness of distributed entity semantics for discourse relation detection.

Dataset	Annotation	Training (%)	Test (%)
1. PDTB	Automatic	27.4	29.1
2. PDTB∩Onto	Automatic	26.2	32.3
3. PDTB∩Onto	Gold	40.9	49.3

Table 2: Proportion of relations with coreferent entities, according to automatic coreference resolution and gold coreference annotation.

Additional features We supplement our classification model using additional surface features proposed by Lin et al. (2009). These include four categories: word pair features, constituent parse features, dependency parse features, and contextual features. As done in this prior work, we use mutual information to select features in the first three categories, obtaining 500 word pair features, 100 constituent features, and 100 dependency features. In addition, Rutherford and Xue (2014) discovered that replacing word pair with their Brown

cluster assignments could give further improvements. In our implementation, we used the Brown word clusters provided by Turian et al. (2010), in which words from the Reuters Corpus (RCV1) are grouped into 3,200 clusters. The feature selection method of Lin et al. (2009) was then used to obtain a set of 600 Brown cluster features.

6 Experiments

We evaluate our approach on the Penn Discourse Treebank (PDTB; Prasad *et al.*, 2008), which provides a discourse level annotation over the Wall Street Journal corpus. In the PDTB, each discourse relation is annotated between two argument spans. Identifying the argument spans of discourse relations is a challenging task (Lin et al., 2012), which we do not attempt here; instead, we use gold argument spans, as in most of the relevant prior work. PDTB relations may be **explicit**, meaning that they are signaled by discourse connectives (e.g., *because*); alternatively, they may be **implicit**, meaning that the connective is absent. Pitler et al. (2008) show that most explicit connectives are unambiguous, so we focus on the problem of classifying implicit discourse relations.

The PDTB provides a three-level hierarchy of discourse relations. The first level consists of four major relation **classes**: TEMPORAL, CONTINGENCY, COMPARISON and EXPANSION. For each class, a second level of **types** is defined to provide finer semantic or pragmatic distinctions; there are sixteen such relation types. A third level of **subtypes** is defined for only some types, specifying the semantic contribution of each argument.

There are two main approaches to evaluating implicit discourse relation classification. Multiclass classification requires identifying the discourse relation from all possible choices. This task was explored by Lin et al. (2009), who focus on second-level discourse relations. More recent work has emphasized binary classification, where the goal is to build and evaluate separate "one-versus-all" classifiers for each discourse relation (Pitler et al., 2009; Park and Cardie, 2012; Biran and McKeown, 2013). We primarily focus on multiclass classification, because it is more relevant for the ultimate goal of building a PDTB parser; however, to compare with recent prior work, we also evaluate on binary relation classification.

6.1 Multiclass classification

Our main evaluation involves predicting the correct discourse relation for each argument pair, from among the second-level relation types. The training and test set construction follows Lin et al. (2009) with a few changes:

- We use sections 2-20 of the PDTB as a training set, sections 0-1 as a development set for parameter tuning, and sections 21-22 for testing.
- Five relation types have a combined total of only nine instances in the training set, and are therefore excluded by Lin et al. (2009): CONDITION, PRAGMATIC CONDITION, PRAGMATIC CONTRAST, PRAGMATIC CONCESSION and EXCEPTION. None of these relations appear in the test or development data. We tried training with and without these relation types in the training data, and found no difference in the overall results.
- In the main multiclass experiment, we consider only the problem of distinguishing between implicit relations. We perform an additional, reviewer-recommended experiment that distinguishes implicit relations from entity-based coherence relations, labeled ENTREL. See below for more detail.
- Roughly 2% of the implicit relations in the PDTB are annotated with more than one type. During training, each argument pair that is annotated with two relation types is considered as two training instances, each with one relation type. During testing, if the classifier assigns either of the two types, it is considered to be correct.

6.1.1 Baseline and competitive systems

Most common class The most common class is CAUSE, accounting for 26.03% of the implicit discourse relations in the PDTB test set.

Additive word representations Blacoe and Lapata (2012) show that simply adding word vectors can perform surprisingly well at assessing the meaning of short phrases. In this baseline, we represent each argument as a sum of its word representations, and estimate a bilinear prediction matrix.

