Global Features for Shallow Discourse Parsing

Sucheta Ghosh Giuseppe Riccardi
University of Trento, Italy
{ghosh,riccardi}@disi.unitn.it

Richard Johansson
University of Gothenburg, Sweden
richard.johansson@gu.se

Abstract

A coherently related group of sentences may be referred to as a discourse. In this paper we address the problem of parsing coherence relations as defined in the Penn Discourse Tree Bank (PDTB). A good model for discourse structure analysis needs to account both for local dependencies at the token-level and for global dependencies and statistics. We present techniques on using inter-sentential or sentence-level (global), data-driven, non-grammatical features in the task of parsing discourse. The parser model follows up previous approach based on using token-level (local) features with conditional random fields for shallow discourse parsing, which is lacking in structural knowledge of discourse. The parser adopts a two-stage approach where first the local constraints are applied and then global constraints are used on a reduced weighted search space (n-best). In the latter stage we experiment with different rerankers trained on the first stage n-best parses, which are generated using lexico-syntactic local features. The two-stage parser yields significant improvements over the best performing model of discourse parser on the PDTB corpus.

1 Introduction

There are relevant studies on the impact of global and local features on the models for natural language understanding. In this work we address a similar problem in the context of discourse parsing. Although a good number of the papers in this area heavily rely on local classifiers (Grosz et al., 1995; Soricut et al., 2003; Lapata, 2003; Barzilay et al., 2005), there are still some important works using global and local informations together to form a model of discourse (Grosz et al., 1992; Barzilay et al., 2004; Soricut et al., 2006).

One of the main issues is the basis of the choice between a global or local or a joint model for discourse parsing: it all depends on the criteria to be able to capture maximum amount of information inside the discourse model. The policy for discourse segmentation plays a big role to formulate the maximizing criteria (Grosz et al., 1992). We study in the literature that defining a discourse segment is mostly a data-driven process: some argue for prosodic units, some for intentional structure and some for clause-like structures. We work with PDTB 2.0 annotation framework, therefore use a clause-like structure. Soricut et al. (2003) empirically showed that at the sentence level, there is a strong correlation between syntax and discourse, Ghosh et al. (2011b) found the same. Since the discourse structure may span over multiple sentences, intersentential features are needed to improve the performance of a discourse parser.

Linguistic theory suggests that a core argument frame (i.e. a pair of the `Arg1` and the `Arg2` connected with one and only one connective) is a joint structure, with strong dependencies between arguments (Toutanova et al., 2008). Following this, Ghosh et al. (2011a) also injected some structure-level information through the token-level features, for eg. the previous sentence feature. Still there is a room for improvement with more structure-level information to that discourse model; though it is cost-intensive to modify this discourse model. Therefore in this paper we re-use the model (Ghosh et al., 2011a) and optimize the current loss function adding the global features through re-ranking of the single-best model.

Reranking has been a popular technique applied in a variety of comparable NLP problems including parsing (Collins, 2000;

Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pages 150–159,
Seoul, South Korea, 5-6 July 2012.

Charniak and Johnson, 2005), semantic role labeling (Toutanova et al., 2008), NP Bracketing (Daume III et al., 2004), NER (Collins, 2002), opinion expression detection (Johansson and Moschitti, 2010), now we employ this technique in the area of discourse parsing.

In the next sections, we detail on the backgrounds and motivations of this work, before this we also add a short discussion on PDTB (Penn Discourse TreeBank), i.e. the data we used to train the system. Then we proceed to the reranking approaches and results sections after describing our global feature set. Finally we state and analyze the results.

2 The Penn Discourse Treebank 2.0

The Penn Discourse Treebank (PDTB) is a resource containing one million words from the Wall Street Journal corpus (Marcus et al., 1993) annotated with discourse relations.

Connectives in the PTDB are treated as discourse predicates taking two text spans as *arguments* (`Arg`), i.e. parts of the text that describe events, propositions, facts, situations. Such two arguments in the PDTB are called `Arg1` and `Arg2`, with the numbering not necessarily corresponding to their order in text. Indeed, `Arg2` is the argument syntactically bound to the connective, while `Arg1` is the other one.

In the PDTB, discourse relations can be either overtly or implicitly expressed. However, we focus here exclusively on *explicit* connectives and the identification of their arguments, including the exact spans. This kind of classification is very complex, since `Arg1` and `Arg2` can occur in many different configurations (see Table 1).

