Computation of Shallow Discourse Structure

Shallow Discourse Parsing

ESSLLI 2019, Riga

Tatjana Scheffler

tatjana.scheffler@uni-potsdam.de

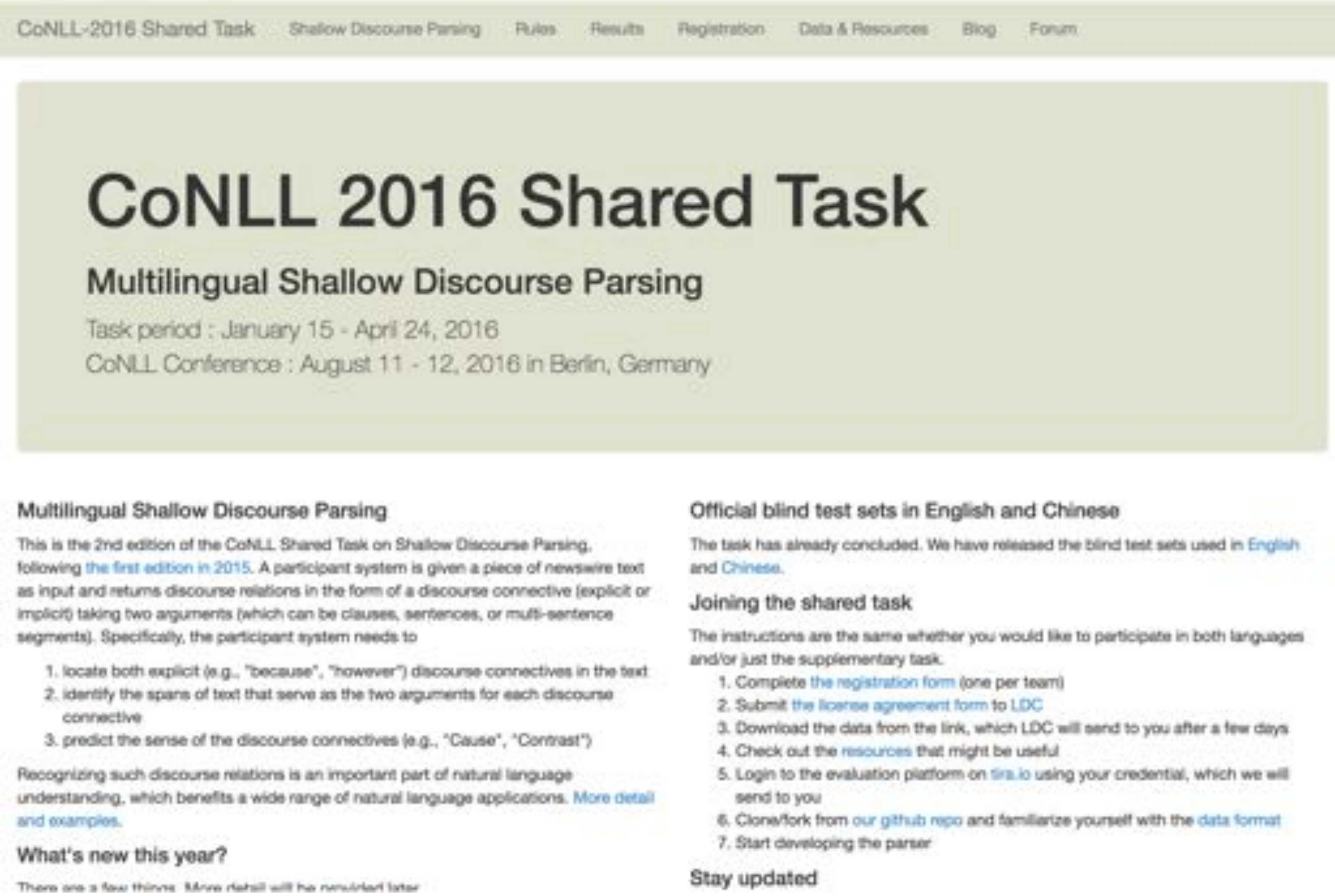
Day 3, August 14, 2019

Shallow Discourse Parsing

CONLL Shared Task

CONLL Shared Task (2015/16)