Lin et al. (2009) To our knowledge, the best published accuracy on multiclass classification

Model	+Entity semantics	+Surface features	K	Accuracy(%)
Baseline models				
1. Most common class				26.03
2. Additive word representations			50	28.73
Prior work				
3. (Lin et al., 2009)		✓		40.2
Our work				
4. Surface features + Brown clusters		✓		40.66
5. DISCO2			50	36.98
6. DISCO2	✓		50	37.63
7. DISCO2		✓	50	43.75*
8. DISCO2	✓	✓	50	44.59*

* signficantly better than lines 3 and 4 with $p < 0.05$

Table 3: Experimental results on multiclass classification of level-2 discourse relations. The results of Lin et al. (2009) are shown in line 3. We reimplemented this system and added the Brown cluster features of Rutherford and Xue (2014), with results shown in line 4.

of second-level implicit discourse relations is from Lin et al. (2009), who apply feature selection to obtain a set of lexical and syntactic features over the arguments.

Surface features + Brown clusters To get a more precise comparison, we reimplemented the system of Lin et al. (2009). The major differences are (1) we apply our online learning framework, rather than batch classification, and (2) we include the Brown cluster features described in Section 5 and originally proposed by Rutherford and Xue (2014).

Compositional Finally, we report results for the method described in this paper. Since it is a **dis**tributional **co**mpositional approach to **disco**urse relations, we name it DISCO2.

6.1.2 Results

Table 3 presents results for multiclass identification of second-level PDTB relations. As shown in lines 7 and 8, DISCO2 outperforms both baseline systems and the prior state-of-the-art (line 3). The strongest performance is obtained by including the entity distributed semantics, with a 4.4% improvement over the accuracy reported by Lin et al. (2009) ($p < .05$ by a binomial test). We also obtain a significant improvement over the Surface Feature + Brown Cluster model. Because we have reimplemented this system, we can observe individual predictions, and can therefore use the sign test for statistical significance, again finding that DISCO2 is significantly better ($p < .05$). Even without entity semantics, DISCO2 significantly outperforms these competitive models from prior work. However, the surface features remain important, as the performance of DISCO2 is substantially worse when only the distributed representation is included. The latent dimension K is chosen from a development set (see Section 5), as shown in Figure 3.

The multiclass evaluation introduced by Lin et al. (2009) focused on classification of implicit relations. Another question is whether it is possible to identify entity-based coherence, annotated in the PDTB as ENTREL, which is when a shared entity is the only meaningful relation that holds between two sentences (Prasad et al., 2008). As suggested by a reviewer, we add ENTREL to the set of possible relations, and perform an additional evaluation. Since this setting has not previously been considered, we cannot evaluate against published results; instead, we retrain and evaluate the following models:

- the surface feature baseline with Brown clusters, corresponding to line 4 of Table 3;
- DISCO2 with surface features but without entity semantics, corresponding to line 7 of Table 3;

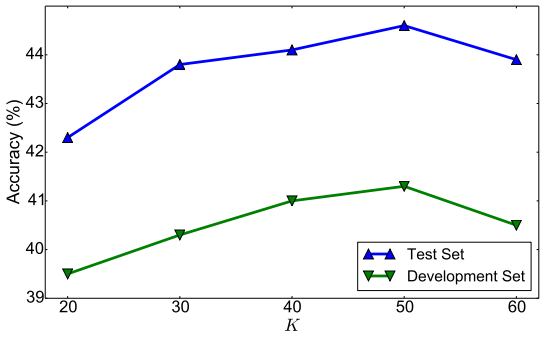


Figure 3: The performance of DISCO2 (full model), over different latent dimensions K.

- DISCO2 with surface features and entity semantics, corresponding to line 8 of Table 3.

As before, all parameters are tuned on a development set. In this evaluation, we obtain larger improvements from our approach: our full model (with entity semantics) gives 47.27% accuracy, as compared to 44.96% without entity semantics; the result for the surface feature baseline is 41.48%.

