A Neural Network for Coordination Boundary Prediction

Jessica Ficler
Computer Science Department
Bar-Ilan University
Israel
jessica.ficler@gmail.com

Yoav Goldberg
Computer Science Department
Bar-Ilan University
Israel
yoav.goldberg@gmail.com

Abstract

We propose a neural-network based model for coordination boundary prediction. The network is designed to incorporate two signals: the similarity between conjuncts and the observation that replacing the whole coordination phrase with a conjunct tends to produce a coherent sentences. The modeling makes use of several LSTM networks. The model is trained solely on conjunction annotations in a Treebank, without using external resources. We show improvements on predicting coordination boundaries on the PTB compared to two state-of-the-art parsers; as well as improvement over previous coordination boundary prediction systems on the Genia corpus.

1 Introduction

Coordination is a common syntactic phenomena, appearing in 38.8% of the sentences in the Penn Treebank (PTB) (Marcus et al., 1993), and in 60.71% of the sentences in the Genia Treebank (Ohta et al., 2002). However, predicting the correct conjuncts span remain one of the biggest challenges for state-of-the-art syntactic parsers. Both the Berkeley and Zpar phrase-structure parsers (Petrov et al., 2006; Zhang and Clark, 2011) achieve F1 scores of around 69% when evaluated on their ability to recover coordination boundaries on the PTB test set. For example, in:

"He has the government's blessing to [build churches] and [spread Unificationism] in that country."

the conjuncts are incorrectly predicted by both parsers:

Berkeley: *"He has the government's blessing to [build churches] and [spread Unificationism in that country]."*
Zpar: *"He [has the government's blessing to build churches] and [spread Unificationism in that country]."*

In this work we focus on coordination boundary prediction, and suggest a specialized model for this task. We treat it as a ranking task, and learn a scoring function over conjuncts candidates such that the correct candidate pair is scored above all other candidates. The scoring model is a neural network with two LSTM-based components, each modeling a different linguistic principle: (1) conjuncts tend to be similar ("symmetry"); and (2) replacing the coordination phrase with each of the conjuncts usually result in a coherent sentence ("replacement"). The *symmetry* component takes into account the conjuncts' syntactic structures, allowing to capture similarities that occur in different levels of the syntactic structure. The *replacement* component considers the coherence of the sequence that is produced when connecting the participant parts. Both of these signals are syntactic in nature, and are learned solely based on information in the Penn Treebank. Our model substantially outperforms both the Berkeley and Zpar parsers on the coordination prediction task, while using the exact same training corpus. Semantic signals (which are likely to be based on resources external to the treebank) are also relevant for coordination disambiguation (Kawahara and Kurohashi, 2008; Hogan, 2007) and provide complementary information. We plan to incorporate such signals in future work.

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 23–32,
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics

2 Background

Coordination is a very common syntactic construction in which several sentential elements (called conjuncts) are linked. For example, in:

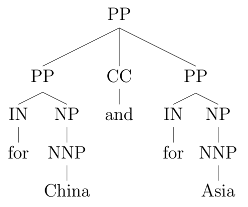
"The Jon Bon Jovi Soul Foundation [was founded in 2006] and$_1$ [exists to combat issues that force (families) and$_2$ (individuals) into economic despair]."

The coordinator *and*$_1$ links the conjuncts surrounded with square brackets and the coordinator *and*$_2$ links the conjuncts surrounded with round brackets.

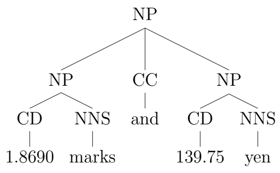
Coordination between NPs and between VPs are the most common, but other grammatical functions can also be coordinated: *"[relatively active]$_{ADJP}$ but [unfocused]$_{ADJP}$"* ; *"[in]$_{IN}$ and [out]$_{IN}$ the market"*. While coordination mostly occurs between elements with the same syntactic category, cross-category conjunctions are also possible: (*"Alice will visit Earth [tomorrow]$_{NP}$ or [in the next decade]$_{PP}$"*). Less common coordinations involve non-constituent elements *"[equal to] or [higher than]"*, argument clusters (*"Alice visited [4 planets] [in 2014] and [3 more] [since then]"*), and gapping (*"[Bob lives on Earth] and [Alice on Saturn]"*) (Dowty, 1988).

