Complex Lexico-Syntactic Reformulation of Sentences using Typed Dependency Representations

Advaith Siddharthan
Department of Computing Science
University of Aberdeen
advaith@abdn.ac.uk

Abstract

We present a framework for reformulating sentences by applying transfer rules on a typed dependency representation. We specify a list of operations that the framework needs to support and argue that typed dependency structures are currently the most suitable formalism for complex lexico-syntactic paraphrasing. We demonstrate our approach by reformulating sentences expressing the discourse relation of *causation* using four lexico-syntactic discourse markers – "cause" as a verb and as a noun, "because" as a conjunction and "because of" as a preposition.

1 Introduction

There are many reasons why a writer might want to choose one formulation of a discourse relation over another; for example, maintaining thread of discourse, avoiding shifts in focus and issues of salience and end weight. There are also reasons to use different formulations for different audiences; for example, to account for differences in reading skills and domain knowledge. In recent work, Siddharthan and Katsos (2010) demonstrated through psycholinguistic experiments that domain experts and lay readers show significant differences in which formulations of *causation* they find acceptable. They further showed that the most appropriate formulation depends both on the domain expertise of the user and the propositional content of the sentence, and that these preferences can be learnt in a supervised machine learning framework. That work, as does much of the related comprehension and literacy literature, used manually reformulated sentences. In this paper, we present an approach to automate such complex reformulation. We consider the four lexico-syntactic discourse markers for *causation* studied by Siddharthan and Katsos (2010); consider 1a.–d. below (from their corpus, but simplified to aid presentation):

(1) a. An incendiary device **caused** the explosion. [A-CAUSE-B]
b. The explosion occurred **because of** an incendiary device. [B-BECAUSEOF-A]
c. The explosion occurred **because** there was an incendiary device. [B-BECAUSE-A]
d. The **cause** of the explosion was an incendiary device. [CAUSEOF-B-A]

These differ in terms of the lexico-syntactic properties of the discourse marker (shown in bold font). Indeed the discourse markers here are verbs, prepositions, conjunctions and nouns. As a consequence, the propositional content is expressed either as a clause or a noun phrase ("*The explosion occurred*" vs "*the explosion*", etc.). Additionally, the order of presentation of propositional content can be varied to give four more lexico-syntactic paraphrases:

(1) e. The explosion **was caused by** an incendiary device. [B-CAUSEBY-A]
f. **Because of** an incendiary device, the explosion occurred. [BECAUSEOF-A-B]
g. **Because** there was an incendiary device, the explosion occurred. [BECAUSE-A-B]
h. An incendiary device was the **cause** of the explosion. [A-CAUSEOF-B]

It is clear that some formulations of a given propositional content can be more felicitous than others; for example, 1e. seems preferable to 1g. However, for different propositional content, other formulations might be more felicitous. While discourse level choices based on information ordering play a role in choosing a formulation, Siddharthan and Katsos (2010) demonstrate that some de-contextualised information orderings within a sentence are deemed unacceptable by some categories of readers. This has implications for text regeneration tasks that try to reformulate texts for different audiences; for instance, simplifying language for low reading ages or summarising technical writing for lay readers. In short, considerations of discourse coherence should not introduce sentence-level unacceptability in regenerated text.

We focus on causal relations for many reasons.

For the purpose of this paper, our main reason is that the 8 formulations selected are different information orderings of 4 different lexico-syntactic constructs. Thus, we explore a broad range of constructions and are confident that the framework we develop covers the range of operations required for text regeneration in general. Of less relevance to this paper, but equally important to our broad goals of reformulating technical writing for lay readers, causal relations are pervasive in science writing and are integral to how humans conceptualise the world. We have a particular interest in scientific writing – reformulating such texts for lay audiences is a highly relevant task today and many news agencies perform this service; e.g., Reuters Health summarises medical literature for lay audiences and BBC online has a Science/Nature section that reports on science. These services rely either on press releases by scientists and universities or on specialist scientific reporters, thus limiting coverage of a growing volume of scientific literature in a digital economy.

