Generating Elliptic Coordination

Claire Gardent
CNRS, LORIA, UMR 7503
Vandoeuvre-lès-Nancy, F-54500, France
claire.gardent@loria.fr

Shashi Narayan
Université de Lorraine, LORIA, UMR 7503
Villers-lès-Nancy, F-54600, France
shashi.narayan@loria.fr

Abstract

In this paper, we focus on the task of generating elliptic sentences. We extract from the data provided by the Surface Realisation (SR) Task (Belz et al., 2011) 2398 input whose corresponding output sentence contain an ellipsis. We show that 9% of the data contains an ellipsis and that both coverage and BLEU score markedly decrease for elliptic input (from 82.3% coverage for non-elliptic sentences to 65.3% for elliptic sentences and from 0.60 BLEU score to 0.47). We argue that elided material should be represented using phonetically empty nodes and we introduce a set of rewrite rules which permits adding these empty categories to the SR data. Finally, we evaluate an existing surface realiser on the resulting dataset. We show that, after rewriting, the generator achieves a coverage of 76% and a BLEU score of 0.74 on the elliptical data.

1 Introduction

To a large extent, previous work on generating ellipsis has assumed a semantically fully specified input (Shaw, 1998; Harbusch and Kempen, 2009; Theune et al., 2006). Given such input, elliptic sentences are then generated by first producing full sentences and second, deleting from these sentences substrings that were identified to obey deletion constraints.

In contrast, recent work on generation often assumes input where repeated material has already been elided. This includes work on sentence compression which regenerates sentences from surface dependency trees derived from parsing the initial text (Filippova and Strube, 2008); Surface realisation approaches which have produced results for regenerating from the Penn Treebank (Langkilde-Geary, 2002; Callaway, 2003; Zhong and Stent, 2005; Cahill and Van Genabith, 2006; White and Rajkumar, 2009); and more recently, the Surface Realisation (SR) Task (Belz et al., 2011) which has proposed dependency trees and graphs derived from the Penn Treebank (PTB) as a common ground input representation for testing and comparing existing surface realisers. In all these approaches, repeated material is omitted from the representation that is input to surface realisation.

As shown in the literature, modelling the interface between the empty phonology and the syntactic structure of ellipses is a difficult task. For parsing, Sarkar and Joshi (1996), Banik (2004) and Seddah (2008) propose either to modify the derivation process of Tree Adjoining Grammar or to introduce elementary trees anchored with empty category in a synchronous TAG to accommodate elliptic coordinations. In HPSG (Head-Driven Phrase Structure Grammar), Levy and Pollard (2001) introduce a neutralisation mechanism to account for unlike constituent coordination ; in LFG (Lexical Functional Grammar), Dalrymple and Kaplan (2000) employ set values to model coordination; in CCG (Combinatory Categorial Grammar, (Steedman, 1996)), it is the non standard notion of constituency assumed by the approach which permits accounting for coordinated structures; finally, in TLCG (Type-Logical Categorial Grammar), gapping is treated as like-category constituent coordinations (Kubota and Levine, 2012).

In this paper, we focus on how surface realisation handles elliptical sentences given an input where repeated material is omitted. We extract from the SR data 2398 input whose corresponding output sentence contain an ellipsis. Based on previous work on how to annotate and to represent ellipsis, we argue that elided material should be represented using phonetically empty nodes (Section 3) and we introduce a set of rewrite rules which permits adding these empty categories to

Proceedings of the 14th European Workshop on Natural Language Generation, pages 40–50,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

the SR data (Section 4). We then evaluate our surface realiser (Narayan and Gardent, 2012b) on the resulting dataset (Section 5) and we show that, on this data, the generator achieves a coverage of 76% and a BLEU score, for the generated sentences, of 0.74. Section 6 discusses related work on generating elliptic coordination. Section 7 concludes.

2 Elliptic Sentences

Elliptic coordination involves a wide range of phenomena including in particular non-constituent coordination (1, NCC) i.e., cases where sequences of constituents are coordinated; gapping (2, G) i.e., cases where the verb and possibly some additional material is elided; shared subjects (3, SS) and right node raising (4, RNR) i.e., cases where a right most constituent is shared by two or more clauses[1].

(1) [It rose]$_i$ 4.8 % in June 1998 and ϵ_i 4.7% in June 1999. NCC

(2) Sumitomo bank [donated]$_i$ $\$500,000$, Tokyo prefecture ϵ_i $\$15,000$ and the city of Osaka ϵ_i $\$10,000$. *Gapping*

(3) [the state agency 's figures]$_i$ ϵ_i confirm previous estimates and ϵ_i leave the index at 178.9 . *Shared Subject*

(4) He commissions ϵ_i and splendidly interprets ϵ_i [fearsome contemporary scores]$_i$. *RNR*

We refer to the non elliptic clause as the *source* and to the elliptic clause as the *target*. In the source, the brackets indicate the element shared with the target while in the target, the ϵ_i sign indicate the elided material with co-indexing indicating the antecedent/ellipsis relation. In gapping clauses, we refer to the constituents in the gapped clause, as *remnants*.