6.1.3 Coreference

The contribution of entity semantics is shown in Table 3 by the accuracy differences between lines 5 and 6, and between lines 7 and 8. On the subset of relations in which the arguments share at least one coreferent entity, the difference is substantially larger: the accuracy of DISCO2 is 45.7% with entity mention semantics, and 43.1% without. Considering that only 29.1% of the relations in the PDTB test set include shared entities, it therefore seems likely that a more sensitive coreference system could yield further improvements for the entity-semantics model. Indeed, gold coreference annotation on the intersection between the PDTB and the OntoNotes corpus shows that 40-50% of discourse relations involve coreferent entities (Table 2). Evaluating on just this intersection, we find that the inclusion of entity semantics yields an improvement in accuracy from 37.5% to 39.1%. Thus, while the overall improvements offered by entity mention semantics are relatively small, this is due in part to the poor recall of the state-of-the-art coreference resolution system; if coreference improved, the impact of the entity mention semantics would increase correspondingly.

A reviewer asked whether it was necessary to have the correct coreference alignment, or whether similar improvements could be obtained by computing bilinear products between all pairs of noun phrases in the two discourse arguments. In fact, this strategy of aligning all entity mentions resulted in a decrease in accuracy, from 44.59 to 42.14%. This is below the performance of DISCO2 without entity semantics.

6.1.4 Examples

The following examples help highlight how entity semantics can improve the accuracy of discourse relation classification.

(3) **Arg 1**: *The drop in profit reflected, in part, continued softness in financial advertising at [The Wall Street Journal] and Barron's magazine.*
Arg 2: *Ad linage at [the Journal] fell 6.1% in the third quarter.*

(4) **Arg 1**: *[Mr. Greenberg] got out just before the 1987 crash and, to [his] regret, never went back even as the market soared.*
Arg 2: *This time [he]'s ready to buy in "when the panic wears off."*

(5) **Arg 1**: *Half of [them]$_1$ are really scared and want to sell but [I]$_2$'m trying to talk them out of it.*
Arg 2: *If [they]$_1$ all were bullish, [I]$_2$'d really be upset.*

In example (3), the entity-augmented model correctly identifies the relation as RESTATEMENT, due in part to the detected coreference between *The Wall Street Journal* and *the Journal*: in both arguments, the entity experiences a drop in profits. Without this information, DISCO2 incorrectly labels this relation as CAUSE. In example (4), the entity-augmented model correctly identifies the relation as CONTRAST, which is reasonable given the very different role of the shared entity *Mr. Greenberg* in the two arguments; without entity semantics, it is classified as CONJUNCTION. Example (5) is more complex because it involves two entities, but again, the CONTRAST relation is correctly detected, in part because of the differing experiences of the two entities in the two arguments; without entity semantics, this example is again incorrectly classified as CONJUNCTION.

6.2 Binary classification

Much of the recent work in PDTB relation detection has focused on binary classification, building

and evaluating separate one-versus-all classifiers for each relation type (Pitler et al., 2009; Park and Cardie, 2012; Biran and McKeown, 2013). This work has focused on recognition of the four *first*-level relations, grouping ENTREL with the EXPANSION relation. We follow this evaluation approach as closely as possible, using sections 2-20 of the PDTB as a training set, sections 0-1 as a development set for parameter tuning, and sections 21-22 for testing.

6.2.1 Classification method

We apply DISCO2 with entity-augmented semantics and the surface features listed in Section 5; this corresponds to the system reported in line 8 of Table 3. However, instead of employing a multiclass classifier for all four relations, we train four binary classifiers, one for each first-level discourse relation. We optimize the hyperparameters K, λ, η separately for each classifier (see Section 5 for details), by performing a grid search to optimize the F-measure on the development data. Following Pitler et al. (2009), we obtain a balanced training set by resampling training instances in each class until the number of positive and negative instances are equal.

6.2.2 Competitive systems

We compare against the published results from several competitive systems, focusing on systems which use the predominant training / test split, with sections 2-20 for training and 21-22 for testing. This means we cannot compare with recent work from Li and Nenkova (2014), who use sections 20-24 for testing.

Pitler et al. (2009) present a classification model using linguistically-informed features, such as polarity tags and Levin verb classes.

Zhou et al. (2010) predict discourse connective words, and then use these predicted connectives as features in a downstream model to predict relations.