2.1 Symmetry between conjuncts

Coordinated conjuncts tend to be semantically related and have a similar syntactic structure. For example, in (a) and (b) the conjuncts include similar words (China/Asia, marks/yen) and have identical syntactic structures.

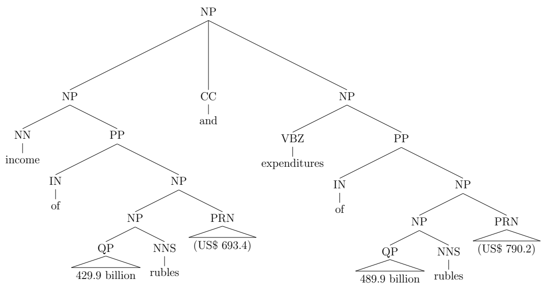


(a)



(b)

Symmetry holds also in larger conjuncts, such as in:



(c)

Similarity between conjuncts was used as a guiding principle in previous work on coordination disambiguation (Hogan, 2007; Shimbo and Hara, 2007; Hara et al., 2009).

2.2 Replaceability

Replacing a conjunct with the whole coordination phrase usually produce a coherent sentence (Huddleston et al., 2002). For example, in *"Ethan has developed [new products] and [a new strategy]"*, replacement results in: *"Ethan has developed new products"*; and *"Ethan has developed a new strategy"*, both valid sentences. Conjuncts replacement holds also for conjuncts of different syntactic types, e.g.: *"inactivation of tumor-suppressor genes, [alone] or [in combination], appears crucial to the development of such scourges as cancer."*.

While both symmetry and replacebility are strong characteristics of coordination, neither principle holds universally. Coordination between syntactically dissimilar conjuncts is possible ("tomorrow and for the entirety of the next decade"), and the replacement principle fails in cases of ellipsis, gapping and others (*"The bank employs [8,000 people in Spain] and [2,000 abroad]"*).

2.3 Coordination in the PTB

Coordination annotation in the Penn Treebank (Marcus et al., 1993) is inconsistent (Hogan, 2007) and lacks internal structure for NPs with nominal modifiers (Bies et al., 1995). In addition, conjuncts in the PTB are not explicitly marked. These deficiencies led previous works on coordination disambiguation (Shimbo and Hara, 2007; Hara et al., 2009; Hanamoto et al., 2012) to use the Genia treebank of biomedical text (Ohta et al., 2002) which explicitly marks coordination phrases. However, using the Genia corpus is not ideal since it is in a specialized domain and much smaller than the PTB. In this work we rely on a version of the PTB released by Ficler and Goldberg (2016) in which the above deficiencies are manually resolved. In particular, coordinating elements, coordination phrases and conjunct boundaries are explicitly marked with specialized function labels.

2.4 Neural Networks and Notation

We use $w_{1:n}$ to indicate a list of vectors $w_1, w_2, \dots w_n$ and $w_{n:1}$ to indicate the reversed list. We use $\circ$ for vector concatenation. When a discrete symbol w is used as a neural network's input, the corresponding embedding vector is assumed.

A multi-layer perceptron (MLP) is a non linear classifier. In this work we take MLP to mean a classifier with a single hidden layer: $MLP(x) = V \cdot g(Wx + b)$ where x is the network's input, g is an activation function such as ReLU or Sigmoid, and W, V and b are trainable parameters. Recurrent Neural Networks (RNNs) (Elman, 1990) allow the representation of arbitrary sized sequences. In this work we use LSTMs (Hochreiter and Schmidhuber, 1997), a variant of RNN that was proven effective in many NLP tasks. LSTM$(w_{1:n})$ is the outcome vector resulting from feeding the sequence $w_{1:n}$ into the LSTM in order. A bi-directional LSTM (biLSTM) takes into account both the past $w_{1:i}$ and the future $w_{i:n}$ when representing the element in position i:

$$biLSTM(w_{1:n}, i) = LSTM_F(w_{1:i}) \circ LSTM_B(w_{n:i})$$

where $LSTM_F$ reads the sequence in its regular order and $LSTM_B$ reads it in reverse.

3 Task Definition and Architecture

Given a coordination word in a sentence, the coordination prediction task aims to returns the two conjuncts that are connected by it, or NONE if the word does not function as a coordinating conjunction of a relevant type.[1] Figure 1 provides an example.