- https://www.cs.brandeis.edu/~clp/conll16st/



CoNLL 2016 Shared Task

Multilingual Shallow Discourse Parsing

Task period : January 15 - April 24, 2016
CoNLL Conference : August 11 - 12, 2016 in Berlin, Germany

Multilingual Shallow Discourse Parsing

This is the 2nd edition of the CoNLL Shared Task on Shallow Discourse Parsing, following the first edition in 2015. A participant system is given a piece of newswire text as input and returns discourse relations in the form of a discourse connective (explicit or implicit) taking two arguments (which can be clauses, sentences, or multi-sentence segments). Specifically, the participant system needs to

1. locate both explicit (e.g., "because", "however") discourse connectives in the text
2. identify the spans of text that serve as the two arguments for each discourse connective
3. predict the sense of the discourse connectives (e.g., "Cause", "Contrast")

Recognizing such discourse relations is an important part of natural language understanding, which benefits a wide range of natural language applications. More detail and examples.

What's new this year?

There are a few things. More detail will be provided later

Official blind test sets in English and Chinese

The task has already concluded. We have released the blind test sets used in English and Chinese.

Joining the shared task

The instructions are the same whether you would like to participate in both languages and/or just the supplementary task.

1. Complete the registration form (one per team)
2. Submit the license agreement form to LDC
3. Download the data from the link, which LDC will send to you after a few days
4. Check out the resources that might be useful
5. Login to the evaluation platform on tira.io using your credential, which we will send to you
6. Clone/fork from our github repo and familiarize yourself with the data format
7. Start developing the parser

Stay updated

CONLL Shared Task (2015/16)

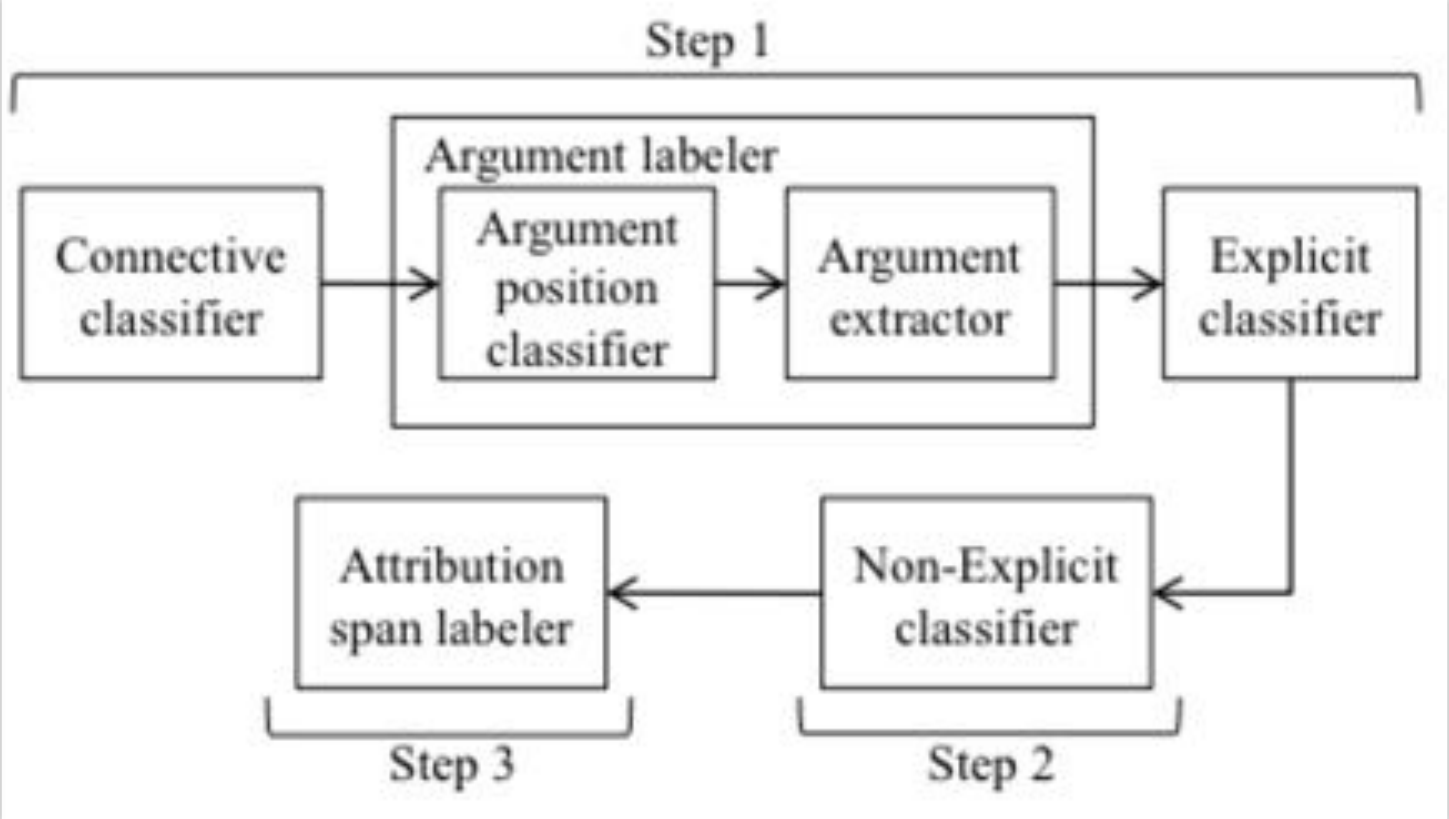
- https://www.cs.brandeis.edu/~clp/conll16st/
- common data sets (training, dev) and test platform
- well-defined problem
- standard input and output formats
- https://nbviewer.jupyter.org/github/attapol/conll16st/blob/master/tutorial/tutorial.ipynb