3 Representing and Annotating Elided Material

We now briefly review how elided material is represented in the literature.

Linguistic Approaches. While Sag (1976), Williams (1977), Kehler (2002), Merchant (2001) and van Craenenbroeck (2010) have argued for a structural approach i.e., one which posits syntactic structure for the elided material, Keenan (1971), Hardt (1993), Dalrymple et al. (1991), Ginzburg and Sag (2000) and Culicover and Jackendoff (2005) all defend a non structural approach. Although no consensus has yet been reached on these questions, many of these approaches do postulate an abstract syntax for ellipsis. That is they posit that elided material licenses the introduction of phonetically empty categories in the syntax or at some more abstract level (e.g., the logical form of generative linguistics).

Treebanks. Similarly, in computational linguistics, the treebanks used to train and evaluate parsers propose different means of representing ellipsis.

For phrase structure syntax, the Penn Treebank Bracketing Guidelines extensively describe how to annotate coordination and missing material in English (Bies et al., 1995). For shared complements (e.g., shared subject and right node raising constructions), these guidelines state that the elided material licenses the introduction of an empty *RNR* category co-indexed with the shared complement (cf. Figure 1) while gapping constructions are handled by labelling the gapping remnants (i.e., the constituents present in the gapping clause) with the index of their parallel element in the source (cf. Figure 2).

```
(S
 (VP (VB Do)(VP (VB avoid)
   (S (VP (VPG puncturing(NP *RNR*-5))
   (CC or)
   (VP (VBG cutting)(PP (IN into)
       (NP *RNR*-5)))
    (NP-5 meats)))))
```

Figure 1: Penn Treebank annotation for Right Node Raising "*Do avoid puncturing ϵ_i or cutting into ϵ_i [meats]$_i$.*"

```
(S
  (S (NP-SBJ-10 Mary)
     (VP (VBZ likes) (NP-11 potatoes)))
  (CC and)
  (S (NP-SBJ=10 Bill)
     (, ,) (NP=11 ostriches)))
```

Figure 2: Penn Treebank annotation for gapping "*Mary [likes]$_i$ potatoes and Bill ϵ_i ostriches.*"

In dependency treebanks, headless elliptic constructs such as gapping additionally raise the is-

[1] Other types of elliptic coordination include sluicing and Verb-Phrase ellipsis. These will not be discussed here because they can be handled by the generator by having the appropriate categories in the grammar and the lexicon e.g., in a Tree Adjoining Grammar, an auxiliary anchoring a verb phrase for VP ellipsis and question words anchoring a sentence for sluicing.

sue of how to represent the daughters of an empty head. Three main types of approaches have been proposed. In dependency treebanks for German (Daum et al., 2004; Hajič et al., 2009) and in the Czech treebank (Čmejrek et al., 2004; Hajič et al., 2009), one of the dependents of the headless phrase is declared to be the head. This is a rather undesirable solution because it hides the fact that there the clause lacks a head. In contrast, the Hungarian dependency treebank (Vincze et al., 2010) explicitly represents the elided elements in the trees by introducing phonetically empty elements that serve as attachment points to other tokens. This is the cleanest solution from a linguistic point of view. Similarly, Seeker and Kuhn (2012) present a conversion of the German Tiger treebank which introduces empty nodes for verb ellipses if a phrase normally headed by a verb is lacking a head. They compare the performance of two statistical dependency parsers on the canonical version and the CoNLL 2009 Shared Task data and show that the converted dependency treebank they propose yields better parsing results than the treebank not containing empty heads.

In sum, while some linguists have argued for an approach where ellipsis has no syntactic representation, many have provided strong empirical evidence for positing empty nodes as place-holders for elliptic material. Similarly, in devising treebanks, computational linguists have oscillated between representations with and without empty categories. In the following section, we present the way in which elided material is represented in the SR data; we show that it underspecifies the sentences to be generated; and we propose to modify the SR representations by making the relationship between ellipsis and antecedent explicit using phonetically empty categories and co-indexing.

4 Rewriting the SR Data

The SR Task 2011 made available two types of data for surface realisers to be tested on: shallow dependency trees and deep dependency graphs. Here we focus on the shallow dependency trees i.e., on syntactic structures.

The input data provided by the SR Task were obtained from the Penn Treebank. They were derived indirectly from the LTH Constituent-to-Dependency Conversion Tool for Penn-style Treebanks (Pennconverter, (Johansson and Nugues, 2007)) by post-processing the CoNLL data to remove word order, inflections etc. It consists of a set of unordered labelled syntactic dependency trees whose nodes are labelled with word forms, part of speech categories, partial morphosyntactic information such as tense and number and, in some cases, a sense tag identifier. The edges are labelled with the syntactic labels provided by the Pennconverter. All words (including punctuation) of the original sentence are represented by a node in the tree. Figures 3, 4, 5 and 6 show (simplified) input trees from the SR data.