In Section 2, we relate our research to the existing linguistic and computational literature. Then in Section 3, we compare three different linguistic representations with respect to their suitability for lexico-syntactic reformulation. We found typed dependency structures to be the most promising and present an evaluation in Section 4.

2 Related Work

2.1 Discourse Connectives and Comprehension

Previous work has shown that when texts have been manually rewritten to make the language more accessible (L'Allier, 1980), or to make the content more transparent (Beck et al., 1991), students' reading comprehension shows significant improvements. An example of a revision choice that might be applied differentially depending on the literacy skills of the reader involves connectives such as *because*. Connectives that permit pre-posed adverbial clauses have been found to be difficult for third to fifth grade readers, even when the order of mention coincides with the causal (and temporal) order (Anderson and Davison, 1988); this experimental result is consistent with the observed order of emergence of connectives in children's narratives (Levy, 2003).

Thus the b) version of the following example would be preferred for children who can grasp causation, but who have not yet become comfortable with alternative clause orders (example from Anderson and Davison (1988), p. 35):

(2) a. Because Mexico allowed slavery, many Americans and their slaves moved to Mexico during that time.

b. Many Americans and their slaves moved to Mexico during that time, because Mexico allowed slavery.

Such studies show that comprehension can be improved by reformulating text for readers with low reading skills (Linderholm et al., 2000; Beck et al., 1991) and for readers with low levels of domain expertise (Noordman and Vonk, 1992). Further, specific information orderings were found to be facilitatory by Anderson and Davison (1988). All these studies suggest that the automatic lexico-syntactic reformulation of causation can benefit various categories of readers.

2.2 Connectives and Text (Re)Generation

Much of the work regarding (re)generation of text based on discourse connectives aims to simplify text in certain ways, to make it more accessible to particular classes of readers. The PSET project (Carroll et al., 1998) considered simplifying news reports for aphasics. The PSET project focused mainly on lexical simplification (replacing difficult words with easier ones), but there has been work on syntactic simplification and, in particular, the way syntactic rewrites interact with discourse structure and text cohesion (Siddharthan, 2006). These were restricted to string substitution and sentence splitting based on pattern matching over chunked text. Our work aims to extend these strands of research by allowing for more sophisticated insertion, deletion and substitution operations that can involve substantial reorganisation and modification of content within a sentence.

Elsewhere, there has been interest in *paraphrasing*, including the replacement of words (especially verbs) with their dictionary definitions (Kaji et al., 2002) and the replacement of idiomatic or otherwise troublesome expressions with simpler ones. The emphasis has been on automatically learning paraphrases from comparable or aligned corpora (Barzilay and Lee, 2003; Ibrahim et al., 2003). The text simplification and paraphrasing literature does not address paraphrasing that requires syntactic alterations such as those in Example 1 or the question of appropriateness of different formulations of a discourse relation.

Some natural language generation systems incorporate results from psycholinguistic studies to make principled choices between alternative formulations. For example, SkillSum (Williams and Reiter, 2008) and ICONOCLAST (Power et al.,

2003) are two contemporary generation systems that allow for specifying aspects of style such as choice of discourse marker, clause order, repetition and sentence and paragraph lengths in the form of constraints that can be optimised. However, to date, these systems do not consider syntactic reformulations of the type we are interested in. Our research is directly relevant to such generation systems as it can help such systems make decisions in a principled manner.

Williams et al. (2003) examined the impact of discourse level choices on readability in the domain of reporting the results of literacy assessment tests, using the results of the test to control both the content and the realisation of the generated report. Our research aims to facilitate the transfer of such user-driven generation research to text regeneration areas.

2.3 Sentence Compression

Sentence compression is a research area that aims to shorten sentences for the purpose of summarising the main content. There are similarities between our interest in reformulation and existing work in sentence compression. Sentence compression has usually been addressed in a generative framework, where transformation rules are learnt from parsed corpora of sentences aligned with manually compressed versions. The compression rules learnt are therefore tree-tree transformations (Knight and Marcu, 2000; Galley and McKeown, 2007; Riezler et al., 2003) of some variety. These approaches focus on *deletion* operations, mostly performed low down in the parse tree to remove modifiers. Further they make assumptions about isomorphism between the aligned tree, which means they cannot be readily applied to more complex reformulation operations such as *insertion* and *reordering* that are essential to perform reformulations such as those in Example 1. Cohn and Lapata (2009) provide an approach based on Synchronous Tree Substitution Grammar (STSG) that in principle can handle the range of reformulation operations. However, given their focus on sentence compression, they restricted themselves to local transformations near the bottom of the parse tree. In this paper, we explore whether this framework could prove useful to more involved reformulation tasks. Our experience (see Section 3.2) suggests that parse trees are the wrong representation for learning complex transformation rules and that dependency structures are more suited for complex lexico-syntactic reformulation.