Task

- Identify the text span of an explicit discourse connective, if present, or the position between adjacent sentences as the proxy site of an implicit discourse relation
- Identify the two text spans that serve as arguments to the relation
- Label the argument spans as Arg1 or Arg2, as appropriate
- Predict the sense of the discourse relation (e.g., “Cause”, “Condition”, “Contrast”)

PDTB-style Discourse Parser

Lin et al. (2012)



https://www.comp.nus.edu.sg/~kanmy/papers/nleLin2012.pdf

Evaluation

- Gold standard (GS) parses without error propagation

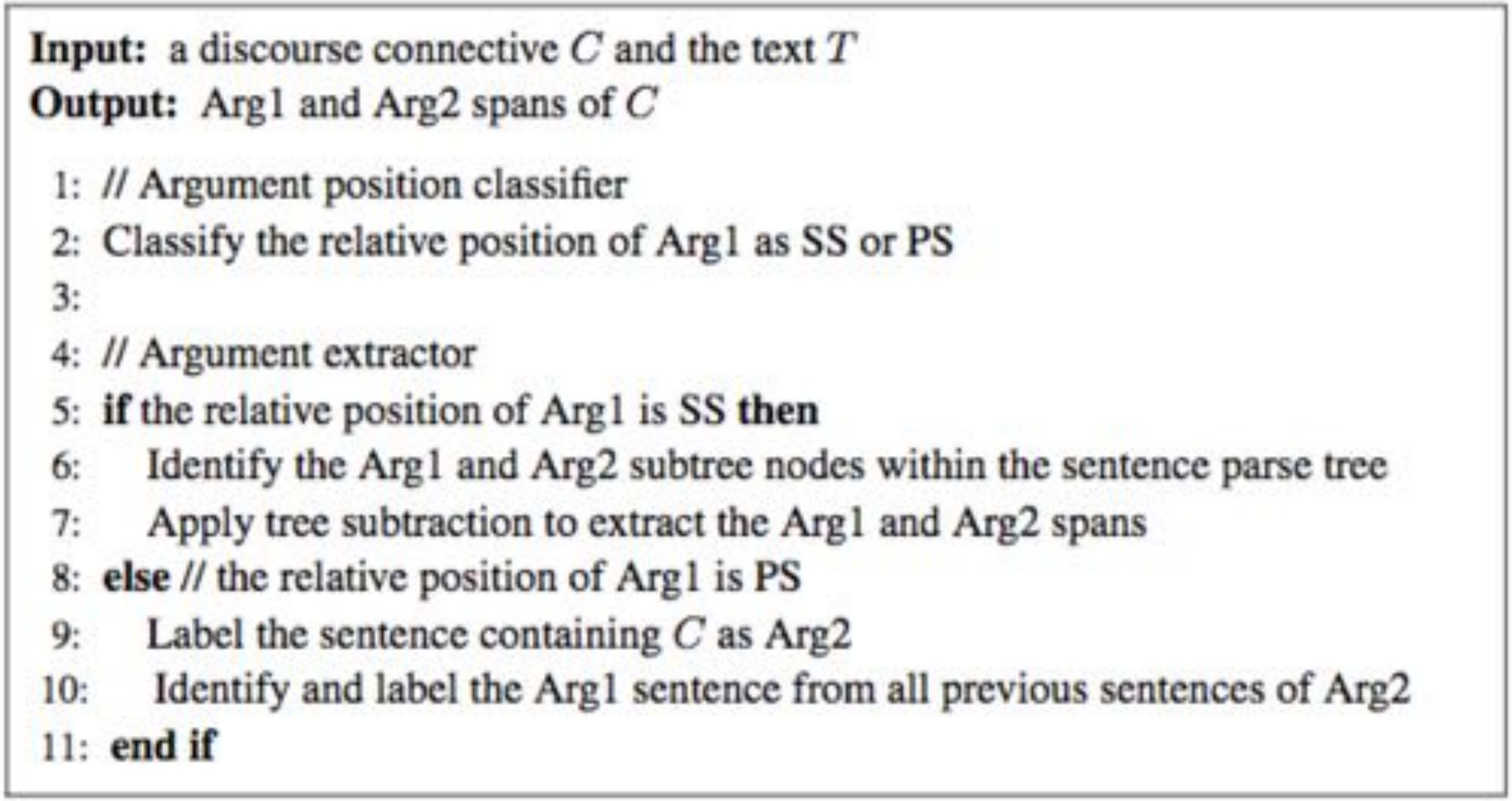
1. connective classifier

- assume a list of explicit connectives
- disambiguation problem
- MaxEnt classifier (logistic regression)
- syntactic and lexical features: connective, its POS-tag and immediate context, syntactic sisters and path to root
- F_1 = 95.76%

2. argument position

- relative position of Arg1 and Arg2
- same sentence, previous sentence
- features:
  - position of connective
  - contextual features
- Component F_1 = 97.94%

