Are All Commas Equal? Detecting Coordination in the Penn Treebank

Wolfgang Maier **Sandra Kübler**

University of Düsseldorf
Düsseldorf, Germany
maierw@hhu.de

Indiana University
Bloomington, Indiana
skuebler@indiana.edu

Abstract

Coordination has always been a difficult phenomenon, with regard to linguistic analysis, manual annotation, and automatic analysis. There is a considerable body of work on detecting coordination and on improving parsing for this phenomenon. However, most approaches were restricted to certain types of coordination, such as NP coordination or symmetrical coordination. We present the first approach to classifying punctuation signs into whether they function as separators between conjuncts in coordination or not.We show that by using information from a parser in combination with context information, we reach an F-score of 89.22 on positive cases.

1 Introduction

The syntactic analysis and annotation of coordinated structures has generally been recognized as a difficult problem, in linguistics as well as in computational linguistics. Most linguistic frameworks still struggle with finding an account for coordination that is descriptively fully adequate [7]. In parsing approaches, coordinated structures constitute one of the main sources for errors [9]. Therefore, there are approaches in parsing that focus on improving parser performance specifically for this phenomenon [9, 11, 8, for example]. Others focus on detecting the scope of coordinations [6, 17]. However, most of these approaches focus on a closely defined subset of coordination types: either noun phrase coordination [9] or symmetric coordination of two conjuncts [13].

For English, one of the reasons for this focus lies in the annotation of the Penn Treebank (PTB) [15], which does not mark coordinated structures as such (for more details see section 3). For this reason, it is difficult to find certain coordinations, namely phrasal coordination without an overt conjunction, coordination on the clausal level, and coordinations with more than two conjuncts. However, there is an additional resource, which annotates all punctuation signs in the Penn

Treebank as to whether they function as conjunctions or not [14]. The current version of the annotation covers all sentences of the Penn Treebank release 3. Four annotators were involved.[1]

In this paper, we use this annotation in combination with the Penn Treebank to develop an automatic approach to detecting coordination and identifying its internal conjunct boundaries. More specifically, we interpret this task as a binary classification problem, in which a classifier decides whether a punctuation sign has a coordinating function, given its context, or not. If we can detect all punctuation signs, and combine them with the syntactic annotation, it is possible to determine the scope of the coordination, but also the number of conjuncts. In the current work, we focus on determining the types of information that are useful for the classification task: basic information such as part-of-speech (POS) tags, context words, or information about context punctuation; gold syntactic information, or syntactic information from a state-of-the-art parser.

The remainder of the paper is structured as follows: Section 2 describes work on parsing coordinations, section 3 gives our definition of coordination and describes the annotations in [14]. In section 4, we describe our experimental setup, in section 5 the results, and in section 6 our conclusions and future work.

2 Related Work

There are only a few approaches to explicitly identifying conjuncts. Ogren [17] concentrates on finding the full scope of coordinations with an overt conjunction. His method constructs simplified sentences out of coordinated sentences, each containing only one conjunct. He extracts possible left and right conjuncts and then evaluates their quality by a 4-gram language model. While Ogren is aware that there are coordinations with more than two conjuncts, he seems to group all conjuncts left of the conjunction into one maximal conjunct. Ogren's task is at the same time easier and more difficult than ours: He identifies only one left and one right conjunct of overt conjunctions, but then, his evaluation is rather strict in that the span of the conjuncts has to match the gold standard exactly.

Hara and Shimbo [6] work on the identification of the scopes of all coordinations with overt conjunctions in a sentence. Their input is a whole sentence, which they parse using a simple, manually written, overgenerating grammar for coordinations, in combination with a perceptron model to determine the optimal scope of potential conjuncts.

All other related works either aim at disambiguating specific subsets of coordinations or at improving parser models to better handle coordinated cases: Chantree et al. [2] approach cases of noun coordination involving one modifier, such as `old boots and shoes`. They use *SketchEngine* [12] to retrieve collocation statistics and distributional similarities. Based on this information, the candidates are ranked by heuristics. Dale and Mazur [4] tackle the problem of conjunction ambiguity in

[1]The resource will be made available soon via the Linguistic Data Consortium.

named entity recognition with a supervised approach. Bergsma and Yarowsky [1] use specialized classifiers based on bilingual, aligned data as well as on Google n-grams for the same task.