Park and Cardie (2012) showed that the performance on each relation can be improved by selecting a locally-optimal feature set.

Biran and McKeown (2013) reweight word pair features using distributional statistics from the Gigaword corpus, obtaining denser aggregated score features.

6.2.3 Experimental results

Table 4 presents the performance of the DISCO2 model and the published results of competitive systems. DISCO2 achieves the best results on most metrics, achieving F-measure improvements of 4.14% on COMPARISON, 2.96% on CONTINGENCY, 0.8% on EXPANSION, and 1.06% on TEMPORAL. These results are attained without performing per-relation feature selection, as in prior work. While computing significance over F-measures is challenging, we can compute statistical significance on the accuracy results by using the binomial test. We find that DISCO2 is significantly more accurate than all other systems on the CONTINGENCY and TEMPORAL relations $p \ll .001$, not significantly more accurate on the EXPANSION relation, and significantly less accurate than the Park and Cardie (2012) system on the COMPARISON relation at $p \ll .001$.

7 Related Work

This paper draws on previous work in discourse relation detection and compositional distributed semantics.

7.1 Discourse relations

Many models of discourse structure focus on relations between spans of text (Knott, 1996), including rhetorical structure theory (RST; Mann and Thompson, 1988), lexicalized tree-adjoining grammar for discourse (D-LTAG; Webber, 2004), and even centering theory (Grosz et al., 1995), which posits relations such as CONTINUATION and SMOOTH SHIFT between adjacent spans. Consequently, the automatic identification of discourse relations has long been considered a key component of discourse parsing (Marcu, 1999). We work within the D-LTAG framework, as annotated in the Penn Discourse Treebank (PDTB; Prasad et al., 2008), with the task of identifying *implicit* discourse relations. The seminal work in this task is from Pitler et al. (2009) and Lin et al. (2009). Pitler et al. (2009) focus on lexical features, including linguistically motivated word groupings such as Levin verb classes and polarity tags. Lin et al. (2009) identify four different feature categories, based on the raw text, the context, and syntactic parse trees; the same feature sets are used in later work on end-to-end discourse parsing (Lin et al., 2012), which also includes components for identifying argument spans. Subsequent

	COMPARISON		CONTINGENCY		EXPANSION		TEMPORAL	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Competitive systems								
1. (Pitler et al., 2009)	21.96	56.59	47.13	67.30	76.42	63.62	16.76	63.49
2. (Zhou et al., 2010)	31.79	58.22	47.16	48.96	70.11	54.54	20.30	55.48
3. (Park and Cardie, 2012)	31.32	**74.66**	49.82	72.09	79.22	69.14	26.57	79.32
4. (Biran and McKeown, 2013)	25.40	63.36	46.94	68.09	75.87	62.84	20.23	68.35
Our work								
5. DISCO2	**35.93**	70.27	**52.78**	**76.95**	**80.02**	**69.80**	**27.63**	**87.11**

Table 4: Evaluation on the first-level discourse relation identification. The results of the competitive systems are reprinted.

research has explored feature selection (Park and Cardie, 2012; Lin et al., 2012), as well as combating feature sparsity by aggregating features (Biran and McKeown, 2013). Our model includes surface features that are based on a reimplementation of the work of Lin et al. (2009), because they also undertake the task of multiclass relation classification; however, the techniques introduced in more recent research may also be applicable and complementary to the distributed representation that constitutes the central contribution of this paper; if so, applying these techniques could further improve performance.

Our contribution of entity-augmented distributed semantics is motivated by the intuition that entities play a central role in discourse structure. Centering theory draws heavily on referring expressions to entities over the discourse (Grosz et al., 1995; Barzilay and Lapata, 2008); similar ideas have been extended to rhetorical structure theory (Corston-Oliver, 1998; Cristea et al., 1998). In the specific case of PDTB relations, Louis et al. (2010b) explore a number of entity-based features, including grammatical role, syntactic realization, and information status. Despite the solid linguistic foundation for these features, they are shown to contribute little in comparison with more traditional word-pair features. This suggests that syntax and information status may not be enough, and that it is crucial to capture the semantics of each entity's role in the discourse. Our approach does this by propagating distributed semantics from throughout the sentence into the entity span, using our up-down compositional procedure. In recent work, Rutherford and Xue (2014) take an alternative approach, using features that represent whether coreferent mentions are arguments of similar predicates (using Brown clusters); they obtain nearly a 1% improvement on CONTINGENCY relations but no significant improvement on the other three first-level relation types. Finally, Kehler and Rohde (2013) show that information also flows in the opposite direction, from discourse relations to coreference: in some cases, knowing the discourse relation is crucial to resolving pronoun ambiguity. Future work should therefore consider joint models of discourse analysis and coreference resolution.