Our system works in three phases: first, we determine if the coordinating word is indeed part of a conjunction of a desired type. We then extract a ranked list of candidate conjuncts, where a candidate is a pair of spans of the form $((i, j), (l, m))$. The candidates are then scored and the highest scoring pair is returned. Section 4 describes the scoring model, which is the main contribution of this work. The coordination classification and candidate extraction components are described in Section 5.

Sentence:
And$_1$ the city decided to treat its guests more like royalty or$_2$ rock stars than factory owners.
Expected output:
and$_1$: NONE
or$_2$: (11-11) royalty ; (12-13) rock stars

Sentence:
The president is expected to visit Minnesota, New York and$_1$ North Dakota by the end of the year.
Expected output:
and$_1$: (9-10) New York ; (12-13) North Dakota

Figure 1: The coordination prediction task.

4 Candidate Conjunctions Scoring

Our scoring model takes into account two signals, symmetry between conjuncts and the possibility of replacing the whole coordination phrase with its participating conjuncts.

4.1 The Symmetry Component

As noted in Section 2.1, many conjuncts spans have similar syntactic structure. However, while the similarity is clear to human readers, it is often not easy to formally define, such as in:

"about/IN half/NN its/PRP$ revenue/NN
and/CC
more/JJR than/IN half/NN its/PRP$ profit/NN"

If we could score the amount of similarity between two spans, we could use that to identify correct coordination structures. However, we do not know the similarity function. We approach this by training the similarity function in a data-dependent manner. Specifically, we train an encoder that encodes spans into vectors such that vectors of similar spans will have a small Euclidean distance between them. This architecture is similar to Siamese Networks, which are used for learning similarity functions in vision tasks (Chopra et al., 2005).

[1] We consider *and*, *or*, *but*, *nor* as coordination words. In case of more than two coordinated elements (conjuncts), we focus on the two conjuncts which are closest to the coordinator.

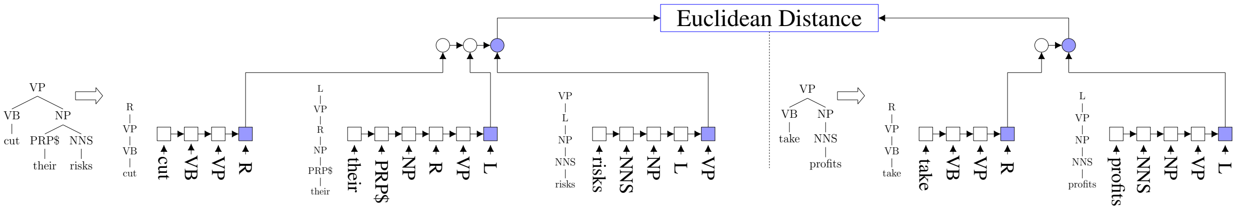


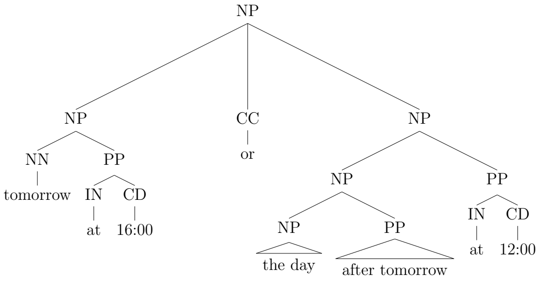
Figure 2: Illustration of the symmetry scoring component that takes into account the conjuncts syntactic structures. Each conjunct tree is decomposed into paths that are fed into the path-LSTMs (squares). The resulting vectors are fed into the symmetry LSTM function (circles). The outcome vectors (blue circles) are then fed into the euclidean distance function.

Given two spans of lengths k and m with corresponding vector sequences $u_{1:k}$ and $v_{1:m}$ we encode each sequences using an LSTM, and take the euclidean distance between the resulting representations:

$$Sym(u_{1:k}, v_{1:m}) = ||LSTM(u_{1:k}) - LSTM(v_{1:m})||$$

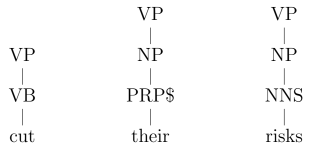
The network is trained such that the distance is minimized for compatible spans and large for incompatible ones in order to learn that vectors that represent correct conjuncts are closer than vectors that do not represent conjuncts.