Explicit connectives (tokens)	18,459
Explicit connectives (types)	100
`Arg1` in same sentence as connective	60.9%
`Arg1` in previous, adjacent sentence	30.1%
`Arg1` in previous, non adjacent sentence	9.0%

Table 1: Statistics about PDTB annotation from Prasad et al(2008).

In PDTB the senses are assigned according to a three-layered hierarchy: the top-level classes are the most generic ones and include TEMPORAL, CONTINGENCY, COMPARISON and EXPANSION labels. We used these four surface senses only in our task.

2.1 Backgrounds & Motivation

Currently we are using the single-best discourse parser by Ghosh et al. (2011a). This discourse parser can automatically extract of discourse arguments using a pipeline, illustrated in Fig 1. First, we input the explicit discourse connectives (with senses) to the system. These can be the gold labeled or automatically identified (Pitler and Nenkova, 2009); for simplicity here we use Penn Discourse TreeBank (PDTB 2.0) gold-standard connectives (*cf.* see 2). Then a cascaded module is applied extracting the `Arg2` arguments, then the `Arg1`s are extracted.

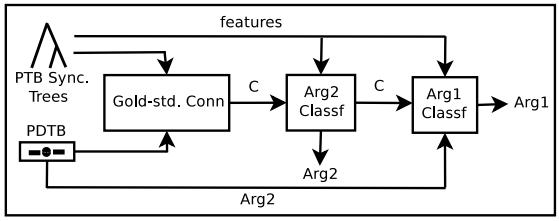

Figure 1: Pipeline for argument detection given a connective.

The `Arg2` and `Arg1` extractors are implemented as conditional random field sequence labelers, which use a set of syntactic and structural features (*cf.* Ghosh et al. (2011a)). In order to reduce the complexities, the sentence containing the connective, and a context window of up to two sentences before and after are supplied to the sequence labelers.

We present a passage of 6 sentences from a nutrition journal article parsed with that parser [1].:

```
<Conn id=1,sense=Comparison>
Although</Conn id=1> <ARG2 id=1>
the mechanism of obesity development
is not fully understood, it is confirmed
<ARG1 id=2>that obesity occurs</ARG1 id=2>
<Conn id=2,sense=Temporal>when</Conn id=2>
<ARG2 id=2>energy intake exceeds energy
expenditure</ARG2 id=2> </ARG2 id=1>.
There are multiple etiologies
for this imbalance, hence,
<Conn id=3, sense=Expansion>
and </Conn id=3> <ARG2 id=3>the rising
prevalence of obesity cannot be addressed
by a single etiology</ARG2 id=3>.
<ARG1 id=4>Genetic factors influence
the susceptibility of a given child to an
obesity-conducive environment</ARG1 id=4>.
<Conn id=4, sense=Comparison>However
```

[1] we used best model of (Ghosh et al., 2011b; Ghosh et al., 2011a) and Stanford lexicalized parser (Klein and Manning, 2003) to parse the text also used AddDiscourse tool to parse the connective and the senses (Pitler and Nenkova, 2009);parser took 17 second to parse

</Conn id=4>, <ARG2 id=4>**environmental factors, lifestyle preferences, and cultural environment seem to play major roles in the rising prevalence of obesity worldwide**</ARG2 id=4>. In a small number of cases, childhood obesity is due to genes such as leptin deficiency or medical causes such as hypothyroidism and growth hormone deficiency or side effects due to drugs (e.g. - steroids). Most of the time, <Conn id=5, sense= Comparison> ***however*** </Conn id=5>, <ARG2 id=5>**personal lifestyle choices and cultural environment significantly influence obesity**</ARG2 id=5>.

In the evaluations of Ghosh et al. (2011a), it states that recall was much lower than precision for both the arguments, especially in case of Arg1. The system often failed to predict Arg1. It is harder to identify since it is not always syntactically bound to the connective, like Arg2, moreover it is typically more distant than the Arg2s.

We notice the same in the parser output. The parser found all five Arg2s for all five connectives, though there may be disagreement on the selected boundaries; the number of parsed Arg1s is only two, whereas the second one with id of 4 is a previous sentence argument.

To improve the recall, (Ghosh et al., 2012) implemented a weighted constraint-based *handcrafted* postprocessor to force the Ghosh et al. (2011a) system to output arguments of each type abiding the requirements defined by the PDTB annotation guidelines.