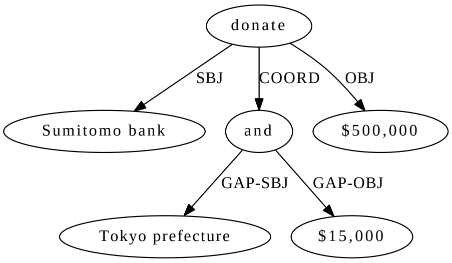

Figure 3: Gapping in the SR data. "*Sumitomo bank [donated]$_i$* $500,000$ *and Tokyo prefecture* ϵ_i $15,000$."

In the SR data, the representation of ellipsis adopted in the Penn Treebank is preserved modulo some important differences regarding co-indexing.

Gapping is represented as in the PTB by labelling the remnants with a marker indicating the source element parallel to each remnant. However while in the PTB, this parallelism is made explicit by co-indexing (the source element is marked with an index i and its parallel target element with the marker $= i$), in the SR data this parallelism is approximated using functions. For instance, if the remnant is parallel to the source subject, it will be labelled GAP-SBJ (cf. Figure 3).

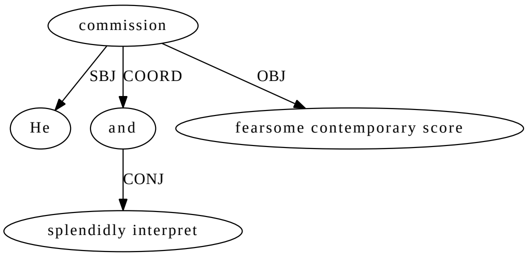

Figure 4: Subject Sharing and RNR in the SR data. "*[He]$_j$ ϵ_j commissions ϵ_i and ϵ_j splendidly interprets ϵ_i [fearsome contemporary scores]$_i$.*"

For right-node raising and shared subjects, the coindexation present in the PTB is dropped in the SR data. As a result, the SR representation under-

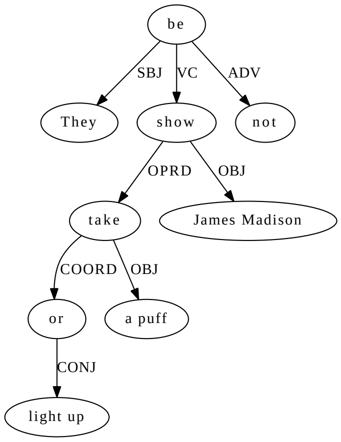

Figure 5: Non shared Object "*They aren't showing James Madison taking a puff or lighting up*"

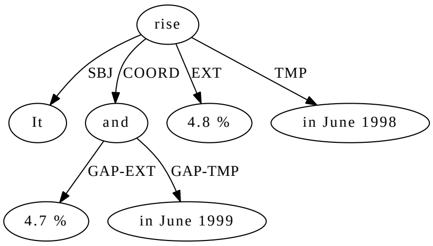

Figure 6: NCC in the SR data. "*It rose 4.8 % in June 1998 and 4.7% in June 1999.*"

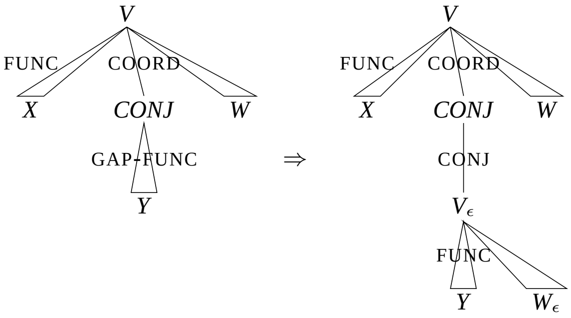

Figure 7: Gapping and Non Constituent Coordination structures rewriting (*V*: a verb, *CONJ*: a conjunctive coordination, *X*, *Y* and *W* three sets of dependents). The antecedent verb (*V*) and the source material without counterpart in the gapping clause (*W*) are copied over to the gapping clause and marked as phonetically empty.

specifies the relation between the object and the coordinated verbs in RNR constructions: the object could be shared as in *He commissions ϵ_i and splendidly interprets ϵ_i [fearsome contemporary scores]$_i$.* (Figure 4) or not as in *They aren't showing James Madison taking a puff or lighting up* (Figure 5). In both cases, the representation is the same i.e., the shared object (*fearsome contemporary scores*) and the unshared object (*a puff*) are both attached to the first verb.

Finally, NCC structures are handled in the same way as gapping by having the gapping remnants labelled with a GAP prefixed function (e.g., GAP-SBJ) indicating which element in the source the gapping remnant is parallel to (cf. Figure 6).

Summing up, the SR representation schema underspecifies ellipsis in two ways. For gapping and non-constituent coordination, it describes parallelism between source and target elements rather than specifying the syntax of the elided material. For subject sharing and right node raising, it fails to explicitly specify argument sharing.