What are the elements in the sequences to be compared? One choice is to take the vectors u_i to correspond to embeddings of the ith POS in the span. This approach works reasonably well, but does not consider the conjuncts' syntactic structure, which may be useful as symmetry often occurs on a higher level than POS tags. For example, in:

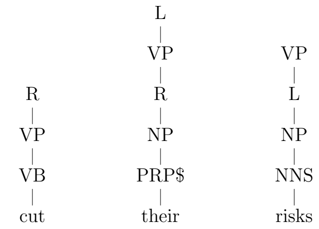


the similarity is more substantial in the third level of the tree than in the POS level.

A way to allow the model access to higher levels of syntactic symmetry is to represent each word as the projection of the grammatical functions from the word to the root.[2] For example, the projections for the first conjunct in Figure 2 are:



This decomposition captures the syntactic context of each word, but does not uniquely determine the structure of the tree. To remedy this, we add to the paths special symbols, R and L, which marks the lowest common ancestors with the right and left words respectively. These are added to the path above the corresponding nodes. For example consider the following paths which corresponds to the above syntactic structure:



The lowest common ancestor of *"their"* and *"risks"* is NP. Thus, R is added after NP in the path of *"their"* and L is added after NP in the path of *"risks"*. Similarly, L and R are added after the VP in the *"their"* and *"cut"* paths.

The path for each word is encoded using an LSTM receiving vector embeddings of the elements in the path from the word to the root. We then use the resulting encodings instead of the POS-tag embeddings as input to the LSTMs in the similarity function. Figure 2 depicts the complete process for the spans "cut their risks" and "take profits".

Using syntactic projections requires the syntactic structures of the conjuncts. This is obtained by running the Berkeley parser over the sentence and taking the subtree with the highest probability from the

[2] Similar in spirit to the spines used in Carreras et al. (2008) and Shen et al. (2003).

Sentence:

$$\underbrace{\text{Rudolph }\underset{w_{i-1}}{\text{Agnew,}}}_{Pre}\ \underbrace{\underset{w_i}{\text{[55}}\text{ years }\underset{w_j}{\text{old]}}}_{Conj1}\ \underset{w_k}{\text{and}}\ \underbrace{\underset{w_l}{\text{[former}}\text{ chairman of CGF }\underset{w_m}{\text{PLC]}}}_{Conj2}\ \underbrace{\underset{w_{m+1}}{\text{,was}}\text{ named a nonexecutive director.}}_{Post}$$

Expansions:

Rudolph Agnew, 55 years old ,was named a nonexecutive director.
Rudolph Agnew, former chairman of CGF PLC ,was named a nonexecutive director.

Figure 3: The correct conjuncts spans of the coordinator *and* in the sentence and the outcome expansions.

corresponding cell in the CKY chart.[3] In both approaches, the POS embeddings are initialized with vectors that are pre-trained by running word2vec (Mikolov et al., 2013) on the POS sequences in PTB training set.

4.2 The Replacement Component

The replacement component is based on the observation that, in many cases, the coordination phrase can be replaced with either one of its conjuncts while still preserving a grammatical and semantically coherent sentence (Section 2.2)

When attempting such a replacement on incorrect conjuncts, the resulting sentence is likely to be either syntactically or semantically incorrect. For example, in the following erroneous analysis: *"Rudolph Agnew, [55 years old] and [former chairman] of Consolidated Gold Fields PLC"* replacing the conjunction with the first conjunct results in the semantically incoherent sequence *"Rudolph Agnew, 55 years old of Consolidated Golden Fields, PLC"*.[4]

Our goal is to distinguish replacements resulting from correct conjuncts from those resulting from erroneous ones. To this end, we focus on the *connection points*. A connection point in a resulting sentence is the point where the sentence splits into two sequences that were not connected in the original sentence. For example, consider the sentence in Figure 3. It has four parts, marked as *Pre*, *Conj1*, *Conj2* and *Post*. Replacing the coordination phrase *Conj1 and Conj2* with *Conj2* results in a connection point between *Pre* and *Conj2*. Likewise, replacing the coordination phrase with *Conj1* results in connection point between *Conj1* and *Post*.