In order to find the best solution with a minimum of constraint violations, the top k analyses output are generated by the CRF (Conditional Random Field) (Lafferty et al., 2001) for every sentence; these analyses can then be combined to form the k top analyses for the whole 5-sentence window around the connective. This combination is most efficiently carried out using a priority queue similar to a chart cell in the k-best parsing algorithm by Huang and Chiang (2005). (see Ghosh et al. (2012) for details)

2.2 Feature Set of Baseline System

We summarize the feature set of the base system (Ghosh et al., 2011a) to emphasize the distinction between the local and global feature set for this work.

The token-level (local) feature set in the Table 2 can be divided into four categories:

Features used for Arg1 and Arg2 segmentation and labeling.	
F1.	Token (T)
F2.	Sense of Connective (CONN)
F3.	IOB chain (IOB)
F4.	PoS tag
F5.	Lemma (L)
F6.	Inflection (INFL)
F7.	Main verb of main clause (MV)
F8.	Boolean feature for MV (BMV)
F9.	Previous sentence feature (PREV)
Additional feature used only for Arg1	
F10.	Arg2 Labels

Table 2: Feature sets for Arg1 and Arg2 segmentation and labeling in base system (Ghosh et al 2011a).

1. Syntactic. $\{F3, F4, F6\}$ [2]
2. Semantic. $\{F2\}$
3. Lexical $\{F5, F7, F8\}$
4. Structure related token-level features. $\{F9, F10\}$

The remaining one (F1) is the token itself. The sense of the connective feature (F2) extracted from PDTB for the base system, though for the fully automatic one (Ghosh et al., 2011b) it needs the PTB (Penn TreeBank)-style syntactic parse trees as input (Pitler and Nenkova, 2009). The IOB(Inside-Outside-Begin) chain (F3) [3] (F3) is extracted from a full parse tree and corresponds to the syntactic categories of all the constituents on the path between the root note and the current leaf node of the tree. Experiments with other syntactic features proved that IOB chain conveys all deep syntactic information needed in the task, and makes all other syntactic information redundant, for example clause boundaries, token distance from the connective, constituent label, etc.

In order to extract the morphological features needed, we use the *morpha* tool (Minnen et al., 2001), which outputs lemma (F5) and inflection information (F6) of the candidate token. The latter is the ending usually added to the word root to convey inflectional information. It includes for example the *-ing* and *-ed* suffixes in verb endings as well as the *-s* to form the plural of nouns.

As for features (F7) and (F8), they rely on information about the main verb of the current sentence. More specifically, feature (F7) is the main verb token , extracted following the head-finding

[2] Infection can be defined as morpho-syntactic feature.

[3] We extracted this feature using the Chunklink.pl script made available by Sabine Buchholz at ilk.uvt.nl/team/sabine/chunklink/README.html

strategy by Yamada and Matsumoto (2003), while feature (F8) is a boolean feature that indicates for each token if it is the main verb in the sentence or not.[4]

The structure related token-level features do not use any parse tree. The `Arg2` label (F10) features are generated from the word sequence index in PDTB for the base system (for automatic system it is generated by the pipeline (Ghosh et al., 2011b)); this feature is used to classify `Arg1` . The previous sentence feature "Prev" (F9) is a connective-surface feature and is used to capture if the following sentence begins with a connective. This is meant for the classification of the `Arg1` that resides in the previous sentence of the connective. The feature value for each candidate token of a sentence corresponds to the connective token that appears at the beginning of the following sentence, if any. Otherwise, it is equal to 0.

Although both of the structure-related features are strong features according to the feature analysis in Ghosh et al. (2011a), the base system is not able to capture all available global features inside the 5-sentence discourse context, merely uses 2-sentence context. This is due to the fact that CRF classifier uses a narrow window, that can only capture the information nearby the token under consideration. Therefore it becomes impossible to inject more information about the 5-sentence discourse window structure.

3 Global Feature Set

We use a global feature-set. The global features are defined as the data-driven, hand-crafted rule generated and non-grammatical (i.e. no syntactic parse tree is used to generate this features) features.

The model of Ghosh et al. (2011a) is based on Conditional Random Fields (CRF), and incorporating a set of structural and lexical features. At the core part of the model lies a local classifier, which labels each token sequentially with one of the possible argument labels or OTHER in a pipeline. Now global information can be integrated into the model using global features at a longer-distance context, by defining a small set of global constraints (if too many dependencies are encoded, the model will over-fit the training data and will not generalize well).

The global features are computed using each list of k-best lists, in contrast to the lexico-syntactically generated local features for each token item for each sentence of n-best lists. The usage of global feature is meant for exploring the yet undiscovered dimension of the each 5-sentence discourse window. Global feature set consists of the eight features that works on a full 5-sentence discourse window (*cf.* sec. 2.1). The first six (i.e. GF0-GF5) of these are same with the constrained system 2.1.

