Open Information Extraction from Conjunctive Sentences

Swarnadeep Saha[*]
IBM Research - India
swarnads@in.ibm.com

Mausam
Indian Institute of Technology, Delhi
mausam@cse.iitd.ac.in

Abstract

We develop CALM, a coordination analyzer that improves upon the conjuncts identified from dependency parses. It uses a language model based scoring and several linguistic constraints to search over hierarchical conjunct boundaries (for nested coordination). By splitting a conjunctive sentence around these conjuncts, CALM outputs several simple sentences. We demonstrate the value of our coordination analyzer in the end task of Open Information Extraction (Open IE).

State-of-the-art Open IE systems lose substantial yield due to ineffective processing of conjunctive sentences. Our Open IE system, CALMIE, performs extraction over the simple sentences identified by CALM to obtain up to 1.8x yield with a moderate increase in precision compared to extractions from original sentences.

1 Introduction

Open Information Extraction (Open IE) (Etzioni et al., 2008) extracts relational tuples from text in an unsupervised domain-independent manner, by identifying relational phrases and arguments from the sentences themselves. Recent work (Saha et al., 2017) has highlighted the lack of proper conjunction processing as the most significant source of missed yield in Open IE. We found Open IE 4.2 (Christensen et al., 2011; Pal and Mausam, 2016) and ClausIE (Corro and Gemulla, 2013) to frequently miss important extractions due to conjunctive relation phrases (see Table 1), and occasionally output conjunctive arguments, which are not ideal for readability or downstream applications (Angeli et al., 2015; Stanovsky et al., 2016a).

Most modern Open IE systems process dependency parses to obtain extractions. However, dependency parsers frequently make errors in resolving coordination ambiguity. Predicting the correct conjunct span is considered to be one of the biggest challenges for parsers (Ficler and Goldberg, 2016). The state of the art approach by Ficler and Goldberg (2016) trains an LSTM-based network for predicting the boundaries for the two conjuncts on either side of the coordinating conjunction, but does not handle cases where a conjunction coordinates more than two conjuncts.

Contributions: We propose a novel coordination analyzer called CALM (**C**oordination **A**nalyzer using **L**anguage **M**odel), which *corrects* the typical errors made by dependency parsers (specifically Clear parser, which is used in Open IE 4.2) using additional features and linguistic constraints. Under the intuition that one can split a conjunction to form multiple coherent simple sentences (see Table 2), CALM scores each simple sentence using a (modified) language model. It additionally employs several linguistic constraints to reduce errors further. An important linguistic constraint is that multiple coordination structures in a sentence must either be disjoint or completely nested. A key contribution of our work is operationalizing this constraint through a novel *hierarchical coordination tree*, which is helpful in sentences with multiple coordinations. Experiments on 577 conjunctive sentences in British News Corpus demonstrate that CALM improves upon the conjuncts from the parser, with significant benefits for sentences with multiple coordinations.

---

[*]Most work was done when Swarnadeep Saha was a graduate student at Indian Institute of Technology, Delhi.

Extraction	Systems
"Gates, an American investor and co-founder of Microsoft, stepped down as CEO of Microsoft in January 2000, but remained as chairman and created the position of chief software architect for himself and transferred his duties to Ray Ozzie and Craig Mundie."	
1. (Gates; stepped down as; CEO of Microsoft)	[Cm, O4, C]
2. (Gates; stepped down as CEO of Microsoft; in January 2000)	[Cm, O4]
3. (Gates; is; an American investor)	[Cm]
4. (Gates; is an investor from; United States)	[Cm, O4]
5. (Gates; is co-founder of; Microsoft)	[Cm]
6. (Gates; is; an American investor and co-founder of Microsoft)	[C]
7. (Gates; remained as; chairman)	[Cm, O4, C]
8. (Gates; created; the position of chief software architect for himself)	[Cm, O4, C]
9. (Gates; transferred; his duties)	[Cm]
10. (Gates; transferred his duties to; Ray Ozzie)	[Cm]
11. (Gates; transferred his duties to; Craig Mundie)	[Cm]
12. (His; has; duties)	[C]
13. (Gates; transferred his duties to Ray Ozzie; the position of chief software architect for himself)	[C]
14. (Gates; transferred his duties to Craig Mundie; the position of chief software architect for himself)	[C]

Table 1: Comparison of extractions of different systems on a conjunctive sentence. [Cm] : CALMIE(O) , [O4] : Open IE4.2, [C] : ClausIE. Green = correct, Red = incorrect.

Gates, an American investor, stepped down as CEO of Microsoft in January 2000.
Gates, co-founder of Microsoft, stepped down as CEO of Microsoft in January 2000.
Gates remained as chairman.
Gates created the position of chief software architect for himself.
Gates transferred his duties to Ray Ozzie.
Gates transferred his duties to Craig Mundie.

Table 2: Simple sentences generated by our system for the conjunctive sentence in Table 1.