In order to model the validity of the connection points, we represent each connection point as the concatenation of a forward and reverse LSTMs centered around that point. Specifically, for the spans in Figure 3 the two connection points are represented as:

$LSTM_F$(Rudolph,...,old)$\circ LSTM_B$(director,...,was,,)
and
$LSTM_F$(Rudolph,Agnew,,)$\circ LSTM_B$(director,...,former)

Formally, assuming words $w_{1:n}$ in a sentence with coordination at position k and conjuncts $w_{i:j}$ and $w_{l:m}$,[5] the connection points are between $w_{1:j}$ and $w_{m+1:n}$; and between $w_{1:i-1}$ and $w_{l:n}$. The two connection points representations are then concatenated, resulting in a *replacement vector*:

$$\begin{aligned}&\textsc{Repl}(w_{1:n}, i, j, l, m) = \\ &\qquad \textsc{ConPoint}(w_{1:n}, i-1, l) \circ \textsc{ConPoint}(w_{1:n}, j, m+1)\end{aligned}$$

where:

$$\begin{aligned}&\textsc{ConPoint}(w_{1:n}, i, j) = \\ &\qquad LSTM_F(w_{1:i}) \circ LSTM_B(w_{n:j})\end{aligned}$$

We use two variants of the replacement vectors, corresponding to two levels of representation. The first variant is based on the sentence's words, while the second is based on its POS-tags.

4.3 Parser based Features

In addition to the symmetry and replacement signals, we also incorporate some scores that are derived from the Berkeley parser. As detailed in Section 5, a list of conjuncts candidates are extracted

[3] The parser's CKY chart did not include a tree for 10% of the candidate spans, which have inside probability 0 and outside probability > 0. For those, we obtained the syntactic structure by running the parser on the span words only.

[4] While uncommon, incorrect conjuncts may also result in valid sentences, e.g. *"He paid \$ 7 for cold [drinks] and [pizza] that just came out of the oven."*

[5] Usually $j = k - 1$ and $l = k + 1$, but in some cases punctuation symbols may interfere.

from the CKY chart of the parser. The candidates are then sorted in descending order according to the multiplication of inside and outside scores of the candidate's spans:[6] $I_{(i,j)} \times O_{(i,j)} \times I_{(l,m)} \times O_{(l,m)}$. Each candidate $\{(i,j),(l,m)\}$ is assigned two numerical features based on this ranking: its position in the ranking, and the ratio between its score and the score of the adjacent higher-ranked candidate. We add an additional binary feature indicating whether the candidate spans are in the 1-best tree predicted by the parser. These three features are denoted as $Feats(i,j,l,m)$.

4.4 Final Scoring and Training

Finally, the score of a candidate $\{(i,j),(l,m)\}$ in a sentence with words $w_{1:n}$ and POS tags $p_{1:n}$ is computed as:

$$\begin{aligned}&\textsc{Score}(w_{1:n}, p_{1:n}, \{(i,j),(l,m)\}) = \\ &\quad MLP(\\ &\qquad Sym(v^{Path}_{i:j}, v^{Path}_{l:m}) \\ &\qquad \circ\, Repl(w_{1:n}, i, j, l, m) \\ &\qquad \circ\, Repl(p_{1:n}, i, j, l, m) \\ &\qquad \circ\, Feats(i, j, l, m)\ \ )\end{aligned}$$

where $v^{Path}_{i:j}$ and $v^{Path}_{l:m}$ are the vectors resulting from the path LSTMs, and Sym, $Repl$ and $Feats$ are the networks defined in Sections 4.1 – 4.3 above. The network is trained jointly, attempting to minimize a pairwise ranking loss function, where the loss for each training case is given by:

$$loss = max(0, 1 - (\hat{y} - y_g))$$

where $\hat{y}$ is the highest scoring candidate and y_g is the correct candidate. The model is trained on all the coordination cases in Section 2–21 in the PTB.

5 Candidates Extraction and Supporting Classifiers

Candidates Extraction We extract candidate spans based on the inside-outside probabilities assigned by the Berkeley parser. Specifically, to obtain candidates for conjunct span we collect spans that are marked with COORD, are adjacent to the coordinating word, and have non-zero inside or outside probabilities. We then take as candidates all possible pairs of collected spans. On the PTB dev set, this method produces 6.25 candidates for each coordinating word on average and includes the correct candidates for 94% of the coordinations.

Coordination Classification We decide whether a coordination word w_k in a sentence $w_{1:n}$ functions as a coordinator by feeding the biLSTM vector centered around w_k into a logistic classifier:

$$\sigma(v \cdot \text{biLSTM}(w_{1:n}, k) + b).$$

The training examples are all the coordination words (marked with CC) in the PTB training set. The model achieves 99.46 F1 on development set and 99.19 F1 on test set.