None of the features are extracted from any parse tree. All the seven features (GF1-GF7) are derived from the generated `Arg` tags of the n-best lists, the first one is the logarithm of posterior probability computed from the CRF posterior probability output for each list of the n-best lists. The finer description of each feature is given below.

GF0. *logarithm of Posterior Probability*. this feature is generated by the base CRF classifier. The CRF generates probability per sentence, for each list of the n-best lists. We calculate sum of the log of each probability during generation of k-best lists forming 5-sentence discourse window.

GF1. *Overgeneration*. It is possible for an argument to be split into more than one part in same sentence, we found these cases several times in PDTB. This constraint is violated if an `Arg1` or `Arg2` is split over multiple sentences. This is a predominant problem for those lists of the n-best lists those are generated with low posteriors. This feature exhibits the problem of overgeneration to the reranker with the counts.

GF2. *Undergeneration*. According to PDTB annotation scheme every connective must have arguments of each type, this constraint is violated if an argument is missing. This is the prevalent problem in the single-best system, especially for the `Arg1` classification. This feature works to specify where a discourse structure missing the argument(s) - one of the main problems that motivated this work.

GF3. *Intersentential Arg2* (used only for `Arg2` reranker). Count of `Arg2`, if any, occurs classified outside connective sentence - this way the system is constrained to have any inter-sentential `Arg2`. This is a hypothetically motivated feature to reduce the complexity of the classification problem; although in fact in PDTB 2.0, there are a few cases

[4] We used the head rules by Yamada & Matsumoto (`http://www.jaist.ac.jp/~h-yamada/`)

of Arg2 of explicit connective (i.e. the 114 out of 18459), where it extends beyond the connectives sentence to include additional sentences in the subsequent discourse (Prasad et al., 2008).

GF4. *Arg1 after the connective sentence*. Count of Arg1, if any, occurs classified after connective sentence. Through this feature we attempt to constrain the system to have Arg1s always occurring in the previous sentence or before the previous sentence of the connective sentence.

GF5. *Argument overlapping with the connective*. Count of the cases if there is any token overlap between Args and connective tokens. This is also not possible for the PDTB-style annotation, so we intend to constrain the overlapping, if any.

GF6. *Argument begins with -I tag*. Count of the cases if the generated Arg chunks begins with the -I (inside) tag, violating the principle of IOB tags for chunking. This is only possible if the CRF chunker fails to tag the boundaries properly.

GF7. *Argument begins with -E tag*. Count of the cases if the generated Arg chunks begins with the -E (end) tag instead of a -B(begin) tag. This is also possible if only the CRF chunker fails to tag the chunk boundaries properly.

We attempt to categorize this feature set according to the properties they bear: $\{GF0\}$ is the *intrinsic* global feature - it is the evidence of confidence on decisions made by the single-best model; $\{GF1, GF2\}$ check the *prevalent problems* seen through the evaluation of decisions by the single best model; $\{GF3, GF4, GF5\}$ are the *hypothetical* global features those reduce classification complexities, they are inspired by the general trends or rules for annotation in PDTB. $\{G6, G7\}$ check the *mistakes* in IOB tagging by the CRF chunker.

4 Reranking Approaches

We formalize the reranking algorithm as follows: for a given sentence s, a reranker selects the best parse $\hat{y}$ among the set of candidates candidate(s) according to some scoring function:

$$\hat{y} = argmax_{y \in \text{candidate}(s)} \text{score}(y) \qquad (1)$$

In n-best reranking, candidate(s) is simply a set of n-best parses from the baseline parser, that is, candidate$(s) = \{y_1, y_2, ..., y_n\}$.

In this paper we followed two approaches for the reranking task:

1. Structured Learning Approach: in this case the reranker learns directly from a scoring function that is trained to maximize the performance of the reranking task (Collins and Duffy, 2002). We also investigate two popular and efficient online structured learning algorithms: the structured voted perceptron by Collins and Duffy (2002) and Passive-Aggressive(PA) algorithm by Crammer et al. (2006). The weight-vectors observed from the training phase are averaged following Schapire and Freund (1999). In case of structured perceptron for each of the candidate in a ranked list the scoring function of equation 1 is computed as follows:

$$\text{score}(y_i) = \mathbf{w} \cdot \mathbf{\Phi}(x_{i,j}) \qquad (2)$$

where $\mathbf{w}$ is the parameter weight-vector and Φ is the feature representing function of $x_{i,j}$; $x_{i,j}$ denotes the j-th token of the i-th sentence. Since the PA algorithm is based on the theory of large-margin, it attempts find a score that violates the margin maximally by adding an extra cost i.e. $\sqrt{\rho(x_{i,j})}$ to the basic score function for structured perceptron i.e. equation 2. Here ρ is computed as $1 - F(x_{i.j})$, F: F-measure. The online PA also takes care of the learning rate of perceptron, which is considered as 1 in structured perceptron. The learning rate in online PA is min-value between a regularization constant and normalized score function value.