3 Regeneration using Transfer Rules

We experimented with three representations – phrasal parse trees, typed dependencies and Minimal Recursion Semantics (MRS). In this section, we first describe our data, and then report our experience with performing text reformulation using these representations.

3.1 Data

We use the corpus described in Siddharthan and Katsos (2010). This corpus contains examples of complex lexico-syntactic reformulations such as those in Example 1a–f; each example consists of 8 formulations, 7 of which are manual reformulations. The corpus contains 144 such examples from three genres, giving 1152 sentences in total. The manual reformulation is formulaic and Example 1 is indicative of the process. To make a clause out of a noun phrase, either the copula or the verb "occur" is introduced, based on a subjective judgement of whether this is an event or a continuous phenomenon. Conversely, to create a noun phrase from a clause, a possessive and gerund are used; for example (from Siddharthan and Katsos (2010)):

(3) a. Irwin had triumphed because he was so good a man.
b. The cause of Irwin's having triumphed was his being so good a man.

The corpus contains equal numbers of sentences from three different genres: PubMed Abstracts[1] (technical writing from the Biomedical domain), and articles from the British National Corpus[2] tagged as World News or Natural Science (popular science writing in the mainstream media).

3.2 Reformulation using Phrasal Parse Trees

As described above, we have access to a corpus that contains aligned sentences for each pair of types (a type is a combination of a discourse marker and an information order; thus we have 8 types). In principle it should be easy to learn transfer rules between parse trees of aligned sentences. Figure 1 shows parse trees (using the RASP parser (Briscoe et al., 2006)) for the active and the passive voice with "cause" as a verb. A transfer rule is derived by aligning nodes between two parse trees so that the rule only contains the differences in structure between the trees. In the representation in Figure 1, the variable ??X0[NP] maps

[1] PubMed URL: http://www.ncbi.nlm.nih.gov/pubmed/

[2] The British National Corpus, version 3 (BNC XML Edition). 2007. http://www.natcorp.ox.ac.uk

The explosion was caused by an incendiary device.
(S
(NP (AT The) (NN1 explosion))
(VP (VBDZ be+ed)
(VP (VVN cause+ed)
(PP (II by)
(NP (AT1 an) (JJ incendiary) (NN1 device))))))

An incendiary device caused the explosion.
(S
(NP (AT1 An) (JJ incendiary) (NN1 device))
(VP (VVD cause+ed)
(NP (AT the) (NN1 explosion))))

Derived Rule:
(S
(??X0[NP])
(VP (VBZ be+s)
(VP(VVN cause+ed) (PP(II by+) (??X1[NP])))))

↓

(S
(??X1[NP])
(VP (VVZ cause+s) (??X0[NP])))

Figure 1: Example of a transfer rule derived from two parse trees.

onto any node (subtree) with label NP. RASP performs a morphological analysis of words (shown as lemma+suffix in the figure). Thus such rules can be used to account for changes in morphology, as in example 3a.–b. above.