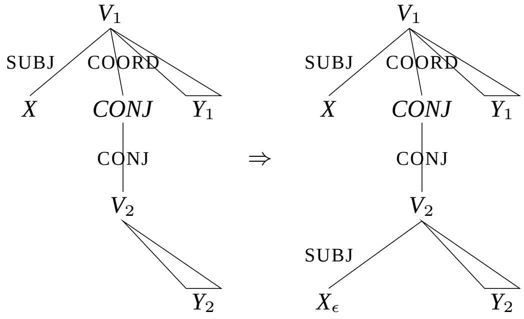

Figure 8: Subject sharing: the subject dependent is copied over to the target clause and marked as phonetically empty.

To resolve this underspecification, we rewrite the SR data using tree rewrite rules as follows.

In Gapping and NCC structures, we copy the source material that has no (GAP- marked) counterpart in the target clause to the target clause marking it to indicate a phonetically empty category (cf. Figure 7).

For Subject sharing, we copy the shared subject of the source clause in the target clause and mark it to be a phonetically empty category (cf. Figure 8).

For Right-Node-Raising, we unfold the ambiguity producing structures where arguments present in the source but not in the target are optionally copied over to the target (cf. Figure 9).

These rewrite rules are implemented efficiently

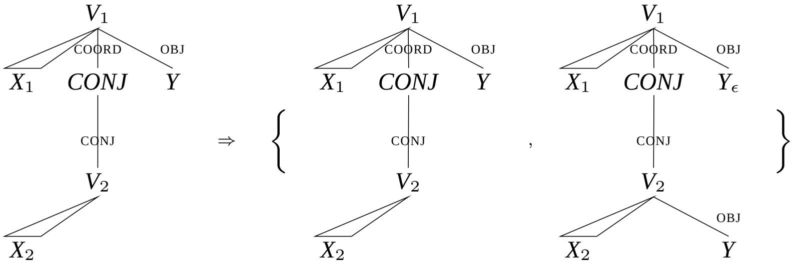

Figure 9: Right-Node-Raising: the object dependent is optionally copied over to the target clause and marked as phonetically empty in the source clause.

using GrGen, an efficient graph rewriting system (Geißet al., 2006).

5 Generating Elliptic Coordination

5.1 The Surface Realiser

To generate sentences from the SR data, we use our surface realiser (Narayan and Gardent, 2012b), a grammar-based generator based on a Feature-Based Lexicalised Tree Adjoining Grammar (FB-LTAG) for English. This generator first selects the elementary FB-LTAG trees associated in the lexicon with the lemmas and part of speech tags associated with each node in the input dependency tree. It then attempts to combine the selected trees bottom-up taking into account the structure of the input tree (only trees that are selected by nodes belonging to the same local input tree are tried for combination). A language model is used to implement a beam search letting through only the n most likely phrases at each bottom up combination step. In this experiment, we set n to 5. The generator thus outputs at most 5 sentences for each input.

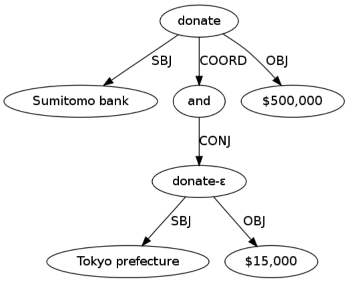

Figure 10: Gapping after rewriting "*Sumitomo bank [donated]*$_i$ $\$500,000$ *and Tokyo prefecture* ϵ_i $\$15,000$."

As mentioned in the introduction, most computational grammars have difficulty accounting for ellipses and FB-LTAG is no exception.

The difficulty stems from the fact that in elliptical sentences, there is meaning without sound. As a result, the usual form/meaning mappings that in non-elliptic sentences allow us to map sounds onto their corresponding meanings, break down. For instance, in the sentence *John eats apples and Mary pear*, the Subject-Verb-Object structure which can be used in English to express a binary relation is present in the source clause but not in the elided one. In practice, the syntax of elliptical sentences often leads to a duplication of the grammatical system, one system allowing for non elliptical sentences and the other for their elided counterpart.

For parsing with TAG, two main methods have been proposed for processing elliptical sentences. (Sarkar and Joshi, 1996) introduces an additional operation for combining TAG trees which yields derivation graphs rather than trees. (Seddah, 2008) uses Multi-Component TAG and proposes to associate each elementary verb tree with an elliptic tree with different pairs representing different types of ellipses.

We could use either of these approaches for generation. The first approach however has the drawback that it leads to a non standard notion of derivation (the derivation trees become derivation graphs). The second on the other hand, induces a proliferation of trees in the grammar and impacts efficiency.

Instead, we show that, given an input enriched with empty categories as proposed in the previous section, neither the grammar nor the tree combination operation need changing. Indeed, our FB-LTAG surface realiser directly supports the generation of elliptic sentences. It suffices to assume that an FB-LTAG elementary tree may be anchored by the empty string. Given an input node marked as phonetically empty, the generator will then select all FB-LTAG rules that are compatible with the lexical and the morpho-syntactic features labelling that node. Generation will then proceed as usual by composing the trees selected on the basis of the input using substitution and adjunction; and by retrieving from the generation forest those sentences whose phrase structure tree covers the input.

For instance, given the rewritten input shown in Figure 10, the TAG trees associated in the lexicon with *donate* will be selected; anchored with the empty string and combined with the TAG trees built for *Tokyo Prefecture* and *$15,000* thus yielding the derivation shown in Figure 11.