2. Best vs. rest Approach: in the preference kernel approach (Shen and Joshi, 2003) the reranking problem is reduced to a binary classification task on pairs. This reduction enables even a standard support vector machine to optimize the problem. We use a component of this task. We define the best scored discourse window (section 4.1) as a positive example and the rest are the negatives to the system. We use a standard support vector machine (Vapnik, 1995) with linear kernel.

3. Preference Kernel Approach: we also investigated the classical approach of preference kernel, as it is introduced by (Shen and Joshi, 2003). In this method, the reranking problem learning to select the correct candidate h^1 from a candidate set $\{h^1, \cdots, h^k\}$ is reduced to a binary classification problem by creating pairs: positive training instances $\langle h^1, h^2\rangle, \cdots, \langle h^1, h^k\rangle$ and negative instances $\langle h^2, h^1\rangle, \cdots, \langle h^k, h^1\rangle$. The advantage of using this approach is that there are abundant tools for binary machine learning.

If we have a kernel K over the candidate space T, we can construct a *preference* kernel

(Shen and Joshi, 2003) P_K over the space of pairs $T \times T$ as follows:

$$\begin{aligned} P_K &= K(h_1^1, h_2^1) + K(h_1^2, h_2^2) \\ &- K(h_1^1, h_2^2) - K(h_1^2, h_2^1) \end{aligned} \quad (3)$$

In our case, we make pair from the n-best hypotheses h_i as $\langle h_i^1, h_i^2 \rangle$ generated by the base model. We used linear kernel to train the reranker.

Thus we create the feature vectors extracted from the candidate sequences using the features described in Section 3. We then trained linear SVMs (Support Vector Machine) using the LIBLINEAR software (Fan et al., 2008), using L1 loss and L2 regularization.

4.1 Experiments

We use PennDiscourse TreeBank (Prasad et al., 2008) and Penn TreeBank (Marcus et al., 1993) data through this entire work. We keep the split of data as follows: $02 - 22$ folders of PDTB (& PTB) are used for training, $23 - 24$ folders of the same are used for testing; remaining 00-01 folders are meant for development split, it is used only to study the impact of feature (*cf.* 5).

We prepare the n-best outputs of sentences from the base system (*cf.* 2.1). The training data is prepared from the input of n-best lists of the train split, using a oracle module, which generates k-best oracle lists from the n-best single outputs. We procure k-best lists from oracle using the evaluator module (see section 4.2), ordered by the highest to the lowest probability score. Each of the list of the k-best list is a 5-sentence discourse window.

We prepare the test data given the n-best lists of the test split. We obtain k-best list for testing, prepared with the module described in section 2.1. We re-integrate the sentences connected with the same discourse connective id into the 5-sentence discourse window keeping the connective-bearing sentence in the middle. This re-integration done using a priority queue in the style of Huang and Chiang (2005). Each of the list from the k-best list are ordered by the highest to the lowest score with sum of the log of posterior probabilities of each sentence in the n-best list.

Therefore, in short, the n-best list is the list of sentence-level analyses whereas the k-best list is the list of 5-sentence discourse window-level analyses.

Baseline: we consider the performance of the single-best output from the base implementation (*cf.* 2.1) as the baseline.

4.2 Evaluation

We present our results using precision, recall and F1 measures. Following Johansson and Moschitti (2010), we use three scoring schemes: *exact*, *intersection* (or *partial*), and *overlap* scoring. In the exact scoring scheme, a span extracted by the system is counted as correct if its extent exactly coincides with one in the gold standard. We also include two other scoring schemes to have a rough approximation of the argument spans. In the overlap scheme, an expression is counted as correctly detected if it overlaps with a gold standard argument. The intersection scheme assigns a score between 0 and 1 for every predicted span based on how much it overlaps with a gold standard span, so unlike the other two schemes it will reward close matches.