3. argument extractor



Input: a discourse connective C and the text T

Output: Arg1 and Arg2 spans of C

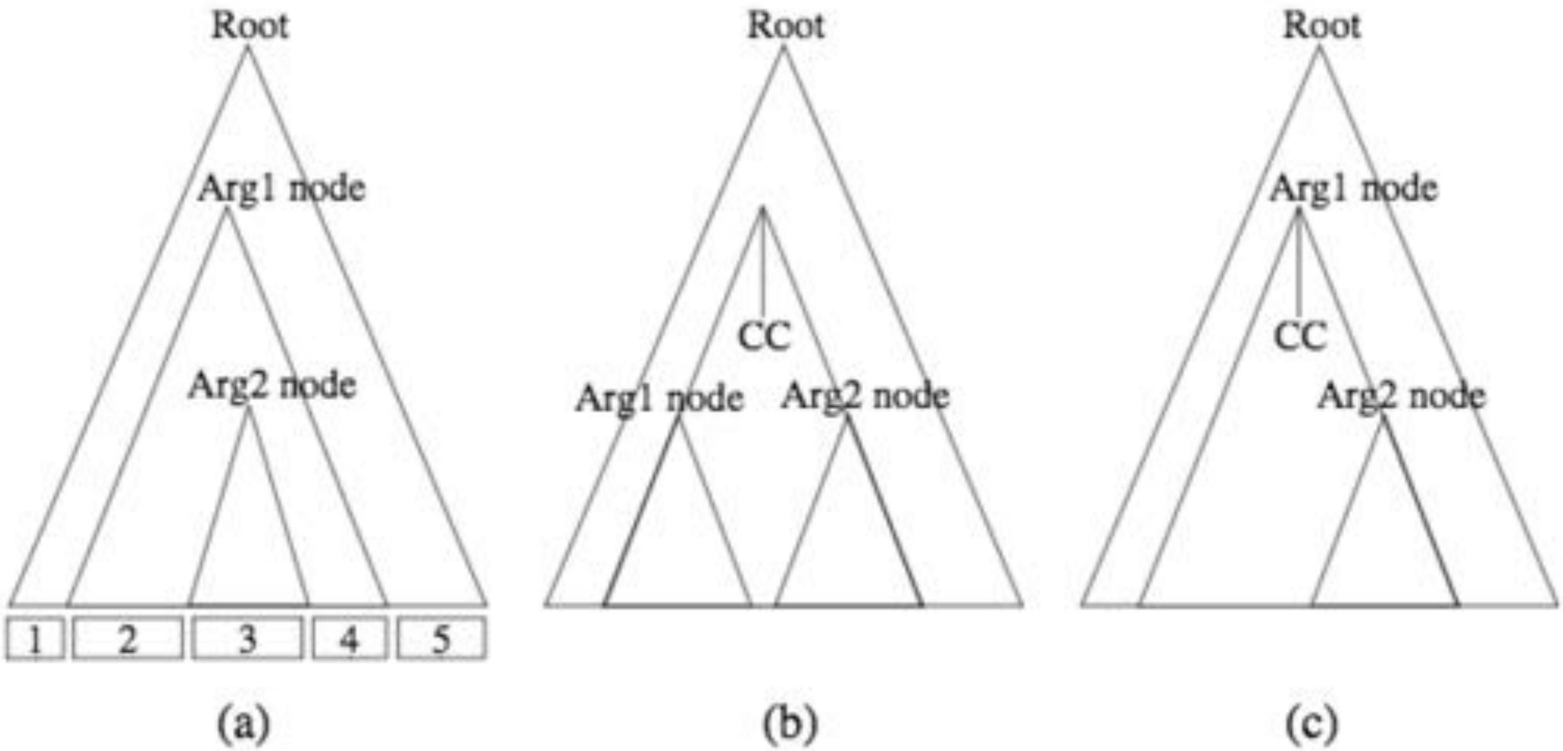
```
// Argument position classifier
Classify the relative position of Arg1 as SS or PS

// Argument extractor
if the relative position of Arg1 is SS then
    Identify the Arg1 and Arg2 subtree nodes within the sentence parse tree
    Apply tree subtraction to extract the Arg1 and Arg2 spans
else // the relative position of Arg1 is PS
    Label the sentence containing $C$ as Arg2
    Identify and label the Arg1 sentence from all previous sentences of Arg2
end if
```

3. argument extractor (results)

- split sentence into clauses
- assign probabilities to each node
- subtract subtrees from potential arguments
- component F_1 = 86.24% for partial matches
- component F_1 = 53.85% for exact matches

relation of Arg1 and Arg2 (SS)