We use CALM's output for the goal of improving Open IE. Our system, CALMIE, splits a sentence into multiple simple sentences for each distributive coordination and pass those to two state-of-the-art Open IE systems, Open IE 4.2 and ClausIE. We call them CALMIE(O) and CALMIE(C) respectively. We evaluate on two different datasets and find that CALMIE(O) and CALMIE(C) always outperforms Open IE 4.2 and ClausIE respectively. CALMIE(O) particularly achieves 1.8x the yield with a 5 pt precision gain against Open IE 4.2 on a random sample of 100 conjunctive sentences from ClueWeb. Finally, we compare CALM with Ficler's system. On the subset of cases where a conjunction coordinates more than two conjuncts, our methods significantly outperform Ficler's, on an Open IE evaluation. We release our implementations of CALM, sentence splitter and CALMIE(O)[1] for further research.

2 Related Work

Open Information Extraction: Mausam (2016) surveys the progress in Open IE systems and its downstream applications (e.g., (Christensen et al., 2014; Stanovsky et al., 2015)). Existing Open IE systems are based on manually written patterns (Etzioni et al., 2011; de Sá Mesquita et al., 2013; Angeli et al., 2015), machine-learned patterns over bootstrapped training data (Mausam et al., 2012), sentence restructuring and decomposition (Bast and Haussmann, 2013; Schmidek and Barbosa, 2014), tree kernels (Xu et al., 2013) and simple inference (Bast and Haussmann, 2014). Some systems focus on specific kinds of extractions, e.g., noun-mediated (Pal and Mausam, 2016), numerical (Saha et al., 2017) and nested (Bhutani et al., 2016). We compare our methods to Open IE 4.2 (a combination of SRL-based IE (Christensen et al., 2011) and Relnoun (Pal and Mausam, 2016)) and ClausIE (Corro and Gemulla, 2013) – these outperformed others in a recent large-scale evaluation (Stanovsky and Dagan, 2016). ClausIE can also split some conjunctive clauses, thereby obtaining a higher yield.

There hasn't been any system with a specific focus on conjunctive sentences, which are recently found to be a reason for significant missed recall (Saha et al., 2017). While some of the missed recall is because of conjunctions appearing in arguments of extractions (extraction #6 in Table 1), which can be separated by reducing the argument spans (Stanovsky et al., 2016b), there are also important parts of the sentence which do not produce any extraction from either Open IE 4.2 or ClausIE (extraction #9, #10, #11 in

---

[1] CALMIE(O) is integrated into OpenIE 5.0, the latest software of OpenIE. It is publicly available at https://github.com/dair-iitd/OpenIE-standalone.
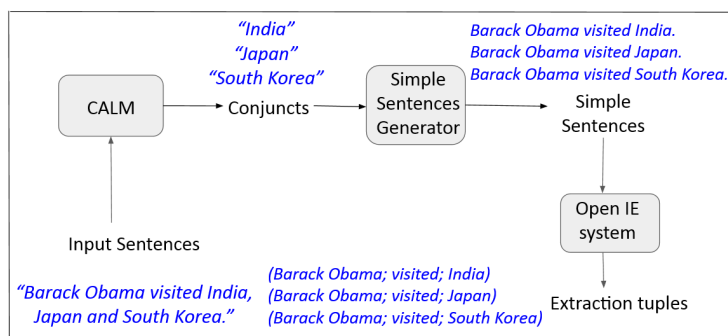
Figure 1: Flow diagram of CALMIE

Table 1), thereby motivating the need to split conjunctive sentences.

Conjunct Boundary Detection: Although co-ordination disambiguation has attracted attention of researchers over the years, it still remains one of the hardest problems to solve. Prior work has used two main principles for resolving ambiguities – (1) coordinated conjuncts have similar syntactic structures (Hogan, 2007; Shimbo and Hara, 2007; Hara et al., 2009; Hanamoto et al., 2012) and (2) replacing a full coordinated phrase with just one conjunct produces coherent (simple) sentences (Ficler and Goldberg, 2016).

The state-of-the-art coordination analyzer is by Ficler and Goldberg (2016), which uses LSTM-based components for operationalizing both these principles. It is a machine-learned model, which requires explicit annotation of coordination phrases for training (which isn't available in original Penn TreeBank). Importantly, it only outputs spans for the two conjuncts on either side of the conjunctive word and ignores any other conjuncts coordinated by the same word. Sentences often have a long list of comma separated conjuncts and not separating all of them would mean a substantial performance loss for end-tasks like Open IE.

In contrast, our approach eschews the first principle, as we find that it is not true often enough to be helpful. Our ranking of conjunct spans is based on the second principle (coherence of simple sentences). However, the search space is additionally restricted by various linguistic constraints for both single and nested coordination cases. Our system operates on top of Clear dependency parses (as opposed to constituency parses for Ficler's) and does not need any task-specific training data.

Sentence Simplification: CALM-based sentence splitting can be seen as a form of sentence simplification. Existing works on sentence simplification (Zhu et al., 2010; Vickrey and Koller, 2008; Vanderwende et al., 2007; Miwa et al., 2010) operate on top of syntactic parses, assuming them to be correct. CALM, on the other hand, corrects typical errors made by the parser to output corrected conjunction spans. These spans should naturally improve any sentence simplification task as well.

3 CALMIE

Figure 1 illustrates various steps of CALMIE. First, CALM identifies specific conjuncts to split the sentence into multiple simple sentences. Then an Open IE system acts on simple sentences to generate extractions. In this section we first focus on our key technical contribution, the coordination analyzer named CALM and conclude with a discussion on the generation of simple sentences.