4.3 Classifier Results

	ARG1 Results			ARG2 Results		
Exact	P	R	F	P	R	F
Baseline	69.88	48.51	57.26	83.44	75.14	79.07
Online PA	**66.10**	**53.92**	**59.39**(16)	82.59	76.39	79.37(4)
Struct Per	67.18	52.64	59.03(4)	**82.96**	**76.28**	**79.48**(8)
BestVsRest	66.19	52.83	58.94(8)	81.69	77.14	79.35(4)
Pref-Linear	66.54	53.31	59.20(4)	82.82	76.28	79.42(4)

Table 3: Exact Match Results for four classifiers. Baseline scores in the first row. Used n-best list numbers in parenthesis. The best performances are boldfaced.

We observe that reranking with global features improved the F1 scores for `Arg1` significantly, although for `Arg2` the improvement is insignificant [5]. Since in most of the cases the `Arg2` is syntactically bound with the connective, it is obvious that lexico-syntactically motivated local features help the classification of `Arg2`. On the other hand, the classification of `Arg1` is considerably dependent on non-grammatical, hand-crafted rule generated features. If we compare to our reranking classification results of `Arg1` with that one without previous sentence feature in Ghosh et al. (2011a) then we observe that the global and globally motivated structural feature improved the classification

[5]Throughout this work the permutation test is used to compute the significance of difference, whereas to compute the confidence interval bootstrap resampling is used(Hjorth, 1993). We determined the significant digits for presenting results using the methods illustrated by Weisstein E. W. (Weisstein, 2012)

of Arg1 by more than 10 points.

We also notice from the table for both the argument classification cases that we achieve balanced scores in terms of the precision and the recall with the structured global features. In fact there is a good improvement of recall without much loss in terms of precision.There is not any significant improvement in case of Arg2 reranking because the problem of the classification mostly resides on boundary detection of Arg2; also we know that estimation of position of an Arg2 is pretty easy task given the connective is correctly identified.

	ARG1 Results			ARG2 Results		
Exact	P	R	F	P	R	F
Baseline	82.90	61.65	70.72	93.40	84.20	88.56
Online PA	**80.11**	**69.43**	**74.39**(16)	**92.94**	**85.73**	**89.19**(4)
Struct Per	81.18	67.03	73.43(4)	93.20	85.50	89.17(8)
BestVsRest	81.25	66.46	73.11(8)	93.03	85.16	89.1(4)
Pref-linear	80.55	68.49	74.03(4)	93.12	85.56	89.18(4)

Table 4: Partial Match Results for four classifiers. Baseline scores in the first row. Used n-best list numbers in parenthesis. The best performances are boldfaced.

We mark an improvement of the Arg1 in table 4, with softer partial evaluation metrics; we also observe the same trend in results for Arg2 classification as in the table 3.

4.3.1 Candidate Set Size

We conduct experiments to study the influence of candidate set size on the quality of reranked output. In addition we also attempt to notice the upper-bound of reranker performance, i.e. the oracle performance. We choose the reranker based on online PA among the four classifier. Since all the four classifiers performed comparably the same way, it is enough to study the performance of one of them on candidate set size, that will reflect the performance of the other classifiers. We also describe and discuss the results on the exact partial measures only, as we notice from the previous section that the effect of reranking is comparable with the exact measure and softer measures.

	Reranked ARG1			Oracle		
k	P	R	F	P	R	F
1	69.88	48.51	57.26	69.88	48.51	57.26
2	67.26	52.34	58.87	81.26	61.70	70.14
4	66.39	53.56	59.29	88.35	71.91	79.29
8	66.11	53.86	59.36	92.47	79.09	85.26
16	66.10	53.92	59.39	93.80	83.77	88.50

Table 5: Oracle and reranker performance as a function of candidate set size of Arg1.

In both the tables (5, 6) we notice that the oracle performance is steadily increasing with 16-best lists. We observe that the performance of classification of both Arg1 and Arg2 increases at the level of 2-best list then it stagnates after 4-best performance. This nature of increment is may be related to the simple but high-level feature set used in this task of the discourse parsing; and it can also be some issues involved with local feature set, as we observed a huge difference of posterior probabilities between the single-best and the each of the $(n-1)$ lists of a n-best decision by CRF.

	Reranked ARG2			Oracle		
k	P	R	F	P	R	F
1	83.44	75.14	79.07	83.44	75.14	79.07
2	82.90	75.69	79.13	90.13	82.43	86.11
4	82.59	76.39	79.37	92.27	86.53	89.31
8	82.41	76.44	79.32	92.81	88.13	90.41
16	83.41	76.44	79.32	92.82	88.54	90.63

Table 6: Oracle and reranker performance as a function of candidate set size of Arg2.