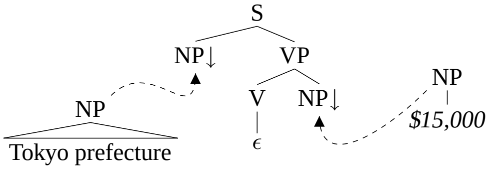

Figure 11: Derivation for "Tokyo prefecture ϵ $15,000"

5.2 The Data

We use both the SR test data (2398 sentences) and the SR training data (26604 sentences) to evaluate the performance of the surface realiser on elliptic coordination. Since the realiser we are using is not trained on this data (the grammar was written manually), this does not bias evaluation. Using the training data allows us to gather a larger set of elliptic sentences for evaluation while evaluating also on the test data allows comparison with other realisers.

Elliptic Coordination Data				
Elliptic Coordination		Pass	BLEU Scores	
			COV	ALL
RNR (384)	Before	66%	0.68	0.45
	After	81%	0.70	0.57
	Delta	+15	+0.02	+0.12
SS (1462)	Before	70%	0.74	0.52
	After	75%	0.75	0.56
	Delta	+5	+0.01	+0.04
SS + RNR (456)	Before	61%	0.71	0.43
	After	74%	0.73	0.54
	Delta	+13	+0.02	+0.11
Gapping (36)	Before	3%	0.53	0.01
	After	67%	0.74	0.49
	Delta	+64	+0.21	+0.48
NCC (60)	Before	5%	0.68	0.03
	After	73%	0.74	0.54
	Delta	+68	+0.06	+0.51
Total (2398)	Before	65%	0.72	0.47
	After	76%	0.74	0.56
	Delta	**+11**	+0.02	+0.09

Table 1: Generation results on elliptical data before and after input rewriting (SS: Shared Subject, NCC: Non Constituent Coordination, RNR: Right Node Raising). The number in brackets in the first column is the number of cases. Pass stands for the coverage of the generator. COV and ALL in BLEU scores column stand for BLEU scores for the covered and the total input data.

To focus on ellipses, we retrieve those sentences which were identified by our rewrite rules as potentially containing an elliptic coordination. In essence, these rewrite rules will identify all cases of non-constituent coordination and gapping (because these involve GAP-X dependencies with "X" a dependency relation and are therefore easily detected) and of shared-subjects (because the tree patterns used to detect are unambiguous i.e., only apply if there is indeed a shared subject). For RNR, as discussed in the previous section, the SR format is ambiguous and consequently, the rewrite rules might identify as object sharing cases where in fact the object is not shared. As noted by one of our reviewers, the false interpretation could be dropped out by consulting the Penn Treebank[2]. The approach would not generalise to other data however.

In total, we retrieve 2398 sentences[3] potentially containing an elliptic coordination from the SR training data. The number and distribution of these sentences in terms of ellipsis types are given in Table 1. From the test data, we retrieve an additional 182 elliptic sentences.

5.3 Evaluation

We ran the surface realiser on the SR input data both before and after rewriting elliptic coordinations; on the sentences estimated to contain ellipsis; on sentences devoid of ellipsis; and on all sentences. The results are shown in Table 2. They indicate coverage and BLEU score before and after rewriting. BLEU score is given both with respect to covered sentences (COV) i.e., the set of input for which generation succeeds; and for all sentences (ALL). We evaluate both with respect to the SR test data and with respect to the SR training

[2] The Penn Treebank makes the RNR interpretations explicit (refer to Figure 1).

[3] It is just a coincidence that the size of the SR test data and the number of extracted elliptic sentences are the same.

SR Data			Pass	BLEU Scores	
				COV	ALL
Test	+E (182)	Before	58%	0.59	0.34
		After	67%	0.59	0.40
		Delta	+9	+0.00	+0.06
	-E (2216)	Before	80%	0.59	0.47
		After	80%	0.59	0.48
		Delta	+0	+0.00	+0.01
	T (2398)	Before	78%	0.58	0.46
		After	79%	0.59	0.47
		Delta	+1	+0.01	+0.01
Training	+E (2398) (Table 1)	Before	65%	0.72	0.47
		After	76%	0.74	0.56
		Delta	**+11**	+0.02	+0.09
	-E (24206)	Before	82%	0.73	0.60
		After	82%	0.73	0.60
		Delta	+0	+0.00	+0.00
	T (26604)	Before	81%	0.72	0.58
		After	**82%**	**0.73**	0.60
		Delta	**+1**	+0.01	+0.02

Table 2: Generation results on SR test and SR training data before and after input rewriting (+E stands for elliptical data, -E for non elliptical data and T for total.)

data. We use the SR Task scripts for the computation of the BLEU score.