3.1 CALM

CALM's goal is to output all *conjunct lists* from a sentence. For e.g., for sentence of Table 1, it should likely output five lists: (1) ⟨*an American investor'*,*'co-founder of Microsoft'*⟩, (2) ⟨*stepped down...2000', 'remained as...Craig Mundie'*⟩, (3) ⟨*'remained as chairman', 'created...Mundie'*⟩, (4) ⟨*'created...himself', 'transferred his...Mundie'*⟩, (5) ⟨*'Ray Ozzie','Craig Mundie'*⟩.[2]

---

[2]The conjunct lists #2, #3, #4 can change depending on the nesting level of the corresponding conjunctions.
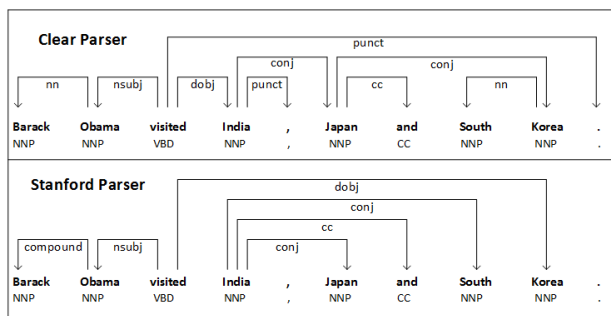
Figure 2: Clear and Stanford parses of a sample sentence

| 1. Generate the dependency parse of the sentence. |
| 2. Find one of the conjunct heads through 'cc' edge. |
| 3. Find all other conjunct heads through 'conj' edges. |
| 4. Expand the conjunct heads to generate the conjuncts. |

Table 3: Algorithm for Parser Baseline

We describe CALM's algorithm in two subsections. First, we describe CALM for a sentence that has only one coordinating conjunction (e.g., sentence of Figure 1). The case of multiple conjunctions, which may be nested (as in Table 1), is discussed in the latter subsection.

3.1.1 Single Coordinating Conjunction

We create a rule-based baseline to convert a dependency parse into a conjunct list (pseudo-code in Table 3). It finds various conjunct heads via *cc* and *conj* edges and expands the heads to generate conjunct spans. In the last step of the algorithm, while expanding on the conjunct heads, we don't expand on commas, the corresponding conjunction ('*and*', '*but*', etc) or any of the other conjunct heads.

We implement the baseline over Stanford[3] and Clear[4] parsers. Figure 2 shows that these dependency parsers have slightly different styles of notating conjuncts – Clear connects conjunct heads serially, whereas in Stanford parses one central conjunct heads connects to all others. Preliminary experiments reveal that Stanford parser makes many more mistakes than Clear. So we choose Clear parser for further algorithm design.

Analysis of Clear Parser-based Baseline Algorithm: Clear Parser identifies the conjunct heads correctly in most cases, but often makes mistakes in conjunct spans. We analyze a sample of 100 conjunctive sentences from BNC[5]. We find that 32 of them have correct parses with correctly identified conjuncts. Most correctly identified conjuncts are noun phrases (NP) while most incorrect ones involve verb phrases (VP). For e.g., in the sentence *"It also helps draw out toxins and excess oils."*, the NP conjuncts '*toxins*' and '*excess oils*' are identified correctly.

Our analysis also reveals that almost all wrong conjunct boundaries happen at the first and the last conjunct in the list, where the start of the first conjunct or the end of the last conjunct are identified incorrectly. Of the 68 incorrect parses, 57 of them have the first and last conjunct longer than necessary, suggesting that we can focus on shortening the first and last conjuncts. Further, out of these 57 sentences, 23 have a common NP subject distributed over multiple VPs and are wrongly represented in the parse. For e.g., in the sentence *"Angels danced in the air and settled reverently into their alcoves."*, the NP '*Angels*' incorrectly comes in the subtree of the VP '*danced in the air*', leading to the generation of two conjuncts - '*Angels danced in the air*' and '*settled reverently into their alcoves*'. Most of the remaining 34 sentences also contain a phrase that is distributed over two VPs, but appears in the subtree of only one. For e.g., in the sentence *"He rejoices at the fact that they started off with smalltown views , and began thinking globally."*, the phrase *"that they"* appear in the subtree of *"started"*. Note that the conjuncts again are VPs. Finally, incorrect conjunct boundaries almost always result in the generation of ungrammatical simple sentences.

We design CALM on the basis of these observations – it shortens the first and last conjunct spans in a way that the resulting simple sentences are coherent, as evaluated by a language model.

Language-Model Based Algorithm: CALM needs to find the best start of the first conjunct and best end

---

[3]nlp.stanford.edu/software/stanford-dependencies.shtml
[4]https://github.com/clir/clearnlp
[5]http://nclt.computing.dcu.ie/~jfoster/resources/bnc1000.html

2291

	Example 1	Example 2
Sentence	She gasped as he reached out and clasped her shoulders.	Still dazed, the man eventually got himself home and called police.
Parser Baseline (Clear Parser)	1. She gasped as he reached out. 2. She gasped clasped her shoulders.	1. Still dazed, the man eventually got himself home. 2. called police.
+ Language Model	1. She gasped as he reached out. 2. She gasped **as he** clasped her shoulders.	1. Still dazed, the man eventually got himself home. 2. called police.
+ Constraints	1. She gasped as he reached out. 2. She gasped **as he** clasped her shoulders.	1. Still dazed, the man eventually got himself home. 2. **Still dazed, the man eventually** called police.