4.3.2 Reranked Intersentential ARG1

We also attempt to observe the effect with respect to inter-sentential classification in case of Arg1, with the results obtained with online PA perceptron. As expected, the change we notice the effects in the table 7 is a fraction of potential improvement. We find comparing the inter-sentential vs. overall classification results of Arg1 that the increment in inter-sentential Arg1 classification considerably contribute to the overall Arg1 classification.

		P	R	F1
Baseline	Exact	52.87	27.80	36.44
	Partial	68.93	41.06	51.48
	Overlap	79.62	41.88	54.88
Best Reranked ARG1	Exact	50.41	30.04	37.56
	Partial	66.51	44.95	53.78
	Overlap	76.13	44.54	56.23

Table 7: Inter-sentential Reranked Arg1 Results.

5 Impact of Feature on ARG1

We study the impact of global features on the performance on Arg1 reranker with the development set (*cf.* Section 4.1). We are leaving behind the feature performance of the Arg2, as the improvement by the reranker for this case is not significant.

The Table 8 shows the results of investigation through an incremental greedy-search based feature selection. All the performance steps are evaluated with a k of 16.

This impact table starts with the *log posterior* only (GF0). This results to the best result achieved by Ghosh et al. (2011a) through the hill-climbing feature analysis. Beside this, we also checked that if we run the reranker with this feature only, then it results to the baseline performance with the test split.

Then the *undergeneration* feature (GF2) is chosen through greedy search among the other features. It gives us, jointly with the log posterior, a significant improvement over the baseline. The impact is predictable as GF2 addresses the basic problem that has driven us to the current task.

The addition of the *overgeneration* (GF1) feature also increased the performance, though non-significantly; this feature is important for the reranker because this is meant for fixing a predominant overgeneration problem in the n-best lists.

We observe that the F1 measure increases significantly after adding the next important feature: *Arg1 after the connective sentence* (GF4); in this case the recall increases more in comparison to the increment in the precision.

In the next step, the feature: *Argument overlapping with connective* (GF5) is added. This decreases the F1 score a bit, though it increases the precision lowering the recall.

We reach to the second-best performance of the `Arg1` reranker after adding the feature: *Argument begins with -I tag* (GF6).

The addition of the feature: *Argument begins with -E tag* (GF7) does not improve the performance much. It is possible that there was no such mistake by CRF inside the test data.

The scores with partial and overlap matches show the same trend so we leave the discussion with them in order to avoid the redundancy.

Additionally, we also perform the individual effect of each of features from the set (GF1,GF2,GF4,GF5,GF6,GF7), jointly with the intrinsic feature GF0, but none other than the undergeneration feature increased the performance over the baseline.

The *intrinsic* GF0 is contributing to achieve the baseline performance; the undergeneration (GF2) feature is also contributing significantly. In summary, the combination of features optimizes the performance of system in terms of F1-measure by decreasing the value of precision and raising the value of recall.

System	P	R	$F1$
GF0 (Posterior Only)	73.12	50.36	59.64
GF0+GF2	69.62	55.34	61.67
GF0+GF2+GF1	69.92	55.21	61.70
GF0+GF2+GF1+GF4	70.12	56.05	62.30
GF0+GF2+GF1+GF4+GF5	72.36	53.72	61.66
GF0+GF2+GF1+GF4+GF5+GF6	71.10	55.28	62.20
GF0+GF2+GF1+GF4+GF5+GF6+GF7	71.84	54.82	62.19

Table 8: Exact Match Results for `Arg1` through Incremental Feature Selection.

6 Conclusion

We note a significant improvement over the best performing model of discourse parser on the PDTB corpus. This is mostly contributed by the better performance in `Arg1` classification.

We also find that global features have greater impact on `Arg1` classification than that of `Arg2`. We investigate that that the performance of `Arg1` improved by more than 10 points in terms of F1 measure using the global (see Section 3) and structure related features (see Ghosh et al. (2011a)). This happens perhaps due to the fact `Arg2` is syntactically bound to the connective, whereas `Arg1` is not. `Arg2` depends more on local features (*cf.* Section 2.1) than global one. Basically this nature of dependency of `Arg1` on both local and global features are inherited through the PDTB annotation corpus, as well the local feature dependency of `Arg2` are completely data-driven.

The motivation of the paper is to make a balanced classification for both the `Arg1` and `Arg2`, achieved by implementing the constrained-system with global features. This enables to increase a huge recall without losing much in terms of precision.