NP coordinations amount to about half of the coordination cases in the PTB, and previous work is often evaluated specifically on NP coordination. When evaluating on NP coordination, we depart from the unrealistic scenario used in most previous work where the type of coordination is assumed to be known a-priori, and train a specialized model for predicting the coordination type. For a coordination candidate $\{(i,j),(l,m)\}$ with a coordinator w_k, we predict if it is NP coordination or not by feeding a logistic classifier with a biLSTM vector centered around the coordinator and constrained to the candidate spans:

$$\sigma(v \cdot \text{biLSTM}(w_{i:m}, k) + b).$$

The training examples are coordinations in the PTB training set, where where a coordinator is considered of type NP if its head is labeled with NP or NX. Evaluating on gold coordinations results in F1 scores of 95.06 (dev) and 93.89 (test).

6 Experiments

We evaluate our models on their ability to identify conjunction boundaries in the extended Penn Treebank (Ficler and Goldberg, 2016) and Genia Treebank (Ohta et al., 2002)[7].

When evaluating on the PTB, we compare to the conjunction boundary predictions of the generative

[6] Inside-Outside probabilities (Goodman, 1998) represent the probability of a span with a given non-terminal symbol. The inside probability $I_{(N,i,j)}$ is the probability of generating words $w_i, w_{i+1}, ..., w_j$ given that the root is the non-terminal N. The outside probability $O_{(N,i,j)}$ is the probability of generating words $w_1, w_2, ..., w_{i-1}$, the non-terminal N and the words $w_{j+1}, w_{j+2}, ..., w_n$ with the root S.

[7] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA

	Dev			Test		
	P	R	F	P	R	F
Berkeley	70.14	70.72	70.42	68.52	69.33	68.92
Zpar	72.21	72.72	**72.46**	68.24	69.42	68.82
Ours	72.34	72.25	72.29	72.81	72.61	**72.7**

Table 1: Coordination prediction on PTB (All coordinations).

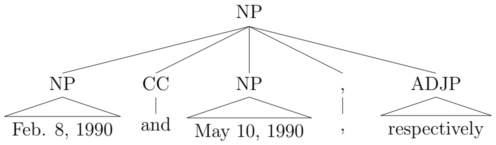
	Dev			Test		
	P	R	F	P	R	F
Berkeley	67.53	70.93	69.18	69.51	72.61	71.02
Zpar	69.14	72.31	70.68	69.81	72.92	71.33
Ours	75.17	74.82	**74.99**	76.91	75.31	**76.1**

Table 2: Coordination prediction on PTB (NP coordinations).

Berkeley parser (Petrov et al., 2006) and the discriminative Zpar parser (Zhang and Clark, 2011). When evaluating on the Genia treebank, we compare to the results of the discriminative coordination-prediction model of Hara et al. (2009).[8]

6.1 Evaluation on PTB

Baseline Our baseline is the performance of the Berkeley and Zpar parsers on the task presented in Section 3, namely: for a given coordinating word, determine the two spans that are being conjoined by it, and return None if the coordinator is not conjoining spans or conjoins spans that are not of the expected type. We convert predicted trees to conjunction predictions by taking the two phrases that are immediately adjacent to the coordinator on both sides (ignoring phrases that contain solely punctuation). For example, in the following Zpar-predicted parse tree the conjunct prediction is (*"Feb. 8, 1990"*,*"May 10, 1990"*).



Cases in which the coordination word is the left-most or right-most non-punctuation element in its phrase (e.g. `(PRN (P -)(CC and)(S it's been painful)(P -)))` are considered as no-coordination ("None").

We consider two setups. In the first we are interested in all occurrences of coordination, and in the second we focus on NP coordination. The second scenario requires typed coordinations. We take the type of a parser-predicted coordination to be the type of the phrase immediately dominating the coordination word.

Evaluation Metrics We measure precision and recall compared to the gold-annotated coordination spans in the extended PTB, where an example is considered correct if both conjunct boundaries match exactly. When focusing on NPs coordinations, the type of the phrase above the CC level must match as well, and phrases of type NP/NX are considered as NP coordination.

Results Tables (1) and (2) summarize the results. The Berkeley and Zpar parsers perform similarly on the coordination prediction task. Our proposed model outperforms both parsers, with a test-set F_1 score of 72.7 (3.78 F_1 points gain over the better parser) when considering all coordinations, and test-set F_1 score of 76.1 (4.77 F_1 points gain) when considering NP coordination.