Table 4: Comparison of sentences generated using different algorithms on two conjunctive sentences.

of the last conjunct. It first generates candidates by successively shortening conjuncts from the respective ends, until a conjunct is shortened to a single word. For e.g., for the first conjunct in *"Angels..reverently"* sentence, the candidates are *'Angels danced in the air'*, *'danced in the air'*, *'in the air'*, *'the air'* and *'air'*.

For each candidate, it constructs simple sentences by replacing the complete coordination structure with each of its conjuncts. It scores the coherence of each simple sentence via LMScore, a language model based score, described below. It picks the best span based on maximizing the total LMScore (over all simple sentences).

To score the coherence of a simple sentence, we could simply use its language model probability – product of probabilities of each word given the entire sentence so far. Most language models approximate this via an n-gram probability for a *fixed* context window of length $n-1$, instead of taking the *entire* sentence so far.

However, using the language model score is ineffective for two reasons. First, we are comparing simple sentences of varying lengths, and language model scores will usually score shorter sentences higher (product of fewer probabilities). A possible correction is to take the $|s|^{th}$ root of the language model score, with $|s|$ being the sentence length. But, this isn't enough for a second, subtle reason.

Consider the incorrect span *'the air'* for the first conjunct, which results in the simple sentence *"Angels danced in settled reverently into their alcoves"*. Notice that the n-gram probability scores will typically assign high probabilities for the parts: *'Angels danced in'* and *'settled reverently into their alcoves'*, since they are bona fide parts of the original sentence. Only the part around the boundary (*'...in settled...'*) would hurt the grammaticality. However, taking the product over the *whole* simple sentence has the tendency that the probabilities around the intersection could get overpowered by high scores from the initial and ending parts of the sentence, yielding an overall high score for an ungrammatical sentence.

To correct for this, CALM only multiplies probabilities starting from the intersection point and upto $n-1$ words (and takes $n-1$th root). All omitted probabilities are simply the product of original sentence fragments and thus are unhelpful in disambiguating the boundary. E.g., for $n=4$ our method LMScore computes the following product:

LMScore(*"Angels danced in settled...alcoves."*)

= [P(*'settled'* | *'Angels danced in'*)

× P(*'reverently'* | *'dance in settled'*)

× P(*'into'* | *'in settled reverently'*)]$^{1/3}$

In case k ($k < n-1$) probabilities are multiplied (for e.g., if a conjunct is too short), then CALM takes kth root of the product.

Use of Linguistic Constraints: Although LMScore corrects many of the wrongly identified conjuncts, we observe some limitations too. In sentences with proper nouns, the n-gram probabilities are low and significantly decrease the overall score of the sentence. Sentences with multi-word named entities also should not be split. For e.g., in the sentence *"Donald Trump gave a speech and flew back to U.S.."*, the simple sentence *"Donald flew back to U.S.."* is grammatical yet undesirable – named entity *'Donald Trump'* should not be split. We further see some obvious grammatical errors in the generated sentences, with two linguistically incoherent adjacent verbs or a preposition *'to'* preceding a past tense verb in the generated sentences.

To correct such errors, CALM picks the candidate that has the highest LMScore without violating any of the following linguistic constraints:

"*[(Jeff Bezos, an American [(electrical engineer) and ([(technology) and (retail)] entrepreneur)], founded [(Amazon.com) and (Blue Origin)]) and (his diversified business interests include [(books), (aerospace) and (newspapers)])].*"

[[(1-18) , (20-29)]]

[[(6-7) , (9-12)]]   [[(15-15) , (17-18)]]   [[(25-25) , (27-27) , (29-29)]]
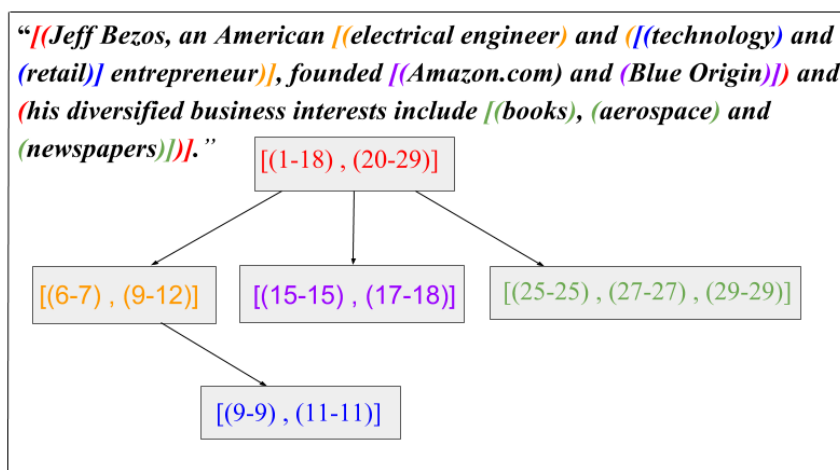
[[(9-9) , (11-11)]]