Integrating methods for improving coordination parsing has been performed on a wide range of languages: Hogan [9] integrates a specialized probability model for symmetrical coordinations into a parser for English. Guo et al. [5] use LFG-approximations to deal with long-distance dependencies in Chinese, including coordinations. Kübler et al. [13] extract possible scopes for coordinations with two conjuncts and an overt conjunction and rerank them. They work on German. Kawahara and Kurohashi [11] introduce a synchronized generation process into a generative dependency parser for Japanese. And Henestroza and Candito [8] use parse revisions to improve e.g. coordination in French.

3 Coordination Annotation

3.1 Coordination

Coordinations are complex syntactic structures that consist of two or more conjuncts. One or more of the conjuncts is often preceded by an (overt coordinating) conjunction, such as `and`, `or`, `neither...nor`, and `but`, see (1) for examples from the Penn Treebank. However, there are also cases of coordinations that lack coordinating conjunctions altogether, see (2).

(1) a. The total of 18 deaths from [malignant mesothelioma, lung cancer and asbestosis] was far higher than expected, the researchers said.
b. [Mr. Katzenstein certainly would have learned something, and it's even possible Mr. Morita would have too].
c. [The [Perch and Dolphin] fields are expected to start producing early next year, and the Seahorse and Tarwhine fields later next year].

(2) a. ... a [humble, "uncomplaining, obedient] soul," ...
b. [The 30-day simple yield fell to an average 8.19% from 8.22%; the 30-day compound yield slid to an average 8.53% from 8.56%].

Coordinate structures typically exhibit syntactic parallelism in the sense that each conjunct belongs to the same lexical or phrasal category. However, coordinations of unlike categories are also possible, as in the example in (3), which conjoins the prepositional phrase `after the changes` and a clause `assuming no dramatic fluctuation in interest rates`. Typically, the conjuncts are syntactic constituents, but there are cases of non-constituent conjunctions, such as, e.g., in example (1-c), which involves an elliptical construction.

(3) He also said that [after the charges, and "assuming no dramatic fluctuation in interest rates], the company expects to achieve near-record earnings in 1990."

```
( (S
    (NP-SBJ-1
      (NP
        (NP
          (NP (DT The) (JJ male) (NN part) )
          (PP (-NONE- *RNR*-2) ))
        (, ,)
        (NP
          (NP (DT the) (NNS anthers) )
          (PP (IN of)
            (NP (DT the) (NN plant) ))))
      (, ,)
      (CC and)
      (NP
        (NP
          (NP (DT the) (NN female) )
          (PP (-NONE- *RNR*-2) ))
        (, ,)
        (NP (DT the) (NNS pistils) )
        (, ,) )
      (PP-2 (IN of)
        (NP (DT the) (JJ same) (NN plant) )))
    (VP (VBP are)
      (UCP-PRD
        (PP-LOC-PRD (IN within)
          (NP
            (NP (DT a) (NN fraction) )
            (PP (IN of)
              (NP (DT an) (NN inch) ))))
        (CC or)
        (ADVP (RB even) )
        (VP (VBN attached)
          (NP (-NONE- *-1) )
          (PP-CLR (TO to)
            (NP (DT each) (JJ other) )))))
    (. .) ))
```

Figure 1: A PTB tree with a symmetrical and an asymmetrical coordination.

3.2 Coordination in the Penn Treebank

In the Penn Treebank [15], symmetrical coordination is generally annotated by grouping the conjuncts plus the conjunction under a node of the same type. The tree in figure 1 shows an example of a coordinated noun phrase (NP) consisting of two complex NPs, `The male part` ... and `the female` .... Especially in cases without a conjunct, such constructions are difficult to distinguish from appositions, such as in (4-a), in which the two NPs `Elsevier N.V.` and `the Dutch publishing group` are also grouped under an NP. In some cases, even the presence of an overt conjunction is misleading when the conjunction introduces a different view of the same entity, as in (4-b). In this example, `EWDB` is another way of expressing the name of the company, and thus an apposition.