The impact of ellipsis on coverage and precision. Previous work on parsing showed that coordination was a main source of parsing failure (Collins, 1999). Similarly, ellipses is an important source of failure for the TAG generator. Ellipses are relatively frequent with 9% of the sentences in the training data containing an elliptic structure and performance markedly decreases in the presence of ellipsis. Thus, before rewriting, coverage decreases from 82.3% for non-elliptic sentences to 80.75% on all sentences (elliptic and non elliptic sentences) and to 65.3% on the set of elliptic sentences. Similarly, BLEU score decreases from 0.60 for non elliptical sentences to 0.58 for all sentences and to 0.47 for elliptic sentences. In sum, both coverage and BLEU score decrease as the number of elliptic input increases.

The impact of the input representation on coverage and precision. Recent work on treebank annotation has shown that the annotation schema adopted for coordination impacts parsing. In particular, Maier et al. (2012) propose revised annotation guidelines for coordinations in the Penn Treebank whose aim is to facilitate the detection of coordinations. And Dukes and Habash (2011) show that treebank annotations which include phonetically empty material for representing elided material allows for better parsing results.

Similarly, Table 2 shows that the way in which ellipsis is represented in the input data has a strong impact on generation. Thus rewriting the input data markedly extends coverage with an overall improvement of 11 points (from 65% to 76%) for elliptic sentences and of almost 1 point for all sentences.

As detailed in Table 1 though, there are important differences between the different types of elliptic constructs: coverage increases by 68 points for NCC and 64 points for gapping against only 15, 13 and 5 points for RNR, mixed RNR-Shared Subject and Shared Subject respectively. The reason for this is that sentences are generated for many input containing the latter types of constructions (RNR and Shared Subject) *even without rewriting*. In fact, generation succeeds on the non rewritten input for a majority of RNR (66% PASS), Shared Subject (70% PASS) and mixed RNR-Shared Subject (61% PASS) constructions whereas it fails for almost all cases of gapping (3% PASS) and of NCC (5% PASS). The reason for this difference is that, while the grammar cannot cope with headless constructions such as gapping and NCC constructions, it can often provide a derivation for shared subject sentences by using the finite verb form in the source sentence and the corresponding infinitival form in the target. Since the infinitival does not require a subject, the target sentence is generated. Similarly, RNR constructions can be generated when the verb in the source clause has both a transitive and an intransitive form: the transitive form is used to generate the source clause and the intransitive for the target clause. In short, many sentences containing a RNR or a shared subject construction can be generated without rewriting because the grammar overgenerates i.e., it produces sentences which are valid sentences of English but whose phrase structure tree is incorrect.

Nevertheless, as the results show, rewriting consistently helps increasing coverage even for RNR (+15 points), Shared Subject (+5 points) and mixed RNR-Shared Subject (+13 points) constructions because (i) not all verbs have both a transitive and an intransitive verb form and (ii) the input for the elliptic clause may require a finite form for the target verb (e.g., in sentences such as “*[they]$_i$ weren't fired but instead ϵ_i were neglected*” where the target clause includes an auxiliary requiring a past participial which in this context requires a subject).

Precision is measured using the BLEU score. For each input, we take the best score obtained within the 5 derivations[4] produced by the generator. Since the BLEU score reflects the degree to which a sentence generated by the system matches the corresponding Penn Treebank sentence, it is impacted not just by elliptic coordination but also by all linguistic constructions present in the sentence. Nonetheless, the results show that rewriting consistently improves the BLEU score with an overall increase of 0.09 points on the set of elliptic sentences. Moreover, the consistent improvement in terms of BLEU score for generated sentences (COV column) shows that rewriting simultaneously improves both coverage and precision that is, that for those sentences that are generated, rewriting consistently improves precision.

Analysing the remaining failure cases. To better assess the extent to which rewriting and the FB-LTAG generation system succeed in generating elliptic coordinations, we performed error mining on the elliptic data using our error miner described in (Narayan and Gardent, 2012a). This method permits highlighting the most likely sources of error given two datasets: a set of successful cases and a set of failure cases. In this case, the successful cases is the subset of rewritten input data for elliptic coordination cases for which generation succeeds . The failure cases is the subset for which generation fails. If elliptic coordination was still a major source of errors, input nodes or edges labelled with labels related to elliptic coordination (e.g., the COORD and the GAP-X dependency relations or the CONJ part of speech tag) would surface as most suspicious forms. In practice however, we found that the 5 top sources of errors highlighted by error mining all include the DEP relation, an unknown dependency relation used by the Pennconverter when it fails to assign a label to a dependency edge. In other words, most of the remaining elliptic cases for which generation fails, fails for reasons unrelated to ellipsis.

Comparison with other surface realisers There is no data available on the performance of surface realisers on elliptic input. However, the performance of the surface realiser can be compared with those participating in the shallow track of the SR challenge. On the SR training data, the TAG surface realiser has an average run time of 2.78 seconds per sentence (with an average of 20 words per sentence), a coverage of 82% and BLEU scores of 0.73 for covered and 0.60 for all. On the SR test data, the realiser achieves a coverage of 79% and BLEU scores of 0.59 for covered and 0.47 for all. In comparison, the statistical systems in the SR Tasks achieved 0.88, 0.85 and 0.67 BLEU score on the SR test set and the best symbolic system 0.25 (Belz et al., 2011).