Figure 3: Hierarchical conjunct tree representation of a sentence with multiple conjunctions

1. Each simple sentence must have a subject. CALM looks for *nsubj* and *nsubjpass* dependency labels in the parse tree.
2. Named Entities (as identified by Stanford NER system[6]) should not be split.
3. If two verbs are adjacent, the first must be a light verb[7]. It handles ungrammatical sentences containing two adjacent incoherent verbs. For e.g., *"lives listening (to music)"* is nonsensical, but *"likes listening (to music)"* is not since *'like'* is a light verb.
4. A simple sentence should not have two consecutive occurrences of the same word.
5. Verb categories *VBD*, *VBZ* and *VBP* must precede a set of predefined POS tags[8]. English rarely allows a preposition, determiner or other such POS tags before the past/participle forms of a verb. However, language models do not always devalue these sentences. This constraint helps eliminate candidates that lead to such simple sentences.

3.1.2 Multiple Coordinating Conjunctions

When a sentence has multiple coordinating conjunctions, CALM identifies all coordination structures, i.e., the conjunct lists associated with each conjunction. Following English grammar, two coordination structures may have nothing in common (disjoint) or one conjunct list may be contained within the span of one of the conjuncts of another list (nested). For example, in sentence of Figure 3, the conjunct lists marked in yellow ({*'electrical engineer', 'technology and retail entrepreneur'*}) and green ({*'books', 'aerospace', 'newspapers'* }) are disjoint, while the one marked in blue ({*'technology', 'retail'*}) is nested within the second conjunct of yellow conjunct list.

Partial intersections, where one conjunct list is only partially contained in a conjunct of another list, are ungrammatical. CALM uses this knowledge as a search space constraint to jointly disambiguate all coordination structures in a sentence. We name this the *multiple conjunction constraint*. To operationalize this, we define a novel representation, *hierarchical coordination tree* (HCTree), for expressing the compositional containment between coordination structures.

An HCTree is a tree with each node representing a conjunct list for a single coordination structure, stored as a sequence of interval offsets. An edge from node A to B represents that B is fully contained within one of the conjuncts of A (nested case). Figure 3 illustrates an HCTree: *'electrical engineer'* is stored as interval 6-7, as it comprises sixth and seventh tokens in the sentence.

The number of legal HCTrees for a multiple conjunction sentence could be huge and a complete search over all tree structures and spans will be prohibitive. However, we reduce the search space by not performing a full search, as Clear parser usually returns a structurally accurate analysis, but, as before, makes errors in exact boundaries.

Use of Multiple Conjunction Constraint: CALM constructs an initial HCTree by converting the Clear parse into conjunct lists (using parser baseline), adding a node per conjunct list, and adding edges as per

---

[6]https://nlp.stanford.edu/software/CRF-NER.shtml
[7]We use the list of light verbs released by (Jain and Mausam, 2016) at https://github.com/dair-iitd/kglr.
[8]CC, CD, EX, NN, NNS, NNP, NNPS, PDT, PRP, RB, RBR, RBS, WDT, WP

2293

containments in the parse. This provides us with a good structure for enforcing the multiple conjunction constraint. CALM keeps the HCTree structure constant, and re-evaluates the exact conjunct boundaries for each node in the tree.

In addition to providing a structure over all conjunct lists, an HCTree also imposes a natural order in which we could correct span errors. Typically, smaller conjuncts are easier to correct than longer ones. So, CALM makes a bottom up pass greedily correcting spans for the conjunct list in each node. As in previous subsection, a conjunct is only shortened. Moreover only start of first conjunct and end of last conjunct in a list are re-evaluated using the single-conjunction version of CALM.

Notice that shortening of a child conjunct doesn't hurt the consistency of an HCTree, since if a conjunct list was contained in a parent conjunct, then the shortened conjunct list is also contained in the same parent conjunct. However, the boundaries of child conjunct list impose an additional constraint on the candidate spans of the parent conjunct list – while shortening the parent conjunct, all child conjuncts must continue to be contained in it. Thus, when applying CALM at the parent node, this additional search constraint is imposed to maintain a valid HCTree.

3.2 Simple Sentences for Open IE

Once the spans of conjunct lists have been determined, the next task is to generate the simple sentences. Our Open IE system, CALMIE generates them in a top-down order of the HCTree. It starts with the original sentence. Now, at each level of the tree, it collects all the conjunct lists and generates all possible sentences out of the sentences generated from the previous level. Since at a particular level of the tree all conjunct lists are disjoint, generating simple sentences requires simply identifying parts of the sentence that do not belong to any conjunct list and concatenating them in order with a conjunct for each coordination structure. Note that it avoids generation of duplicate sentences by identifying which conjunct lists at a particular level are part of which simple sentences from the previous level and finally splitting only those. After processing the conjunct lists at the last level of the HCTree, we get all the simple sentences.

For Open IE, we wish to only produce those simple sentences, whose truth can be inferred from the original sentence. CALMIE doesn't split conjuncts coordinated by *'or'*, *'nor'*, and paired conjunctions like *'either-or'* and *'neither-nor'*. For e.g. splitting *"Adam's nationality is French or German."* will be inaccurate.