(4) a. Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.
    b. The transaction places the three executives squarely at the helm of a major agency with the rather unwieldy name of Eurocom WCRS Della Femina Ball Ltd., or EWDB.

Coordinations of unlike constituents are marked specifically, by a mother node `UCP`, as shown in figure 1 in the verb phrase (VP) where the PP `within a fraction of an inch` is coordinated with the VP `even attached to each other`. However, this is only true on the phrasal level; coordinations of unlike clauses are grouped under an S node.

3.3 Coordination Annotation: Making Coordination Explicit

In the version of the Penn Treebank annotated for coordination, almost all of these phenomena are annotated in the POS tag of the punctuation sign. The annotation focuses on the following punctuation signs: {, ; - – ...}. The only type of coordination not marked is coordination of clauses when there is no overt conjunct. Thus, examples (1-b) and (1-c) are marked for coordination, but (2-b) is not.

Additionally, the coordination annotation follows the syntactic annotation of

no. conjuncts	26	11	10	9	8	7	6	5	4
with CC	1	1	3	2	3	26	45	119	407
without CC	0	0	0	3	1	11	38	41	45

Table 1: An overview of the number of coordinations with many (≥ 4) conjuncts, separated into coordinations with and without overt conjunctions (CC).

the Penn Treebank as closely as possible. However, there are annotation errors in the treebank, such as the attachment of the comma after `plant` in figure 1, which should have been attached to the apposition `anthers of the plant`, not to the coordinated NP. In such cases, the coordination annotation deviates from the treebank annotation, and the comma after `plant` is annotated as non-coordinating.

Table 1 gives an overview of the number of coordinations with 4 or more conjuncts, with and without overt conjunctions over the whole treebank. These numbers show that we have a considerable number of coordinations with a high number of conjuncts, and we also have a non-negligible number of coordinations without an overt conjunction, i.e., cases where the conjuncts are separated by commas, semicolons, dots, or dashes. These frequencies show that it is not advisable to ignore either type of coordination, as has been the standard in previous work on parsing coordinations.

4 Experimental Setup

As mentioned before, we treat the problem of identifying conjunct boundaries which are marked by one of the punctuation signs in { , ; - – ... } as binary classification task, in which a punctuation sign which marks / does not mark a conjunct boundary is classified as positive or negative instance, respectively, given the context in which it appears. For the training of the classifier, we use sections 02 to 21 of the WSJ part of the Penn Treebank. For testing, we use section 22. (We reserve section 23 for parsing results when integrating our annotated punctuation.) The test set contains 1 937 instances of non-coordination and 351 instances of coordinating punctuation. As usual for parsing, in all trees, we remove all traces and empty nodes and the corresponding co-indexation markers on non-terminals.

4.1 Classification

For our experiments we use *k*-nearest-neighbor classification as provided by TiMBL version 6.3 [3].[2] We describe the context of a single punctuation terminal by both word-level and syntactic features. Since the different punctuation signs have different distributions of usage in coordinations, we include the punctuation sign itself

[2]We have also performed experiments with Support Vector Machines (SVM^{light}) [10], and a Maximum-Entropy classifier (MALLET version 2.07) [16]. However, since their performance was extremely low, even after sampling to reduce the skewing, we concentrate on TiMBL in this paper.

as a feature. The words and POS tags around a punctuation sign can be strong indicators for its function. Consider, for example, the words *and* and *who*. While a comma preceding the former has most likely a coordinating function, a comma following the latter most likely does not. We include the window of n tokens and POS tags directly adjacent to the punctuation terminal as features. We also take the normalized position of the punctuation sign, i.e., its position index normalized by sentence length, as indicator for its function, as well as the respective normalized positions of the next coordinating/non-coordinating punctuation signs on its left and right. Additionally, we use the relative distance of the next coordinating conjunction (CC) as information since most coordinations have an overt conjunction.

To describe the syntactic context of a punctuation sign, we first consider the label of its parent node since this should be the node to which the conjuncts to the left and to the right of the punctuation are attached. We add the siblings of the parent, the grandparent, and information whether the parent dominates a coordinating conjunction. We also include *leaf-ancestor (LA) paths* [19] of the n words and POS tags to the left and right of the punctuation. An LA path is the concatenation of labels encountered on the path from a leaf node to the root node (including both). We modify LA paths so that they stop at the parent of the punctuation sign since the global context should not provide much relevant information.