In practise however, the parse tree representation is too dependent on the grammar rules employed by the parser. For instance, the parse tree for the sentence:

The explosion was presumed to be caused by an incendiary device.
(S
(NP (AT The) (NN1 explosion))
(VP (VBDZ be+ed)
(VP (VVN presume+ed)
(VP (TO to)
(VP (VB0 be)
(VP (VVN cause+ed)
(PP (II by) (NP (AT1 an)
(JJ incendiary) (NN1 device)))))))))

looks very different and does not match the rule in figure 1. With longer sentences, further problems arise when similar strings are parsed differently in the two aligned sentences (for example, different PP attachment) – these lead to very complicated rules, often with more than 20 variables. We split our data into development/training (96 instances of passive to active) and test sets (48 instances of passive). Using the top parse for each sentence, we derived 92 rules, including the one shown in Figure 1. However, coverage of these rules over the test corpus was poor (less than 10% recall). By learning rules using the top 20 parses for each sentence rather than just the top parse, we could improve coverage to around 70%, but this involved the acquisition of over 4000 different rules – just to change voice. The situation was even worse for reformulations that change syntactic categories, such as "because" to "cause", and we obtained more than 20,000 rules that still gave us a coverage of only around 15% for the test set.

We concluded that this was not a sensible representation for general text reformulation. In other words, while substitution grammars for parse trees have been shown to be useful for sentence compression tasks (e.g., Cohn and Lapata (2009)), they are less useful for more complex lexico-syntactic reformulation tasks.

3.3 Reformulation using MRS

Another option is to use a bi-directional grammar and perform the transforms at a semantic level. We now briefly discuss the use of Minimal Recursion Semantics (MRS) as a representation for transfer rules. Consider a very short example for ease of illustration:

Tom ate because of his hunger.

This can be analysed by a deep grammar to give a compositional semantic representation which captures the information that is available from the syntax and inflectional morphology. We show this sentence below in the Minimal Recursion Semantics (MRS) (Copestake et al., 2005) representation, as produced by the English Resource Grammar (ERG[3]) (Flickinger, 2000), but considerably simplified for ease of exposition and to save space:

```
named(x5,Tom), _eat_v_1(e2,x5),
_because_of(e2,x11), poss(x11,x16),
pron(x16), _hunger_n(x11)
```

The main part of the MRS structure is a list of elementary predications (EPs), which may have predicates derived from lexemes (e.g., `_eat_v_1`; these are indicated by the leading underscore) or supplied by the grammar (e.g., `poss`). The ERG treats *because of* as a multiword expression and assigns it a semantics comparable to a preposition. Paraphrase rules map between semantic representations; for our application, a possible rule is the following:

```
_because_of(e,x), P(e,y) <->
    _cause_v_1(e10,x,y,l1), l1:P(e,y)
```

Here 'P' is to be understood as a general predicate. The left hand side of the rule will match the preposition-like 'because of' relation when it has an event as an argument, where the event is the

[3] Available at http://www.delph-in.net.

characteristic event of an underspecified verbal EP. The right hand side indicates that the 'because of' can be substituted by a verbal relation corresponding to *cause*, with the verbal EP being a scopal argument. This rule matches the MRS above and maps it to the following (with P=`_eat_v_1`):

```
named(x5,Tom), l1:_eat_v_1(e2,x5),
_cause_v_1(e10,x11,x5,l1), poss(x11,x16),
pron(x16), _hunger_n(x11), x5 aeq x16
```

This can be input to the realiser, giving:

His hunger caused Tom to eat.

Writing transfer rules is intuitive and easy in MRS. Further, the use of a bi-directional grammar for generation ensures that the generated sentence is grammatical. An infrastructure of writing paraphrase rules exists in this framework and semantic transfer has also been explored for machine translation (e.g., Copestake et al. (1995)).

The problem we encountered, however, is that bidirectional grammars such as the ERG fail to parse ill-formed input and will also fail to analyse some well-formed input because of limitations in coverage of unusual constructions. Although the DELPH-IN parsing technology allows for unknown words, missing lexical items can also cause parse failure and even more problems for generation. The ERG gives an acceptable parse 'out of the box' for only around 50-60% of sentences from scientific papers. Further, the generator can get slow and memory intensive for long sentences and many of our sentences are around 30 words long. Much of this processing effort during generation is redundant as the input sentence can be used to narrow down generation choices, but as of now, the infrastructure does not exist to support this. Thus, while using a bi-directional grammar and semantic transfer might indeed be the most intuitive approach to complex lexico-syntactic reformulation, it is not quite feasible yet.