One common class of non-distributive coordination that cannot be split contains prepositions like *'between'* in connecting the conjuncts. For e.g. the sentence *"The world cup final was played between Germany and Argentina."* should not be split. Other examples aggregate information across conjuncts as in the sentence, *"The average of 3 and 5 is 4."* Thus, we create a list of triggers (*'between'*, *'among'*, *'total'*, *'sum'*, *'average'*, *'each other'*, ...) using Thesaurus expansion of seed words which indicate non-distributive coordination . If the arguments of Open IE extractions on the original sentence contain any of the triggers, CALMIE does not split these conjuncts.

4 Experiments

We perform two sets of experiments. Section 4.1 evaluates CALM for coordination analysis task, and Section 4.2 demonstrates the performance increase, when using CALMIE for Open IE.

CALM's implementation uses the Berkeley Language Model,[9] which is based on the Google n-gram corpus. It uses Stupid Back-off smoothing (Brants et al., 2007) for infrequent n-grams. CALMIE runs Open IE 4.2 and ClausIE over the simple sentences generated in the previous section.

4.1 Experiments on Coordination Analysis

Previous work has given credit to a system only when both boundaries of a conjunct match exactly (Ficler and Goldberg, 2016). However, this is not ideal for downstream tasks like Open IE. Consider a sentence with polysyndetic coordination: *"Obama visited India and Japan and South Korea."*. Here, two analyses are equally good, depending upon whether we consider first conjunction as top-level or second, leading

---

[9]https://code.google.com/archive/p/berkeleylm/

	Parser Baseline (Stanford Parser)			Parser Baseline (Clear Parser)			+ Language Model			+ Constraints		
	SC	MC	SC+MC	SC	MC	SC+MC	SC	MC	SC+MC	SC	MC	SC+MC
Precision	83.93	69.20	79.30	**94.69**	86.78	92.14	94.33	87.85	**92.24**	94.22	**88.00**	92.21
Recall	80.91	80.78	80.86	90.22	78.34	86.39	91.36	82.75	88.58	**92.97**	**83.23**	**89.83**
F-score	82.39	74.75	80.07	92.40	82.34	89.17	92.82	85.22	90.37	**93.59**	**85.55**	**91.00**

Table 5: Results for simple sentence evaluation on British News Corpus. CALM obtains about 2 pt F-score improvement over Clear parser baseline. SC: Sentences with one conjunction, MC: Sentences with multiple conjunctions.

	(Ficler and Goldberg, 2016)	CALM
Precision	72.81	**75.12**
Recall	**72.61**	70.64
F1	72.7	**72.81**

Table 6: Results for conjunct boundary detection on Penn Treebank. CALM is competitive with state of the art.

to first conjunct being *'India'* or *'India and Japan'*, respectively. Such artifacts commonly happen when there are multiple conjunctions in a sentence, such as in Table 1.

In response, we evaluate a coordination analysis by generating the simple sentences and comparing them against the gold set of simple sentences. We split all coordinations (whether they are distributive or not) for this evaluation. For each conjunctive sentence, we compare its set of system-generated simple sentences with a gold set by first finding the best one-to-one mapping between the simple sentences in the two sets. Then for each mapping, precision is computed as the number of overlapping words upon the number of words in the predicted sentence. Recall is the number of overlapping words upon the number of words in the gold sentence. Notice that for the sentence above, all (correct) alternative analyses will yield the same simple sentences, obtaining perfect precision and recall scores.

We run our first set of experiments on all conjunctive sentences from British News Corpus test set.[10] It contains the gold parses for all sentences, which are used to generate the gold simple sentences. BNC testset has 577 conjunctive sentences – 391 with one conjunction and 186 with multiple conjunctions.

Table 5 compares the performances of all our algorithms on the whole BNC testset, and also individually on single and multiple conjunction sentences. We find that due to better conjunct heads identification, Clear parser does a much better job than Stanford parser in identifying conjunct boundaries. We get a point of F-score improvement by using language model over Clear parser baseline. Incorporating linguistic constraints yields another two-thirds of a point. More importantly, CALM obtains nearly 3.2 pt gain for sentences with multiple conjunctions; this highlights the value of our HCTree representation. Overall improvement of CALM over Clear baseline for the whole BNC is statistically significant using paired t-test with $p = 0.015$.

For sentences with a single conjunction, there is a slight drop in the final precision as compared to the parser baseline algorithm. Since we are only shortening the conjuncts, an already incorrect analysis can end up having even lesser number of common words with the gold sentence, reducing the precision. While the final precision is comparable, the recall improves significantly. This can be explained by examples in Table 4. E.g., in the second example, parser baseline has perfect precision, but low recall, due to missing words in the second simple sentence (*"called police"*). CALM corrects this by adding words in the sentence, increasing the recall to 1.0.