4.2 Parsing

The syntactic features can be obtained directly from the gold treebank trees. However, in order to provide a more realistic setting, we also investigate the effect of obtaining them from the trees output by a parser. In order to build a parsed version of the training set of the classifier, we use the Berkeley parser [18]. In order to avoid parsing on data seen in training, we use jackknifing on a 5-fold setting. We use the default settings for the Berkeley parser. The results for the concatenated training set and the test set are 84.92/85.03/84.98 and 85.29/85.10/85.19 in terms of precision, recall and F-score.

4.3 Evaluation

Since we treat our problems as a binary classification problem, the obvious evaluation metric would be accuracy. However, the sets are heavily skewed towards negative examples, and many of those are clearly non-coordinating, such as commas before `who`. For this reason, accuracy would place an overly high weight on those negative cases. Thus, we evaluate the classifier performance with regard to both classes, i.e., coordination or non-coordination. We compute precision, recall (the true positive rate), the F-score, the false positive rate. We report significance based on McNemar's test, on the 0.01 and the 0.1 level.

	positive				negative			
	prec.	recall	fp-rate	F-score	prec.	recall	fp-rate	F-score
baseline (pos)	87.31	80.34	2.12	83.68	96.49	97.88	19.66	97.18
pos + focus	88.99	80.63	1.81	84.60	96.55	98.19	19.37	97.36
pos + focus additionally:								
+1r	89.93	76.35	1.55	82.59*	95.83	98.45	23.65	97.12
+2r	90.88	76.64	1.39	83.15	95.88	98.61	23.36	97.23
+3r	90.16	78.35	1.55	83.84	96.17	98.45	21.65	97.30
+1l	88.16	80.63	1.96	84.23	96.54	98.04	19.37	97.28
+2l	88.25	79.20	1.91	83.48	96.30	98.09	20.80	97.19
+3l	89.03	78.63	1.75	83.51	96.21	98.25	21.37	97.22
+1l+1r	90.00	79.49	1.60	84.42	96.36	98.40	20.51	97.37
+2l+2r	90.82	78.92	1.45	84.45	96.27	98.55	21.08	97.40
+3l+3r	91.18	79.49	1.39	84.93	96.37	98.61	20.51	97.47
pos + focus + 3l + 3r additionally:								
+cc	90.25	81.77	1.60	85.80	96.75	98.40	18.23	97.57
+dist	91.83	80.06	1.29	85.54	96.47	98.71	19.94	97.58
+neigh	**92.77**	**84.05**	1.19	**88.19****	**97.16**	**98.81**	15.95	**97.98**
+dist+neigh	92.43	83.48	1.24	87.72*	97.06	98.76	16.52	97.90
+cc+dist+neigh	90.40	83.19	1.60	86.65	97.00	98.40	16.81	97.69

Table 2: Results for local information, with significance tested against baseline for pos + focus, against pos + focus for all other experiments; *=0.1; **=0.01.

5 Results

In our test set, we have 2 288 instances of punctuation. Out of those, 351 are annotated as coordinating. This means that by classifying all punctuation as non-coordinating, we reach an accuracy of $\frac{2\,288-351}{2\,288} = 84.66$. However, in our experiments, we are particularly interested in achieving a high F-score with respect to coordinating punctuation, i.e., the positive class.

To determine parameter settings, we conducted a non-exhaustive search and found the IB1 algorithm with overlap metric for the computation of instance distances, feature weighting via GainRatio, using the $k = 5$ nearest neighbors, and an inverse linear weighting of neighbors as function of their distances to be the best parameter combination (see [3] for an explanation of the parameters). All experiments reported below are based on these settings on the test set.

5.1 Using Local Context

Our baseline consists of three POS tags to the left and to the right of the punctuation sign. See the first row of table 2 for the results. Next, we add the punctuation sign itself (+ focus). This results in an increase of almost one percent point in terms of positive F-score, as shown in row two.