It is also observed that while the performances of oracle of `Arg1` and `Arg2` are increasing steadily, the performances of both the rerankers stagnate at or before the point of 16-best lists; this is perhaps due to our effective, simple and small feature set.

In this task we emphasized on and studied the data-driven, global and non-grammatical feature set. This syntactic parse tree independent feature set may also be effective with the dialogue data annotated with PDTB annotation style.

7 Acknowledgement

This work was partially funded by IBM Collaborative Faculty Award 2011 grant.

References

[Barzilay et al.2004] Regina Barzilay, Lillian Lee, et al. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proc. of NAACL-HLT, 2004*.

[Barzilay et al.2005] Regina Barzilay, Mirella Lapata, et al. 2005. Modeling local coherence: an entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL05)*.

[Charniak and Johnson2005] E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL*.

[Collins and Duffy2002] Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL02*.

[Collins2000] Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Computational Linguistics*, pages 175–182. Morgan Kaufmann.

[Collins2002] Michael Collins. 2002. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of ACL 2002*.

[Crammer et al.2006] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Schwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

[Daume III et al.2004] Hal Daume III, Daniel Marcu, et al. 2004. Np bracketing by maximum entropy tagging and svm reranking. In *Proceedings of EMNLP'04*.

[Fan et al.2008] Rong-En Fan, Chih-Jen Lin, Kai-Wei Chang, Xiang-Rui Wang, et al. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*.

[Ghosh et al.2011a] Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. 2011a. Shallow discourse parsing with conditional random fields. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, Chiang Mai, Thailand.

[Ghosh et al.2011b] Sucheta Ghosh, Sara Tonelli, Giuseppe Riccardi, and Richard Johansson. 2011b. End-to-end discourse parser evaluation. In *Proceedings of 5th IEEE International Conference on Semantic Computing*, Palo Alto, CA, USA.

[Ghosh et al.2012] Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. 2012. Improving the recall of a discourse parser by constraint-based postprocessing. In *Proceedings of International Conference on Languages Resources and Evaluations (LREC 2012)*.

[Grosz et al.1992] B.J. Grosz, J. Hircshberg, et al. 1992. Some intonational characteristics of discourse structure. In Ohala et al., editors, *Proceedings of the International Conference on Spoken Language Processing, Vol. 1*, volume 1, pages 429–432.

[Grosz et al.1995] B.J. Grosz, A. K. Joshi, S. Weinstein, et al. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2).

[Hjorth1993] J. S. Urban Hjorth. 1993. *Computer Intensive Statistical Methods*. Chapman and Hall, London.

[Huang and Chiang2005] Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT 2005)*, pages 53–64, Vancouver, Canada.

[Johansson and Moschitti2010] Richard Johansson and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 67–76.

[Klein and Manning2003] Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. *Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge, MA: MIT Press*.

[Lafferty et al.2001] John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th International Conf. on Machine Learning*. Morgan Kaufmann.

[Lapata2003] Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Meeting of the Association of Computational Linguistics*, pages 545–552.

[Marcus et al.1993] 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.

[Minnen et al.2001] Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*.

[Pitler and Nenkova2009] Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the 47^{th} Annual Meeting of the Association for Computational Linguistics and the 4^{th} International Joint Conference on Natural Language Processing*.

[Prasad et al.2008] 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 the 6^{th}*

International Conference on Languages Resources and Evaluations (LREC 2008), Marrakech, Morocco.

[Schapire and Freund1999] Robert E. Schapire and Yoav Freund. 1999. Large margin classification using the perceptron algorithm. *Machine Learning Journal*, 37(3):277–296.

[Shen and Joshi2003] Libin Shen and Aravind Joshi. 2003. An svm based voting algorithm with application to parse reranking. In *CoNLL 2003*.

[Soricut et al.2003] Radu Soricut, Daniel Marcu, et al. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL), May 27-June 1*.

[Soricut et al.2006] Radu Soricut, Daniel Marcu, et al. 2006. Stochastic coherence modeling, parameter estimation and decoding for text planning applications. In *Proceedings of ACL-2006 (Poster)*, pages 803–810.

[Toutanova et al.2008] Kristina Toutanova, Aria Haghighi, Christopher D. Manning, et al. 2008. Kristina toutanova, aria haghighi, and christopher d. manning, a global joint model for semantic role labeling. *Computational Linguistics*.

[Vapnik1995] V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag.

[Weisstein2012] Eric W. Weisstein. 2012. "significant digits." from mathworld–a wolfram web resource.

[Yamada and Matsumoto2003] Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of 8th International Workshop on Parsing Technologies*.