3.4 Reformulation using Typed Dependencies

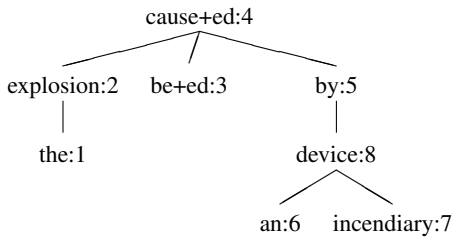
Having had mixed success with transforming phrasal parse trees and semantic representations, we turned our attention to typed dependency structures. We used the RASP toolkit (Briscoe et al., 2006) for finding grammatical relations (GRs) between words in the text. GRs are triplets consisting of a relation-type and arguments and also encode morphology (stem + suffix), word position (after colon) and part-of-speech (after underscore); GRs produced for the sentence:

The explosion was caused by an incendiary device.

are:

(|ncsubj| |cause+ed:4_VVN| |explosion:2_NN1| _)
(|aux| |cause+ed:4_VVN| |be+ed:3_VBDZ|)
(|passive| |cause+ed:4_VVN|)
(|iobj| |cause+ed:4_VVN| |by:5_II|)
(|dobj| |by:5_II| |device:8_NN1|)
(|det| |device:8_NN1| |an:6_AT1|)
(|ncmod| _ |device:8_NN1| |incendiary:7_JJ|)
(|det| |explosion:2_NN1| |the:1_AT|)

This representation shares aspects of phrasal parse trees and MRS. Note that the sets of dependencies (such as those above) represent a tree.[4] While phrase structure trees such as those in Section 3.2 represent the nesting of constituents with the actual words at the leaf nodes, dependency trees have words at every node:



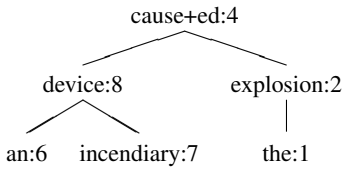
To generate from a dependency tree, we need to know the order in which to process nodes - in general tree traversal will be "inorder"; i.e, left subtrees will be processed before the root and right subtrees after. These are generation decisions that would usually be guided by the type of dependency and statistical preferences for word and phrase order. However, we can simply use the word positions (1–8) from the original sentence.

While typed dependencies share characteristics with parse trees, the flat structure represents dependencies between words, and we can write transformation rules for this representation in fairly compact form. For instance, a transformation rule to convert the above to active voice would require five deletions and two insertions:

1. Match and Delete:
   (a) (|passive| |??X0|)
   (b) (|iobj| |??X0| |??X1(by_II)|)
   (c) (|dobj| |??X1| |??X2|)
   (d) (|ncsubj| |??X0| |??X3| _)
   (e) (|aux| |??X0| |??X4|)
2. Insert:
   (a) (|ncsubj| |??X0| |??X2| _)
   (b) (|dobj| |??X0| |??X3|)

[4] In fact, the GR scheme is only 'almost' acyclic. There are a small number of (predictable) relations that introduce cycles; for instance, dependencies between the head of a relative clause and the verb in the relative clause are represented as both a clausal modifier relation (cmod head verb) and an object relation (obj verb head). To resolve this, we use a fixed set of rules to remove these cycles from the dependency graph and ensure a tree structure.

Thus far, the rule looks very similar to rules written for MRS: one list of predicates is replaced by another. Applying this transformation to the GR set above creates a new dependency tree:



However, unlike the case with MRS, where a statistical generator decides issues of morphology and ordering, we have to specify the consequences of the rule application for generation. Note that we can no longer rely on the original word order to determine the order in which to traverse the tree for generation. Thus our transformation rules, in addition to Deletion and Insertion operations, also need to provide rules for tree traversal order. These only need to be provided for nodes where the transform has reordered subtrees ("??X0", which instantiates to "cause+ed:4" in the trees). Our rule would thus include:

3. Traversal Order Specifications:
   (a) Node ??X0: [??X2, ??X0, ??X3]

This states that for node ??X0, the traversal order should be subtree ??X2 followed by current node ??X0 followed by subtree ??X3. Using this specification would allow us to traverse the tree using the original word order for nodes with no order specification, and the specified order where a specification exist. In the above instance, this would lead us to generate:

An incendiary device caused the explosion.