A separate evaluation of commas and semicolons for the experiment with baseline and focus word shows a positive F-score of 82.56 for commas and 96.23 for semicolons, compared to 84.60 for the complete evaluation.[3] Semicolons are classified much more reliably, most likely due to the fact that most of them are used on the clausal level, where only coordination cases with overt conjuncts are annotated. Thus, the most difficult cases, particularly those which involve a semantic component (such as apposition separator vs. NP coordination), involve commas. Also note that there are many more commas (2 143) than semicolons (76).

We now consecutively add lexical information in order to test our intuition of its discriminative power. The next part of table 2 shows the results. The highest F-score in this part is obtained by using 3 words to the right and to the left (+ 3l + 3r): a positive F-score of 84.93, which results from a high precision of 91.18. However, this increase is not significant over the pos + focus setting. Generally, adding context words increases precision, but has a larger detrimental effect on recall.

In the next section of the table, we add the position of the punctuation sign within the sentence (+ dist), the positions of neighboring (coordinating and non-coordinating) punctuation signs (+ neigh) to the left and right (i.e., 4 features), and the position of the next conjunction to the right (+ cc). The information about the position of the next `CC`-tagged word on the right (+ cc) and the position of the punctuation sign within the sentence (+ dist) surprisingly do not lead to a significant improvement of the F-score, as intuition would suggest. However, the four features of + neigh are particularly effective, leading to a significantly higher F-score. This setting reaches the highest overall results: positive precision: 92.77, recall: 84.05, and F-score: 88.19, negative precision: 97.16, recall: 98.81, and F-score: 97.98.

5.2 Using Gold Syntax

In this section, we investigate the benefit of adding syntactic features. We first add the parent node (+ p) of the punctuation sign. Then we add the grandparent (+ gp) and the two directly adjacent sibling non-terminals (+ sib) of the parent. In a next step, we add leaf-ancestor paths (+ la) as well as information whether the parent dominates a coordinating conjunction (+ ccn). As further features, we use the punctuation sign itself (+ focus), the position of its neighbors (+ neigh), and the three words around it (+ 3l3r). All syntactic features are extracted from the Penn Treebank.

Table 3 presents the results when using gold syntax.[4] The first row repeats the baseline of three POS tags left and right of the punctuation sign (here with gold POS tags). When we add the parent node (+ p), we gain approximately 2.7 percent points in positive precision and about 4.5 percent points in recall. This leads an F-score of 88.09, significantly higher than the baseline, and only 0.1 less than the F-score we gained by adding all the lexical context, thus showing how important the syntactic context of the punctuation sign is. Adding the grandparent

[3] Here, we ignore { – - . . . } because there are no positive instances of them in the test set.

[4] Note that the baseline results are better than for the local context since in the latter, we have used the POS tags created by the Berkeley parser.

	positive				negative			
	precision	recall	fp-rate	F-score	precision	recall	fp-rate	F-score
baseline (pos)	89.46	79.77	1.70	84.34	96.41	98.30	20.23	97.34
+p	92.21	84.33	1.29	88.09**	97.20	98.71	15.67	97.95
+p+gp	92.03	85.47	1.34	88.63	97.40	98.66	14.53	98.02
+p+gp+sib	91.87	86.89	1.39	89.31	97.65	98.61	13.11	98.13
+ p + gp additionally:								
+la	90.73	80.91	1.50	85.54*	96.61	98.50	19.09	97.55
+sib+la	90.69	86.04	1.60	88.30	97.49	98.40	13.96	97.95
+ccn	93.98	88.89	1.03	91.36**	98.01	98.97	11.11	**98.48**
+sib+ccn	93.43	89.17	1.14	91.25**	98.05	98.86	10.83	98.46
+ p + gp + sib + ccn additionally:								
+focus	93.16	**89.17**	1.19	91.12	98.05	98.81	10.83	98.43
+focus+3l3r	92.49	87.75	1.29	90.06*	97.80	98.71	12.25	98.25
+focus+neigh	93.71	**89.17**	1.08	91.39	**98.06**	98.92	10.83	98.48
all features	**94.26**	88.89	0.98	**91.50**	98.00	**99.02**	11.11	**98.51**

Table 3: Results for gold syntax. Significance: against baseline for + p, against + p for first/second experiments, against + ccn for third/fourth set; *=0.1; **=0.01.

adds an additional 1.1 percent points to recall, which reaches 85.47, but results in a minimal decrease of precision, causing no significant change in F-score. We benefit from including the siblings of the parent node of the punctuation sign: the positive F-score increases slightly, though not significantly.