6.2 Evaluation on Genia

To compare our model to previous work, we evaluate also on the Genia treebank (Beta), a collection of constituency trees for 4529 sentences from Medline abstracts. The Genia treebank coordination annotation explicitly marks coordination phrases with a special function label (COOD), making the corpus an appealing resource for previous work on coordination boundary prediction (Shimbo and Hara, 2007; Hara et al., 2009; Hanamoto et al., 2012). Following Hara et al. (2009), we evaluate the models' ability to predict the span of the entire coordination phrase, disregarding the individual conjuncts. For example, in *"My plan is to visit Seychelles, ko Samui and Sardinia by the end of the year"* the goal is to recover *"Seychelles, ko Samui and Sardinia"*. This is a recall measure. We follow the exact protocol of Hara et al. (2009) and train and evaluate the model on 3598 coordination phrases in Genia Treebank Beta and report the micro-averaged results of a five-fold cross validation run.[9] As shown by Hara

[8] Another relevant model in the literature is (Hanamoto et al., 2012), however the results are not directly comparable as they use a slightly different definition of conjuncts, and evaluate on a subset of the Genia treebank, containing only trees that were properly converted to an HPSG formalism.

[9] We thank Kazuo Hara for providing us with the exact details of their splits.

Sym	Correct:	Retail sales volume was [down 0.5% from the previous three months] and [up 1.2% from a year earlier].
	Incorrect:	Everyone was concerned about the [general narrowness of the rally] and [failure of the OTC market] to get into plus territory.
Rep$_w$	Correct:	The newsletter said [she is 44 years old] and [she studied at the University of Puerto Rico School of Medicine].
	Incorrect:	But Robert Showalter said no special [bulletins] or [emergency meetings of the investors' clubs] are planned .
Rep$_p$	Correct:	[On the Big Board floor] and [on trading desks], traders yelped their approval.
	Incorrect:	It suddenly burst upward 7.5 as Goldman, Sachs & Co. [stepped in] and [bought almost] every share offer, traders said.

Figure 4: Correct in incorrect predictions by the individual components.

COOD	#	Our Model	Hara et al.
Overall	3598	**64.14**	61.5
NP	2317	**65.08**	64.2
VP	465	**71.82**	54.2
ADJP	321	74.76	**80.4**
S	188	17.02	**22.9**
PP	167	56.28	**59.9**
UCP	60	**51.66**	36.7
SBAR	56	**91.07**	51.8
ADVP	21	80.95	**85.7**
Others	3	33.33	**66.7**

Table 3: Recall on the Beta version of Genia corpus. Numbers for Hara et al. are taken from their paper.

	All types			NPs		
	P	R	F	P	R	F
Sym	67.13	67.06	67.09	69.69	72.08	70.86
Rep$_p$	69.26	69.18	69.21	69.73	71.16	70.43
Rep$_w$	56.97	56.9	56.93	59.78	64.3	61.95
Feats	70.92	70.83	70.87	72.23	73.22	72.72
Joint	72.34	72.25	**72.29**	75.17	74.82	**74.99**

Table 4: Performance of the individual components on PTB section 22 (dev). Sym: Symmetry. Rep$_p$: POS replacement. Rep$_w$: Word replacement. Feats: features extracted from Berkeley parser. Joint: the complete model.

et al. (2009), syntactic parsers do not perform well on the Genia treebank. Thus, in our symmetry component we opted to not rely on predicted tree structures, and instead use the simpler option of representing each conjunct by its sequence of POS tags. To handle coordination phrases with more than two conjuncts, we extract candidates which includes up to 7 spans and integrate the first and the last span in the model features. Like Hara et al., we use gold POS.

Results Table 3 summarizes the results. Our proposed model achieves Recall score of 64.14 (2.64 Recall points gain over Hara et al.) and significantly improves the score of several coordination types.

6.3 Technical Details

The neural networks (candidate scoring model and supporting classifiers) are implemented using the pyCNN package.[10].

In the supporting models we use words embedding of size 50 and the Sigmoid activation function. The LSTMs have a dimension of 50 as well. The models are trained using SGD for 10 iterations over the train-set, where samples are randomly shuffled before each iteration. We choose the model with the highest F1 score on the development set.

All the LSTMs in the candidate scoring model have a dimension of 50. The input vectors for the symmetry LSTM is of size 50 as well. The MLP in the candidate scoring model uses the Relu activation function, and the model is trained using the Adam optimizer. The words and POS embeddings are shared between the symmetry and replacment components. The syntactic label embeddings are for the path-encoding LSTM, We perform grid search with 5 different seeds and the following: [1] MLP hidden layer size (100, 200, 400); [2] input embeddings size for words, POS and syntactic labels (100, 300). We train for 20 iterations over the train set, randomly shuffling the examples before each iteration. We choose the model that achieves the highest F1 score on the dev set.