Our transfer rule is still incomplete and there is one further issue that needs to be addressed – operations to be performed on nodes rather than relations. There are two node-level operations that might be required for sentence reformulation:

1. Lexical substitution: In our example above, we still need to ensure number agreement for the verb "cause" (??X0). By changing voice, ??X0 now has to agree with ??X2 rather than ??X3. Further the tense of ??X0 was encoded in the auxiliary verb ??X4 that has been deleted from the GRs. We thus need the transfer rule to encode the lexical substitution required for node ??X0:

4. Lexical substitution:
   (a) Node ??X0: IF (??X4 is Present Tense) THEN { IF (??X2 is Plural) THEN {SET ??X0:SUFFIX ="“} ELSE {SET ??X0:SUFFIX ="s"} }

Other lexical substitutions are easier to specify; for instance to reformulate "*John ran because David shouted.*" as "*David's shouting caused John to run*", the following lexical substitution rule is required for node ??Xn representing "shout" that replaces its suffix "ed" with "ing":

Lexical substitution: Node ??Xn: Suffix="ing"

2. Node deletion: This is an operation that removes a node from the tree. Any subtrees are moved to the parent node. If a root node is deleted, one of the children adopts the rest. By default, the right-most child takes the rest as dependents, but we allow the rule to specify the new parent. In the above example, we want to remove the nodes ??X1("by") and ??X4 ("was") (note that deleting a relation does not necessarily remove a node – there might be other nodes connected to ??X1 or ??X4). We would like to move these to the node ??X0 ("cause"):

5. Node Deletion:
   (a) Node ??X1: Target=??X0
   (b) Node ??X4: Target=??X0

Node deletion is easily implemented using search and replace on sets of GRs. It is central to reformulations that alter syntactic categories of discourse markers; for instance, to reformulate "*The cause of X is Y*" as "*Y causes X*", we need to delete the verb "is" and move its dependents to the new verb "causes".

To summarise, we propose a framework for lexico-syntactic reformulation based on typed dependency structures and have discussed the form of a transformation. We now specify the structure of transfer rules and tree nodes more formally.

Specification for Transfer Rules

Our proposal is based on applying transfer rules to lists of grammatical relations (GRs). Our transfer rules take the form of five lists:

1. CONTEXT: Transform only proceeds if this list of GRs can be unified with the input GRs.
2. DELETE: List of GRs to delete from input.
3. INSERT: List of GRs to insert into input.
4. ORDERING: List of nodes with subtree order specified
5. NODE-OPERATIONS: List of lexical substitutions and deletion operations on nodes.

For the reformulations in this paper, the CONTEXT and DELETE lists are one and the same, but one can imagine reformulation tasks where extra context needs to be specified to determine whether reformulation is appropriate. The first three lists

correspond to the CONTEXT, INPUT and OUTPUT lists used to specify transform in the MRS framework. However, because we do not use a formal grammar for generation, we need two further lists that capture changes in morphology or constituent ordering. The list ORDERING is used to traverse the dependency tree constructed from the transformed GRs. Again, the lexical substitution lists are prescriptions for generation. We restrict our lexical substitutions to change of suffix and part of speech (for instance, "X is a *frequent* cause of Y" to "X *frequently* causes Y"), but in general this can be an arbitrary string substitution (for instance, "X and Y are *two* causes of Z" to "X and Y *both* cause Z").

In this paper, we have tried to do away with a generator altogether by encoding generation decisions within the transfer rule. A case can be made, particularly for the issue of agreement, for such issues to be handled by a generator. This would make the transfer rules simpler to write, and easier to learn automatically in a supervised setting.

Specification for Dependency Tree

Applying a transfer rule specified above results in a new set of GRs. To generate a sentence, we need to create a dependency tree from these GRs. As described earlier, a dependency tree needs to be traversed "inorder" to generate a sentence. This means that at each node, the order in which to visit the daughters and the current node needs to be specified. To enable this, we propose that each node in the tree have the following features:

1. VALUE: stem, suffix and part-of-speech of the word;
2. PARENT: parent node;
3. CHILDREN: list of daughters;
4. ORDER: list specifying order in which to visit children and current node.