Adding the leaf-ancestor path to the parent + grandparent combination causes a significant decrease of the F-score, as shown in the second part of the table. Adding it to the combination including siblings does increase the F-score to 88.30. However, this score is lower than the one using sibling information alone. As expected, + ccn is more important, these results significantly improve over the + p setting. Combining it with sibling information gives no significant difference.

In the third part of the table, we add information from the previous section that proved helpful: the focus punctuation sign, its neighbor punctuations, as well as the lexical context. When we add the focus to the combination baseline + parent + grand-parent + siblings + `CC`-daughter of parent, we reach the highest positive recall overall, 89.17, but the F-score does not improve significantly over + ccn. Adding all the information results in the same recall of 88.89 and in the highest precision (94.26), as well as in the highest positive F-score overall (91.50). These experiments show that crafting a successful feature combination is a difficult task; and they indicate that lexical features seem to partially provide a syntactic context when there is no syntactic information.

5.3 Using Parsing

Here, we investigate whether we can replicate the results above with syntactic information from the Berkeley parser. Table 4 presents the results based on parser

	positive				negative			
	precision	recall	fp-rate	F-score	precision	recall	fp-rate	F-score
baseline (pos)	87.31	80.34	2.12	83.68	96.49	97.88	19.66	97.18
+p	90.18	83.76	1.65	86.85**	97.09	98.35	16.24	97.72
+p+gp	91.33	84.05	1.45	87.54	97.15	98.55	15.95	97.85
+p+gp+sib	89.91	83.76	1.70	86.73	97.09	98.30	16.24	97.69
+ p + gp additionally:								
+la	89.84	80.63	1.65	84.98*	96.55	98.35	19.37	97.44
+sib+la	88.29	83.76	2.01	85.97	97.08	97.99	16.24	97.53
+ccn	91.02	83.76	1.50	87.24	97.10	98.50	16.24	97.80
+sib+ccn	90.21	84.05	1.65	87.02	97.14	98.35	15.95	97.74
+ p + gp + sib + ccn additionally:								
+focus	90.91	85.47	1.55	88.11	97.39	98.45	14.53	97.92
+focus+3l3r	92.09	82.91	1.29	87.26	96.96	98.71	17.09	97.83
+focus+neigh	92.64	**86.04**	1.24	**89.22***	**97.50**	98.76	13.96	**98.13**
all features	**93.40**	84.62	1.09	88.79*	97.26	**98.92**	15.39	98.08

Table 4: Results using the Berkeley parser with significance tested against baseline for + p, against + p for all other experiments; *=0.1; **=0.01.

information. For direct comparison, we report the same feature sets from table 3.

We see that some general trends are repeated: Adding the parent, grandparent, and focus information is helpful, adding the leaf-ancestor paths is not. Using gold syntax and baseline features results in an F-score of 84.34; the same experiment using parsing data results in an F-score of 83.68. Major decreases in positive F-score occur only with features which heavily rely on syntactic information, such as + ccn. However, the highest positive F-score achieved with the gold syntax (91.50) is just 2.2 points higher than the one based on parser data.

The lower quality of parse trees is particularly reflected in two features: Features concerning the siblings (+ sib), and, much more so, the parent of a `CC` daugther (+ ccn), are less reliable than their gold syntax counterparts. Instead, we now benefit more from the lexical material around the focus punctuation. The highest positive recall (86.04) and F-score (89.22) is reached by using parent + grandparent + sibling + focus + neighbors + `CC`-daughter information.

We now look at the results with best results from table 4 (+ p + gp + sib + ccn + foc + neigh), and we separate them based on the mother node. Table 5 lists the results for the most frequent categories (all other categories had considerably fewer instances in the test set). Coordinations under S nodes can be detected very reliably, probably because only overt conjunction cases are annotated, and these are the easier cases. NPs and VPs are more difficult because they include a larger range of phenomena: covert conjunction cases, appositions, and superfluous conjunction commas. However, further insights require a more detailed error analysis.