7.2 Compositional distributed semantics

Distributional semantics begins with the hypothesis that words and phrases that tend to appear in the same contexts have the same meaning (Firth, 1957). The current renaissance of interest in distributional and distributed semantics can be attributed in part to the application of discriminative techniques, which emphasize *predictive* models (Bengio et al., 2006; Baroni et al., 2014b), rather than context-counting and matrix factorization (Landauer et al., 1998; Turney and Pantel, 2010). Recent work has made practical the idea of propagating distributed information through linguistic structures (Smolensky, 1990; Collobert et al., 2011). In such models, the distributed representations and compositional operators can be fine-tuned by backpropagating supervision from task-specific labels, enabling accurate and fast models for a wide range of language technologies (Socher et al., 2011; Socher et al., 2013; Chen and Manning, 2014).

Of particular relevance is recent work on two-pass procedures for distributed compositional semantics. Paulus et al. (2014) perform targeted sentiment analysis by propagating information from

the sentence level back to child non-terminals in the parse tree. Their compositional procedure is different from ours: in their work, the "downward" meaning of each non-terminal is reconstructed from the upward and downward meanings of its parents. İrsoy and Cardie (2013) propose an alternative two-pass procedure, where the downward representation for a node is computed from the downward representation of its parent, and from its *own* upward representation. A key difference in our approach is that the siblings in a production are more directly connected: the upward representation of a given node is used to compute the downward representation of its sibling, similar to the inside-outside algorithm. In the models of Paulus et al. (2014) and İrsoy and Cardie (2013), the connection between siblings nodes is less direct, as it is channeled through the representation of the parent node. From this perspective, the most closely related prior work is the Inside-Outside Recursive Neural Network (Le and Zuidema, 2014), published shortly before this paper was submitted. The compositional procedure in this paper is identical, although the application is quite different: rather than inducing distributed representations of entity mentions, the goal of this work is to support an infinite-order generative model of dependency parsing. While Le and Zuidema apply this idea as a generative reranker within a supervised dependency parsing framework, we are interested to explore whether it could be employed to do unsupervised syntactic analysis, which could substitute for the supervised syntactic parser in our system.

The application of distributional and distributed semantics to discourse includes the use of latent semantic analysis for text segmentation (Choi et al., 2001) and coherence assessment (Foltz et al., 1998), as well as paraphrase detection by the factorization of matrices of distributional counts (Kauchak and Barzilay, 2006; Mihalcea et al., 2006). These approaches essentially compute a distributional representation in advance, and then use it alongside other features. In contrast, our approach follows more recent work in which the distributed representation is driven by supervision from discourse annotations. For example, Ji and Eisenstein (2014) show that RST parsing can be performed by learning task-specific word representations, which perform considerably better than generic word2vec representations (Mikolov et al., 2013). Li et al. (2014) propose a recursive neural network approach to RST parsing, which is similar to the upward pass in our model, and Kalchbrenner and Blunsom (2013) show how a recurrent neural network can be used to identify dialogue acts. However, prior work has not applied these ideas to the classification of implicit relations in the PDTB, and does not consider the role of entities. As we argue in the introduction, a single vector representation is insufficiently expressive, because it obliterates the entity chains that help to tie discourse together.

More generally, our entity-augmented distributed representation can be viewed in the context of recent literature on combining distributed and formal semantics: by representing entities, we are taking a small step away from purely vectorial representations, and towards more traditional logical representations of meaning. In this sense, our approach is "bottom-up", as we try to add a small amount of logical formalism to distributed representations; other approaches are "top-down", softening purely logical representations by using distributional clustering (Poon and Domingos, 2009; Lewis and Steedman, 2013) or Bayesian non-parametrics (Titov and Klementiev, 2011) to obtain types for entities and relations. Still more ambitious would be to implement logical semantics within a distributed compositional framework (Clark et al., 2011; Grefenstette, 2013). At present, these combinations of logical and distributed semantics have been explored only at the sentence level. In generalizing such approaches to multi-sentence discourse, we argue that it will not be sufficient to compute distributed representations of sentences: a multitude of other elements, such as entities, will also have to represented.