The parent node is required for DELETE operations and to find the root of the tree (node with no parent). Further, if there is more than one node with no parent, the GRs do not form a tree and generation will result in multiple fragments.

The dependency tree is constructed using the following algorithm:

1. For each word in the list of GRs:
   (a) Create a Node and instantiate the VALUE field.
2. For each GR (relation word1 word2):
   (a) If GR is one that introduces a cycle, remove it from list, else add the node created for word2 to the CHILDREN list of node for word1 and set PARENT of word 2 to word1.
3. After Step 2, the tree is created. Now for each Node:
   (a) If an ORDERING specification is introduced for this node by the transformation rule, copy that list to the ORDER field, else add the daughter nodes to the ORDER list in increasing order of word position.

The reformulated sentence is generated by traversing the tree "inorder", outputting the word at each node visited (the stem, suffix and part-of-speech tag are fed to the RASP morphological generator, which returns the correct word).

4 Evaluating Transformation Rules

In this paper we have proposed a framework for complex lexico-syntactic reformulations. We want to evaluate our framework for (a) how easy it is to write transformation rules, (b) how many are required for intuitive lexico-syntactic reformulations and (c) how robust the transformation is to parsing errors. With this intended purpose, we evaluate hand-written transformation rules that have been developed looking at one third of the corpus (48 sentences) and tested on the remaining two thirds (96 sentences). We report results using:

- **Recall:** The proportion of sentences in the test set for which a transform was performed; i.e., (a) the DELETE pattern matched the input GRs and (b) there was exactly one root node in the transformed GRs resulting in exactly one sentence being output
- **Precision:** The proportion of transformed sentence that were accurate; i.e., grammatical with (a) correct verb agreement and inflexion and (b) modifiers/complements appearing in acceptable orders.

Note that we are merely evaluating the framework and not evaluating the utility of these transformations for text simplification – that would require an evaluation using test subjects drawn from our intended users. Table 1 provides some examples of accurate and inaccurate transformations.

The rule for converting passives to actives described in Section 3.4 already achieves a recall of 42% and precision of 83%. Writing 6 additional rules to handle reduced relative clauses (1a-b, Table 1) etc., we could boost recall to 71% with precision dropping marginally to 82%. We handcrafted rules to implement three other reformulations. These were selected based on results from the Siddharthan and Katsos (2010) study that suggested:

1. cause as a noun (either information ordering), passive voice, "because of" and "because a, b" formulations (versions b,d,e,f,g and h in Example 1, Section 1) are dispreferred by lay readers. Moreover, these are common constructs in scientific writing.
2. cause as a verb in active voice and "b because a" are the most preferred formulations for lay readers.

	Accurate Transformations
1a.	Apart from occasional problems of ensemble caused by the complex rhythms of the outer movements, the orchestra gave an animated and committed reading of the work. [B-CAUSEBY-A→A-CAUSE-B]
b.	Apart from occasional problems of ensemble the complex rhythms of the outer movements caused, the orchestra gave an animated and committed reading of the work.
2a.	Because of transvection, the expression of a gene can be sensitive to the proximity of a homolog. [BEC-OF-A-B→A-CAUSE-B]
b.	Transvection can cause the expression of a gene to be sensitive to the proximity of a homolog.
3a.	Because each myosin is expressed in Drosophila indirect flight muscle, in the absence of other myosin isoforms, this allows for muscle mechanical and whole organism locomotion assays. [BEC-A-B→B-BEC-A]
b.	In the absence of other myosin isoforms, this allows for muscle mechanical and whole organism locomotion assays because each myosin is expressed in Drosophila indirect flight muscle.
4a.	Almost certainly, however, the underlying cause of the war was the problem of Aquitaine. [CAUSEOF-B-A→A-CAUSE-B]
b.	Almost certainly, however, the underlying problem of Aquitaine caused the war.
	Inaccurate Transformation
5a.	Moreover, main road traffic has scarcely been slowed and concern should be caused by the rising number of cyclist casualties. [B-CAUSEBY-A→A-CAUSE-B]
b.	Moreover, the rising number of cyclist casualties should cause main road traffic has scarcely been slowed and concern.
6a.	Because of the risk of injury and the need to kill prey quickly, predators usually predate animals smaller than themselves. [BEC-OF-A-B→A-CAUSE-B]
b	The risk of injury and the need cause kill to prey quickly predators usually predate animals smaller than themselves.