Comparison with Ficler's System: We also directly compare against Ficler's system on *their* dataset (Penn TreeBank). Recall that their evaluation metric is exact match of conjunct boundaries but only for the two conjuncts closest to the conjunction. We use their exact metric (Table 6) and find that CALM has slightly different characteristics – it has a higher precision and a lower recall; but, in aggregate it produces a similar F-score. Given that CALM is not trained directly (except for underlying parsers), we find it creditable that it can match performance of the state of the art system that was specifically trained on this data for this task. Unfortunately, their code isn't available for us to compare performance on BNC.

---

[10]http://nclt.computing.dcu.ie/~jfoster/resources/bnc1000.html

	ClueWeb12				News+Wikipedia			
	[C]	[Cm[C]]	[O4]	[Cm[O]]	[C]	[Cm[C]]	[O4]	[Cm[O]]
Precision	62.50	**64.50**	70.04	**74.80**	67.17	**68.12**	79.12	**81.2**
Yield	267	**381**	199	**349**	204	**325**	172	**315**

Table 7: Open IE comparison on two datasets. [C]: ClausIE, [O4]: Open IE 4.2, [Cm[O]]: CALMIE(O), [Cm[C]]: CALMIE(C)

	Two Conjuncts		More than Two Conjuncts	
	[FG]	[Cm[O]]	[FG]	[Cm[O]]
Precision	**72.71**	72.35	74.50	**74.78**
Yield	323	**330**	346	**445**

Table 8: Open IE comparison on Penn TreeBank. [FG]: Ficler+Open IE 4.2, [Cm[O]]: CALMIE(O).

Error Analysis: CALM sometimes misses conjunctions in sentences due to the inaccuracy of parsers (absence of a 'cc' edge). A more common problem is that of missing contexts in sentences while splitting. E.g., from the sentence *"Two years ago we were carrying huge inventories and that was the big culprit ."*, CALM generates two simple sentences: *"Two years ago we were carrying huge inventories."* and *"that was the big culprit."*. Although the sentences are grammatically correct, we miss the phrase *"Two years ago"* in the second sentence. While this is a missing prefix problem, CALM often tends to miss important phrases in the suffix as well. E.g., for the sentence *"We want to see Nelson Mandela and all our comrades out of the prison."*, it misses the suffix phrase *"out of the prison"*. We believe fixing such problems will require better understanding of the semantics of the sentence.

4.2   Experiments on Open IE

We now evaluate the improvement in performance on the final task of Open IE using conjunctive sentences from three different datasets. We randomly sample 100 conjunctive sentences each from ClueWeb12 (CW)[11] and Open IE benchmarking dataset (NW) of Newswire and Wikipedia sentences (Stanovsky and Dagan, 2016). We consider a sentence conjunctive if its parse has a *cc* edge. Note that we could not use the whole of NW dataset since its gold set of extractions does not split conjunctions in arguments.

We also test on Penn Treebank (PTB) testset used for coordination evaluation in (Ficler and Goldberg, 2016). We report two sets of numbers from the dataset – for a random sample of 100 sentences with two conjuncts per conjunction, and all of 95 conjunctive sentences with more than 2 conjuncts per conjunction.

For CW and NW, we compare CALMIE(C) (CALM generated simple sentences passed through ClausIE) and CALMIE(O) (CALM generated simple sentences passed through Open IE 4.2) against two state-of-the-art Open IE systems, ClausIE and Open IE 4.2. These Open IE systems outperform others in a recent large-scale evaluation. Moreover, ClausIE also splits some conjunctive clauses, making it an especially suitable system for this comparison. As there is no automated way to check the correctness of an extraction, two annotators with NLP experience annotate the extractions for correctness. Each annotator annotated about 6000 extractions in total. We obtain an inter-annotator agreement of 97.2% across all the datasets and report the results on the subset where both agree. Table 7 lists the precision and yield on these test sets. Note that yield is equal to the number of correct extractions and is proportional to recall. It is normally used in Open IE where recall denominator is hard to compute.

On CW, CALMIE(C) and CALMIE(O) obtain 1.4x and 1.8x yields compared to ClausIE and Open IE 4.2 respectively. The respective precision gains are 2 and 5 points. On NW, CALMIE(C) and CALMIE(O) achieve 1.6x and 1.8x better yields with marginal precision gains compared to ClausIE and Open IE 4.2 respectively. All the improvements are statistically significant using paired t-test at $p < 10^{-4}$. We find that both Open IE 4.2 and ClausIE are unable to separate out extractions if the conjunctions are in the arguments. This largely accounts for CALMIE's massive yield improvement. Overall, we see substantial improvement in extractions of both Open IE 4.2 and ClausIE after the application of CALM.

Finally, we split the sentences in PTB sample into simple sentences using CALM and Ficler. Both sets of simple sentences are then fed to Open IE 4.2 for generating extractions, i.e., the pipeline after

---

[11] http://www.lemurproject.org/clueweb12.php

coordination analysis is the same. Table 8 shows that on sentences with two conjuncts per conjunction, the two sets of extractions are comparable. However, on 95 sentences with more than two conjuncts per conjunction, CALMIE(O) achieves 1.3x better yield than Ficler. This is due to Ficler's inability to separate out all the conjuncts.