	#	precision	recall	F-score
S	735	97.12	94.39	95.73
VP	355	87.50	83.05	85.22
NP	912	90.84	83.80	87.18

Table 5: The results of the +p + gp + sib + ccn + foc + neigh experiment based on the type of mother node.

6 Conclusion and Future Work

We have presented work on automatically distinguishing punctuation signs functioning as separators for conjuncts in coordination from such that do not have a coordinating function. We used a version of the Penn Treebank in which punctuation signs are annotated whether they have a coordinating function or not. We have formulated the task of identifying the status of previously unseen punctuation as a binary classification problem. Using memory-based learning, we have achieved an F-score of 91.50 (98.51) on positive (negative) instances of punctuation using features drawn from the local context of the punctuation sign to be classified and from its surrounding syntactic context given by the treebank annotation. Even when using syntactic trees provided by a parser, i.e., without any gold information, we still achieve a F-score for positive (negative) instances of 89.22 (98.13).

Our experiments are a first step towards a reliable, cross-category identification of coordination with scope detection using supervised learning. Our goal is to include all types of coordination, even if no overt conjunction is present. We are also planning to include the information learned here into a parsing approach to improve parsing across all types of coordination.

References

[1] Shane Bergsma, David Yarowsky, and Kenneth Church. Using large monolingual and bilingual corpora to improve coordination disambiguation. In *ACL*, 2011.

[2] Francis Chantree, Adam Kilgarriff, Anne de Roeck, and Alistair Willis. Disambiguating coordinations using word distribution information. In *RANLP*, 2005.

[3] Walter Daelemans, Jakub Zavrel, Ko Van der Sloot, and Antal Van den Bosch. TiMBL: Tilburg Memory Based Learner, version 6.3, Reference Guide. ILK Research Group Technical Report Series 10-01, University of Tilburg, 2010.

[4] Robert Dale and Paweł Mazur. Handling conjunctions in named entities. In *Computational Linguistics and Intelligent Text Processing*, volume 4394 of *LNCS*, pages 131–142. Springer, 2007.

[5] Yuqing Guo, Haifeng Wang, and Josef van Genabith. Recovering non-local dependencies for Chinese. In *EMNLP-CoNLL*, 2007.

[6] Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. Coordinate structure analysis with global structural constraints and alignment-based local features. In *ACL-AFNLP*, 2009.

[7] Katharina Hartmann. *Right Node Raising and Gapping*. John Benjamins, 2000.

[8] Enrique Henestroza Anguiano and Marie Candito. Parse correction with specialized models for difficult attachment types. In *EMNLP*, 2011.

[9] Deirdre Hogan. Coordinate noun phrase disambiguation in a generative parsing model. In *ACL*, 2007.

[10] Thorsten Joachims. SVM^{light} support vector machine, 2002. `http://svmlight.joachims.org`.

[11] Daisuke Kawahara and Sadao Kurohashi. Generative modeling of coordination by factoring parallelism and selectional preferences. In *IJCNLP*, 2011.

[12] Adam Kilgarriff, Pavel Rychlý, Pavel Smrž, and David Tugwell. The Sketch Engine. In *EURALEX*, 2004.

[13] Sandra Kübler, Erhard W. Hinrichs, Wolfgang Maier, and Eva Klett. Parsing coordinations. In *EACL*, 2009.

[14] Wolfgang Maier, Sandra Kübler, Erhard Hinrichs, and Julia Krivanek. Annotating coordination in the Penn Treebank. In *ACL-LAW VI*, 2012.

[15] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Comp. Ling.*, 19(2):313–330, 1993.

[16] Andrew Kachites McCallum. MALLET: A machine learning for language toolkit, 2002. `http://mallet.cs.umass.edu`.

[17] Philip Ogren. Improving syntactic coordination resolution using language modeling. In *NAACL HLT SRW*, 2010.

[18] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *COLING-ACL*, 2006.

[19] Geoffrey Sampson and Anna Babarczy. A test of the leaf-ancestor metric for parse accuracy. *Journal of Natural Language Engineering*, 9:365–380, 2003.