8 Conclusion

Discourse relations are determined by the meaning of their arguments, and progress on discourse parsing therefore requires computing representations of the argument semantics. We present a compositional method for inducing distributed representations not only of discourse arguments, but also of the entities that thread through the discourse. In this approach, semantic composition is applied up the syntactic parse tree to induce the argument-level representation, and then down the parse tree to induce representations of entity

spans. Discourse arguments can then be compared in terms of their overall distributed representation, as well as by the representations of coreferent entity mentions. This enables the compositional operators to be learned by backpropagation from discourse annotations. In combination with traditional surface features, this approach outperforms previous work on classification of implicit discourse relations in the Penn Discourse Treebank. While the entity mention representations offer only a small improvement in overall performance, we show that this is limited by the recall of the coreference resolution system: when evaluated on argument pairs for which coreference is detected, the raw improvement from entity semantics is more than 2%. Future work will consider joint models of discourse structure and coreference, and consideration of coreference across the entire document. In the longer term, we hope to induce and exploit representations of other discourse elements, such as event coreference and shallow semantics.

Acknowledgments This work was supported by a Google Faculty Research Award to the second author. Chris Dyer, Ray Mooney, and Bonnie Webber provided helpful feedback, as did the anonymous reviewers and the editor. Thanks also to Te Rutherford, who both publicly released his code and helped us to use and understand it.

References

Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. 2014a. Frege in space: A program for compositional distributional semantics. *Linguistic Issues in Language Technologies*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014b. Dont count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the Association for Computational Linguistics (ACL)*, Baltimore, MD.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An Entity-Based approach. *Computational Linguistics*, 34(1):1–34, March.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer.

Or Biran and Kathleen McKeown. 2013. Aggregated word pair features for implicit discourse relation disambiguation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 69–73, Sophia, Bulgaria.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 546–556.

Léon Bottou. 1998. Online learning and stochastic approximations. *On-line learning in neural networks*, 17:9.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 740–750.

Freddy YY Choi, Peter Wiemer-Hastings, and Johanna Moore. 2001. Latent semantic analysis for text segmentation. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. 2011. Mathematical foundations for a compositional distributed model of meaning. *Linguistic Analysis*, 36(1-4):345–384.

Shay B Cohen, Karl Stratos, Michael Collins, Dean P Foster, and Lyle Ungar. 2014. Spectral learning of latent-variable PCFGs: Algorithms and sample complexity. *Journal of Machine Learning Research*, 15:2399–2449.

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

Simon Corston-Oliver. 1998. Beyond string matching and cue phrases: Improving efficiency and coverage in discourse analysis. In *The AAAI Spring Symposium on Intelligent Text Summarization*, pages 9–15.

Dan Cristea, Nancy Ide, and Laurent Romary. 1998. Veins theory: A model of global discourse cohesion and coherence. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 281–285. Association for Computational Linguistics.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

J. R. Firth. 1957. *Papers in Linguistics 1934-1951*. Oxford University Press.

Peter W Foltz, Walter Kintsch, and Thomas K Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse processes*, 25(2-3):285–307.

Katherine Forbes-Riley, Bonnie Webber, and Aravind Joshi. 2006. Computing discourse semantics: The predicate-argument semantics of discourse connectives in d-ltag. *Journal of Semantics*, 23(1):55–106.

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, IEEE International Conference on*, pages 347–352. IEEE.

Edward Grefenstette. 2013. Towards a Formal Distributional Semantics: Simulating Logical Calculi with Tensors, April.

Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225.