7 Analysis

Our model combines four signals: symmetry, word-level replacement, POS-level replacement and features from Berkeley parser. Table 4 shows the PTB dev-set performance of each sub-model in isolation. On their own, each of the components' signals is relatively weak, seldom outperforming the parsers. However, they provide complementary information, as evident by the strong performance of the joint model. Figure 4 lists correct and incorrect predictions by each of the components, indicating that the individual models are indeed capturing the patterns they were designed to capture – though these patterns do not always lead to correct predictions.

[10] https://github.com/clab/cnn/tree/master/pycnn

8 Related Work

The similarity property between conjuncts was explored in several previous works on coordination disambiguation. Hogan (2007) incorporated this principle in a generative parsing model by changing the generative process of coordinated NPs to condition on properties of the first conjunct when generating the second one. Shimbo and Hara (2007) proposed a discriminative sequence alignment model to detect similar conjuncts. They focused on disambiguation of non-nested coordination based on the learned edit distance between two conjuncts. Their work was extended by Hara et al. (2009) to handle nested coordinations as well. The discriminative edit distance model in these works is similar in spirit to our symmetry component, but is restricted to sequences of POS-tags, and makes use of a sequence alignment algorithm. We compare our results to Hara et al.'s in Section 6.2. Hanamoto et al. (2012) extended the previous method with dual decomposition and HPSG parsing. In contrast to these symmetry-directed efforts, Kawahara et al. (2008) focuses on the dependency relations that surround the conjuncts. This kind of semantic information provides an additional signal which is complementary to the syntactic signals explored in our work. Our neural-network based model easily supports incorporation of additional signals, and we plan to explore such semantic signals in future work.

9 Conclusions

We presented an neural-network based model for resolving conjuncts boundaries. Our model is based on the observation that (a) conjuncts tend to be similar and (b) that replacing the coordination phrase with a conjunct results in a coherent sentence. Our models rely on syntactic information and do not incorporate resources external to the training treebanks, yet improve over state-of-the-art parsers on the coordination boundary prediction task.

Acknowledgments

This work was supported by The Israeli Science Foundation (grant number 1555/15) as well as the German Research Foundation via the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

References

Ann Bies, Mark Ferguson, Karen Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz, and Britta Schasberger. 1995. Bracketing guidelines for treebank ii style penn treebank project. *University of Pennsylvania*, 97:100.

Xavier Carreras, Michael Collins, and Terry Koo. 2008. Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 9–16. Association for Computational Linguistics.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE.

David Dowty. 1988. Type raising, functional composition, and non-constituent conjunction. In *Categorial grammars and natural language structures*, pages 153–197. Springer.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Jessica Ficler and Yoav Goldberg. 2016. Coordination annotation extension in the penn tree bank. *Association for Computational Linguistics*.

Joshua Goodman. 1998. Parsing inside-out. *arXiv preprint cmp-lg/9805007*.

Atsushi Hanamoto, Takuya Matsuzaki, and Jun'ichi Tsujii. 2012. Coordination structure analysis using dual decomposition. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 430–438. Association for Computational Linguistics.

Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. 2009. Coordinate structure analysis with global structural constraints and alignment-based local features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 967–975. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Deirdre Hogan. 2007. Coordinate noun phrase disambiguation in a generative parsing model. Association for Computational Linguistics.

Rodney Huddleston, Geoffrey K Pullum, et al. 2002. The cambridge grammar of english. *Language. Cambridge: Cambridge University Press*, page 1275.

Daisuke Kawahara and Sadao Kurohashi. 2008. Coordination disambiguation without any similarities. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 425–432. Association for Computational Linguistics.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the second international conference on Human Language Technology Research*, pages 82–86. Morgan Kaufmann Publishers Inc.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics.

Libin Shen, Anoop Sarkar, and Aravind K Joshi. 2003. Using ltag based features in parse reranking. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 89–96. Association for Computational Linguistics.

Masashi Shimbo and Kazuo Hara. 2007. A discriminative learning model for coordinate conjunctions. In *EMNLP-CoNLL*, pages 610–619.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational linguistics*, 37(1):105–151.