6 Related work

Previous work on generating elliptic sentences has mostly focused on identifying material that could be elided and on defining procedures capable of producing input structures for surface realisation that support the generation of elliptic sentences.

Shaw (1998) developed a sentence planner which generates elliptic sentences in 3 steps. First, input data are grouped according to their similarities. Second, repeated elements are marked. Third, constraints are used to determine which occurrences of a marked element should be deleted. The approach is integrated in the PLANDoc system (McKeown et al., 1994) and shown to generate a wide range of elliptic constructs including RNR, VPE and NCC using FUF/SURGE (Elhadad, 1993), a realisation component based on Functional Unification Grammar.

Theune et al. (2006) describe how elliptic sentences are generated in a story generation system. The approach covers conjunction reduction, right node raising, gapping and stripping and uses dependency trees connected by rhetorical relations as input. Before these trees are mapped to sentences, repeated elements are deleted and their antecedent (the*source element*) is related by a SUBORROWED relation to their governor in the elliptic clause and a SUIDENTICAL relation to their governor in the antecedent clause. This is then interpreted by the surface realiser to mean that the repeated element should be realised in the source clause, elided in the target clause and that it licenses the same syntactic structure in both clauses.

Harbusch and Kempen (2009) have proposed a module called Elleipo which takes as input unreduced, non-elliptic, syntactic structures annotated with lexical identity and coreference relationships

[4] The language model used in the generator allows only 5 likely derivations (refer to section 5.1).

between words and word groups in the conjuncts; and returns as output structures annotated with elision marks indicating which elements can be elided and how (i.e., using which type of ellipsis). The focus is on developing a language independent module which can mediate between the unreduced input syntactic structures produced by a generator and syntactic structures that are enriched with elision marks rich enough to determine the range of possible elliptic and non elliptic output sentences.

In CCG, grammar rules (type-raising and composition) permit combining non constituents into a functor category which takes the shared element as argument; and gapping remnants into a clause taking as argument its left-hand coordinated source clause. White (2006) describes a chart based algorithm for generating with CCG and shows that it can efficiently realise NCC and gapping constructions.

Our proposal differs from these approaches in that it focuses on the surface realisation stage (assuming that the repeated elements have already been identified) and is tested on a large corpus of newspaper sentences rather than on hand-made document plans and relatively short sentences.

7 Conclusion

In this paper, we showed that elliptic structures are frequent and can impact the performance of a surface realiser. In line with linguistic theory and with some recent results on treebank annotation, we argued that the representation of ellipsis should involve empty categories and we provided a set of tree rewrite rules to modify the SR data accordingly. We then evaluated the performance of a TAG based surface realiser on 2398 elliptic input derived by the SR task from the Penn Treebank and showed that it achieved a coverage of 76% and a BLEU score of 0.74 on generated sentences. Our approach relies both on the fact that the grammar is lexicalised (each rule is associated with a word from the input) and on TAG extended domain of locality (which permits using a rule anchored with the empty string to reconstruct the missing syntax in the elided clause thereby making it grammatical).

We will release the 2398 input representations we gathered for evaluating the generation of elliptic coordination so as to make it possible for other surface realisers to be evaluated on their ability to generate ellipsis. In particular, its would be interesting to examine how other grammar based generators perform on this dataset such as White's CCG based generator (2006) (which eschews empty categories by adopting a more flexible notion of constituency) and Carroll and Oepen's HPSG based generator (2005) (whose domain of locality differs from that of TAG).

Acknowledgments

We would like to thank Anja Belz and Mike White for providing us with the evaluation data and the evaluation scripts. The research presented in this paper was partially supported by the European Fund for Regional Development within the framework of the INTERREG IV A Allegro Project.

References

Eva Banik. 2004. Semantics of VP coordination in LTAG. In *Proceedings of the 7th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+)*, volume 7, pages 118–125, Vancouver, Canada.

Anja Belz, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, Nancy, France.

Ann Bies, Mark Ferguson, Katz Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Marry Ann Marcinkiewicz, and Britta Schasberger. 1995. Bracketing guidelines for treebank II style penn treebank project. *University of Pennsylvania*.

Aoife Cahill and Josef Van Genabith. 2006. Robust pcfg-based generation using automatically acquired lfg approximations. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING) and the 44th annual meeting of the Association for Computational Linguistics (ACL)*, pages 1033–1040, Sydney, Australia.

Charles B Callaway. 2003. Evaluating coverage for large symbolic nlg grammars. In *Proceedings of the 18th International joint conference on Artificial Intelligence (IJCAI)*, volume 18, pages 811–816, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

John Carroll and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP)*, pages 165–176, Jeju Island, Korea. Springer.

M. Čmejrek, J. Hajič, and V. Kuboň. 2004. Prague czech-english dependency treebank: Syntactically annotated resources for machine translation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Peter W. Culicover and Ray Jackendoff. 2005. *Simpler Syntax*. Oxford University Press.

Mary Dalrymple and Ronald M. Kaplan. 2000. Feature indeterminacy and feature resolution. *Language*, pages 759–798.