example: nesting arguments

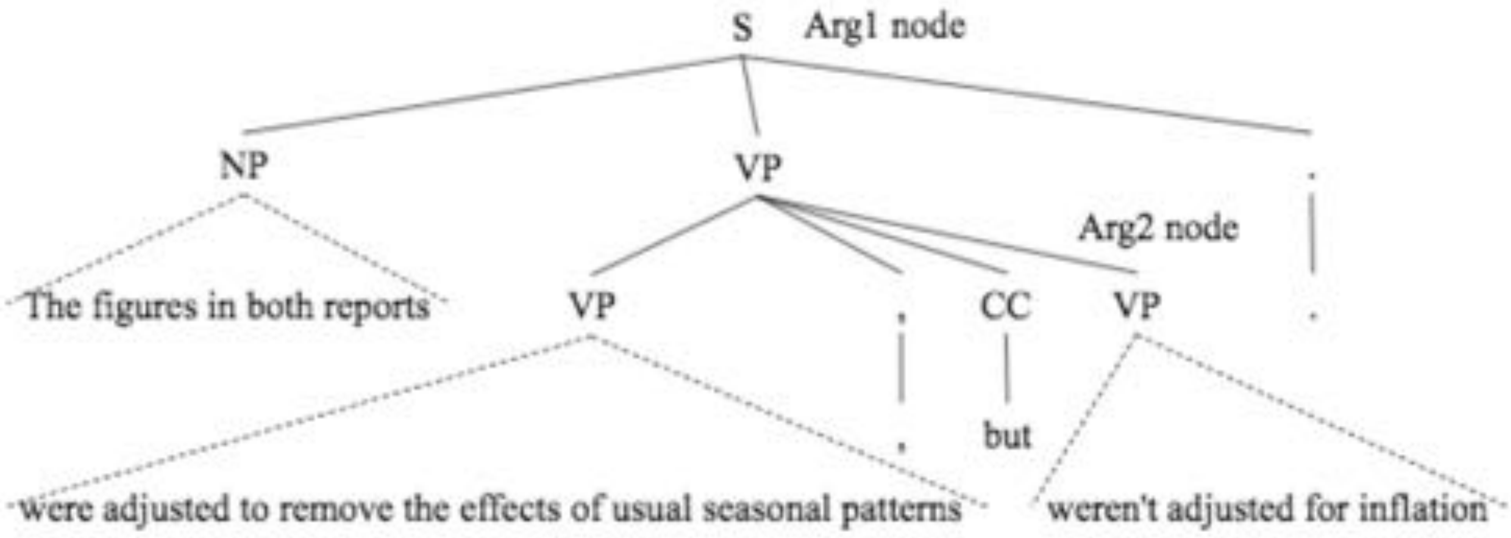

S
Arg1 node
NP
VP
.
Arg2 node
The figures in both reports
VP
,
CC
VP
.
,
but
were adjusted to remove the effects of usual seasonal patterns
weren't adjusted for inflation

Fig. 7. The parse tree for Example 14 to illustrate Figure 6(c).

4. explicit (sense) classifier

- assign sense tag
- features: (only!)
  - connective C
  - its POS tag
  - C + previous word
- component F_1 = 86.77%

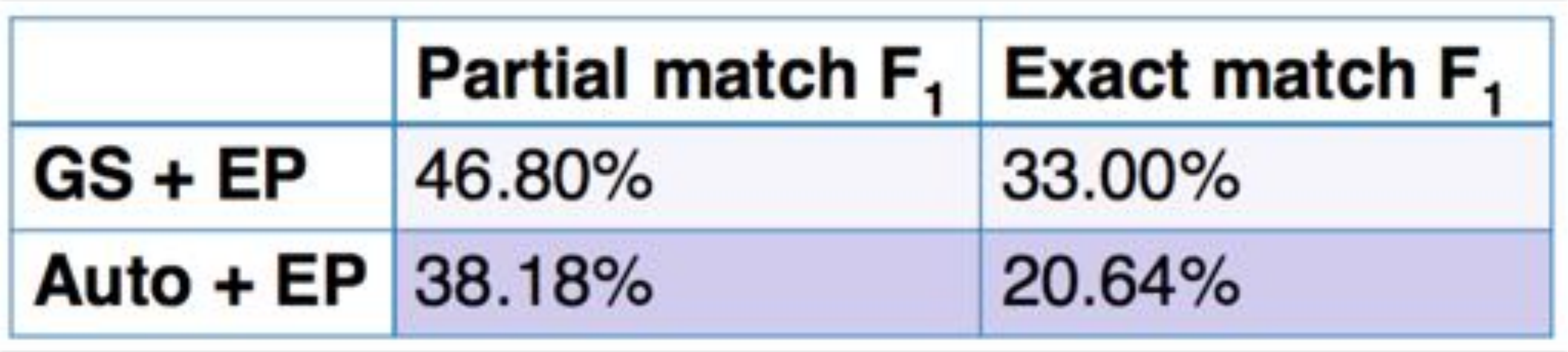
5. non-explicit classifier

- classify all remaining adjacent sentence pairs as Implicit/AltLex, EntRel, NoRel
- one classifier for 11 sense types + EntRel, NoRel
- separately determine AltLex using first three words of Arg2
- features:
  - contextual
  - dependency/constituent parses
  - word pairs
- component F_1 = 39.63%

6. attribution span labeler

- split text into clauses and determine which clauses are attribution spans
- mostly lexical features
- component F_1 = 79.68% for partial matches
- component F_1 = 65.95% for exact matches

results shallow discourse parsing



	Partial match F_1	Exact match F_1
GS + EP	46.80%	33.00%
Auto + EP	38.18%	20.64%