Ozan İrsoy and Claire Cardie. 2013. Bidirectional recursive neural networks for token-level labeling with structure. *CoRR (presented at the 2013 NIPS Workshop on Deep Learning)*, abs/1312.0493.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the Association for Computational Linguistics (ACL)*, Baltimore, MD.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the Association for Computational Linguistics (ACL)*, Baltimore, MD.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, Sofia, Bulgaria, August. Association for Computational Linguistics.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of NAACL*, pages 455–462. Association for Computational Linguistics.

Andrew Kehler and Hannah Rohde. 2013. A probabilistic reconciliation of coherence-driven and centering-driven theories of pronoun interpretation. *Theoretical Linguistics*, 39(1-2):1–37.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 423–430.

Alistair Knott. 1996. *A data-driven methodology for motivating a set of coherence relations*. Ph.D. thesis, The University of Edinburgh.

Thomas Landauer, Peter W. Foltz, and Darrel Laham. 1998. Introduction to Latent Semantic Analysis. *Discource Processes*, 25:259–284.

Karim Lari and Steve J Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56.

Phong Le and Willem Zuidema. 2014. The inside-outside recursive neural network model for dependency parsing. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 729–739.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer.

Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Junyi Jessy Li and Ani Nenkova. 2014. Reducing sparsity improves the recognition of implicit discourse relations. In *Proceedings of SIGDIAL*, pages 199–207.

Jiwei Li, Rumeng Li, and Eduard Hovy. 2014. Recursive deep models for discourse parsing. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 343–351, Singapore.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically Evaluating Text Coherence Using Discourse Relations. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 997–1006, Portland, OR.

Ziheng Lin, Hwee T. Ng, and Min Y. Kan. 2012. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, FirstView:1–34, November.

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010a. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 147–156. Association for Computational Linguistics.

Annie Louis, Aravind Joshi, Rashmi Prasad, and Ani Nenkova. 2010b. Using entity features to classify implicit discourse relations. In *Proceedings of the SIGDIAL*, pages 59–62, Tokyo, Japan, September. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

William Mann. 1984. Discourse structures for text generation. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*, pages 367–375. Association for Computational Linguistics.

Daniel Marcu. 1999. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 365–372. Association for Computational Linguistics.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 746–751, Stroudsburg, Pennsylvania. Association for Computational Linguistics.

Raymond J. Mooney. 2014. Semantic parsing: Past, present, and future. Presentation slides from the ACL Workshop on Semantic Parsing.

Joonsuk Park and Claire Cardie. 2012. Improving Implicit Discourse Relation Recognition Through Feature Set Optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 108–112, Seoul, South Korea, July. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*.

Romain Paulus, Richard Socher, and Christopher D Manning. 2014. Global belief recursive neural networks. In *Neural Information Processing Systems (NIPS)*, pages 2888–2896, Montréal.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 1532–1543.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind Joshi. 2008. Easily identifiable discourse relations. In *Coling 2008: Companion volume: Posters*, pages 87–90, Manchester, UK, August. Coling 2008 Organizing Committee.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic Sense Prediction for Implicit Discourse Relations in Text. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP* .

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 1–10, Singapore.

Sameer S Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. Ontonotes: A unified relational semantic representation. *International Journal of Semantic Computing*, 1(04):405–419.

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*.

Attapol T Rutherford and Nianwen Xue. 2014. Discovering Implicit Discourse Relations Through Brown Cluster Pair Representation and Coreference Patterns. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, Stroudsburg, Pennsylvania. Association for Computational Linguistics.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1):159–216.

Richard Socher, Cliff C. Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, Seattle, WA.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2014. Reasoning With Neural Tensor Networks For Knowledge Base Completion. In *Neural Information Processing Systems (NIPS)*. Lake Tahoe.

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of EMNLP*.

Ivan Titov and Alexandre Klementiev. 2011. A bayesian model for unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1445–1455. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word Representation: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 384–394, Uppsala, Sweden.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *JAIR*, 37:141–188.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2759–2605, November.

Bonnie Webber, Alistair Knott, Matthew Stone, and Aravind Joshi. 1999. Discourse relations: A structural and presuppositional account using lexicalised tag. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 41–48.

Bonnie Webber. 2004. D-LTAG: extending lexicalized TAG to discourse. *Cognitive Science*, 28(5):751–779, September.

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. Dependency-based Discourse Parser for Single-Document Summarization. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. 2010. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1507–1514.