Mary Dalrymple, Stuart M. Sheiber, and Fernando C. N. Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*.

Michael Daum, Kilian Foth, and Wolfgang Menzel. 2004. Automatic transformation of phrase treebanks to dependency trees. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal.

Kais Dukes and Nizar Habash. 2011. One-step statistical parsing of hybrid dependency-constituency syntactic representations. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 92–103, Dublin, Ireland. Association for Computational Linguistics.

Michael Elhadad. 1993. *Using argumentation to control lexical choice: a functional unification implementation*. Ph.D. thesis, Columbia University.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference (INLG)*, pages 25–32, Salt Fork, Ohio, USA. Association for Computational Linguistics.

Rubino Geiß, Gernot Veit Batz, Daniel Grund, Sebastian Hack, and Adam M. Szalkowski. 2006. Grgen: A fast spo-based graph rewriting tool. In *Proceedings of the 3rd International Conference on Graph Transformation*, pages 383–397. Springer. Natal, Brasil.

Jonathan Ginzburg and Ivan Sag. 2000. *Interrogative investigations*. CSLI Publications.

J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M.A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, et al. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning: Shared Task*, pages 1–18.

Karin Harbusch and Gerard Kempen. 2009. Generating clausal coordinate ellipsis multilingually: A uniform approach based on postediting. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 138–145, Athens, Greece. Association for Computational Linguistics.

Daniel Hardt. 1993. *Verb phrase ellipsis: Form, meaning and processing*. Ph.D. thesis, University of Pennsylvania.

Richert Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 105–112, Tartu, Estonia.

Edward Keenan. 1971. Names, quantifiers, and the sloppy identity problem. *Papers in Linguistics*, 4:211–232.

Andrew Kehler. 2002. *Coherence in discourse*. CSLI Publications.

Yusuke Kubota and Robert Levine. 2012. Gapping as like-category coordination. In *Proceedings of the 7th international conference on Logical Aspects of Computational Linguistics (LACL)*, pages 135–150, Nantes, France. Springer-Verlag.

Irene Langkilde-Geary. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the 12th International Natural Language Generation Workshop*, pages 17–24.

Roger Levy and Carl Pollard. 2001. Coordination and neutralization in HPSG. *Technology*, 3:5.

Wolfgang Maier, Erhard Hinrichs, Julia Krivanek, and Sandra Kübler. 2012. Annotating coordination in the Penn Treebank. In *Proceedings of the 6th Linguistic Annotation Workshop (LAW)*, pages 166–174, Jeju, Republic of Korea. Association for Computational Linguistics.

Kathleen McKeown, Karen Kukich, and James Shaw. 1994. Practical issues in automatic documentation generation. In *Proceedings of the fourth conference on Applied natural language processing (ANLC)*, pages 7–14, Stuttgart, Germany. Association for Computational Linguistics.

Jason Merchant. 2001. *The syntax of silence: Sluicing, islands, and the theory of ellipsis*. Oxford University Press.

Shashi Narayan and Claire Gardent. 2012a. Error mining with suspicion trees: Seeing the forest for the trees. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, Mumbai, India.

Shashi Narayan and Claire Gardent. 2012b. Structure-driven lexicalist generation. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, Mumbai, India.

Ivan Sag. 1976. *Deletion and logical form*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Anoop Sarkar and Arvind Joshi. 1996. Coordination in tree adjoining grammars: Formalization and implementation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 610–615, Copenhagen, Denmark. Association for Computational Linguistics.

Djamé Seddah. 2008. The use of mctag to process elliptic coordination. In *Proceedings of The Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 9)*, volume 1, page 2, Tübingen, Germany.

Wolfgang Seeker and Jonas Kuhn. 2012. Making ellipses explicit in dependency conversion for a german treebank. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.

James Shaw. 1998. Segregatory coordination and ellipsis in text generation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 1220–1226, Montreal, Quebec, Canada.

Mark Steedman. 1996. *Surface Structure and Interpretation*, volume 30. MIT press Cambridge, MA.

Mariët Theune, Feikje Hielkema, and Petra Hendriks. 2006. Performing aggregation and ellipsis using discourse structures. *Research on Language & Computation*, 4(4):353–375.

Jeoren van Craenenbroeck. 2010. *The syntax of ellipsis: Evidence from Dutch dialects*. Oxford University Press.

V. Vincze, D. Szauter, A. Almási, G. Móra, Z. Alexin, and J. Csirik. 2010. Hungarian dependency treebank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.

Michael White and Rajakrishnan Rajkumar. 2009. Perceptron reranking for ccg realization. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 410–419, Singapore. Association for Computational Linguistics.

Michael White. 2006. Efficient realization of coordinate structures in combinatory categorial grammar. *Research on Language & Computation*, 4(1):39–75.

Edwin Williams. 1977. Discourse and logical form. *Linguistic Inquiry*.

Huayan Zhong and Amanda Stent. 2005. Building surface realizers automatically from corpora. In *Proceedings of the Workshop on Using Corpora for Natural Language Generation (UCNLG)*, volume 5, pages 49–54.