Error Analysis: The single major challenge in Open IE over conjunctive sentences is in figuring out when not to split. For example, in *"Japan's domestic sales of cars, trucks and buses in October rose by 18%."*, no obvious trigger is present that can indicate non-distributive coordination. Another common class is organization names with *'and'* as part of their names. For e.g., in sentence *"The Perch and Dolphin fields moved their headquarters."*, *'Perch and Dolphin'* should not be split. Use of LEX system could help with better NER tagging (Downey et al., 2007). Finally, non-context free constructions need to be handled differently while splitting as in, *"Germany and Argentina beat Brazil and Netherlands in the semis respectively."* We believe that this remains one of the key future directions for improving the precision of our system.

5 Conclusions and Discussion

Coordination disambiguity is regarded as one of the hardest problems in NLP (Ficler and Goldberg, 2016). We develop CALM, a coordination analyzer that corrects the errors of Clear parser by using three insights: (1) splitting into coherent simple sentences, as judged by a modified language model score, (2) linguistically inspired constraints to obtain better conjunct spans, and (3) a novel hierarchical coordination tree data structure that enforces consistency among conjunct lists for multiple coordinations, leading to performance gain.

We split distributive coordinations to obtain simple sentences for running downstream Open IE. Our Open IE system, CALMIE, obtains enormously higher yields (upto 1.8 times) and comparable or better precisions for conjunctive sentences against state of the art Open IE systems. Empirical evaluation across multiple datasets demonstrate than CALM should likely improve any Open IE system. We also believe that CALM has value outside of Open IE, such as for Closed IE and sentence simplification.

CALM's insights are general, but its implementation is specific to Clear parser, which is used in Open IE 4.2. Minor modifications may be needed when correcting coordination errors in other parsers. In the future, we plan to use CALM to improve Clear parser itself.

A strength (and also weakness) of our work is that it is not an ML system. While we test CALM on several domains, it is conceivable that for some domains or genres, it does not perform as well. Embedding our ideas in the context of a learning system may obtain even better in-domain performance. One approach is to use CALM output as (hard or soft) constraint at inference time, similar to how CCMs use constraints while learning parameters. We also note that our main idea of splitting into simple sentences, may not always work, e.g., in the case of ellipsis and gaping. Such sentences are infrequent in our datasets; extending to these is another direction for the future.

We release Open IE 5.0 which integrates CALMIE(O) into the latest software of OpenIE. We also release the annotated datasets used for evaluating CALMIE(O) and the implementations of CALM and the sentence splitter which can be used for splitting conjunctive sentences in various downstream NLP tasks. All of these are publicly available for further research at `https://github.com/dair-iitd/OpenIE-standalone`.

Acknowledgements

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354.

Hannah Bast and Elmar Haussmann. 2013. Open information extraction via contextual sentence decomposition. In *2013 IEEE Seventh International Conference on Semantic Computing, Irvine, CA, USA, September 16-18, 2013*, pages 154–159.

Hannah Bast and Elmar Haussmann. 2014. More informative open information extraction via simple inference. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 585–590.

Nikita Bhutani, H. V. Jagadish, and Dragomir R. Radev. 2016. Nested propositions in open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 55–64.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz Josef Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 858–867.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011), June 26-29, 2011, Banff, Alberta, Canada*, pages 113–120.

Janara Christensen, Stephen Soderland, Gagan Bansal, and Mausam. 2014. Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 902–912.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 355–366.

Filipe de Sá Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 447–457.

Doug Downey, Matthew Broadhead, and Oren Etzioni. 2007. Locating complex named entities in web text. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2733–2739.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 3–10.

Jessica Ficler and Yoav Goldberg. 2016. A neural network for coordination boundary prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 23–32.

Atsushi Hanamoto, Takuya Matsuzaki, and Jun'ichi Tsujii. 2012. Coordination structure analysis using dual decomposition. In *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012*, pages 430–438.

Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. 2009. Coordinate structure analysis with global structural constraints and alignment-based local features. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 967–975.

Deirdre Hogan. 2007. Coordinate noun phrase disambiguation in a generative parsing model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, June.

Prachi Jain and Mausam. 2016. Knowledge-guided linguistic rewrites for inference rule verification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 86–92.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 523–534.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4074–4077.

Makoto Miwa, Rune Saetre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd international conference on computational linguistics*, pages 788–796. Association for Computational Linguistics.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction, AKBC@NAACL-HLT 2016, San Diego, CA, USA, June 17, 2016*, pages 35–39.

Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for numerical open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 317–323.

Jordan Schmidek and Denilson Barbosa. 2014. Improving open relation extraction via sentence re-structuring. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pages 3720–3723.

Masashi Shimbo and Kazuo Hara. 2007. A discriminative learning model for coordinate conjunctions. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 610–619.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2300–2305.

Gabriel Stanovsky, Ido Dagan, and Mausam. 2015. Open IE as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 303–308.

Gabriel Stanovsky, Meni Adler, and Ido Dagan. 2016a. Specifying and annotating reduced argument span via qa-srl. In *Proceedings of the 54rd Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

Gabriel Stanovsky, Ido Dagan, and Meni Adler. 2016b. Specifying and annotating reduced argument span via QA-SRL. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. *Proceedings of ACL-08: HLT*, pages 344–352.

Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. 2013. Open information extraction with tree kernels. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 868–877.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1353–1361. Association for Computational Linguistics.