Table 1: Examples of automatic reformulations (version a. is the original and b. the reformulation).

Handcrafted rules	**n**	**P**	**R**	**F**
B-CAUSEBY-A → A-CAUSE-B	7	.82 (1.00)	.71 (.75)	.76 (.86)
BEC-OF-A-B→ A-CAUSE-B	9	.75 (.92)	.70 (1.00)	.72 (.97)
BEC-A-B→ B-BEC-A	8	.85 (.92)	.83 (.87)	.84 (.89)
CAUSEOF → A-CAUSE-B	6	.97 (.90)	.78 (1.00)	.86 (.95)

Table 2: Number of Rules (n), Precision, Recall and F-Measure for lexico-syntactic reformulation using hand-crafted rules over GRs. Numbers in brackets are over the subset of the corpus that contains only the original sentences from PubMed and the BNC.

We summarise our results in Table 2. Most of the sentences in the corpus are manual reformulations and some of them are quite stilted. The numbers in brackets show performance over the smaller set of original sentences from PubMed and the BNC. These are more indicative of how the rules will perform on real data. Our results suggests that the framework we propose is adequate for a range of lexico-syntactic reformulations and a fairly small number of rules is required to capture a reformulation.

Loss of recall was usually from parsing error (either misparses, in which case our rules don't match the GRs, or partial parses, where a full tree can't be formed because of missing GRs).

Loss of precision was a more worrying issue as it often resulted in badly corrupted output. This was usually the result of either bad parser decisions regarding attachment or scope or just misparsing (e.g., wide scoping of "and" in 5a-b and parsing "prey" as a verb in 6a-b, Table 1). It might be possible to trade-off recall for improved precision by identifying sentences where ambiguity is a problem (by looking at multiple parses).

5 Conclusions and Future Work

In this paper we have reported our experience with using different linguistic formalisms as representations for applying transform rules to generate complex lexico-syntactic reformulations of sentences expressing the discourse relation of causation. We find typed dependency structures to be the most suited for this task and report that handcrafted transformation rules generalise well to sentences in an unseen test corpus. We believe that the framework we have described is adequate for a range of regeneration tasks focused on text simplification. While in this paper we focus on the discourse relation of causation, other discourse relations commonly used in scientific writing can also be realised using markers with different lexico-syntactic properties; for instance, *contrast* can be expressed using markers such as "while", "unlike", "but", "compared to", "in contrast to" and "the difference between". Our rules for voice conversion and information reordering for subordination are already general enough to be applied to non-causal constructs. We also plan to use our framework to explore sentence simplification and sentence shortening applications.

We would in the future like to learn transformations rules automatically from a corpus. Handcrafting can get tedious as there are 17 types of grammatical relations to take into account in the RASP scheme. Preliminary work by us in this regard suggests that augmenting a few hand-crafted rules with around a hundred automatically learnt rules can increase recall substantially. However, our learning framework as yet does not allow node transformations, and more work is required here.

Acknowledgements

This work was supported by the Economic and Social Research Council (Grant Number RES-000-22-3272). We would also like to thank Dan Flickinger and Ann Copestake for many discussions on the topic of paraphrase and for help with using the ERG.

References

R.C. Anderson and A. Davison. 1988. Conceptual and empirical bases of readibility formulas. In Alice Davison and G. M. Green, editors, *Linguistic Complexity and Text Comprehension: Readability Issues Reconsidered*. Lawrence Erlbaum Associates, Hillsdale, NJ.

R. Barzilay and L. Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003: Main Proceedings*, pages 16–23.

I.L. Beck, M.G. McKeown, G.M. Sinatra, and J.A. Loxterman. 1991. Revising social studies text from a text-processing perspective: Evidence of improved comprehensibility. *Reading Research Quarterly*, pages 251–276.

T. Briscoe, J. Carroll, and R. Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL*, volume 6.

J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait. 1998. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of AAAI98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10, Madison, Wisconsin.

T. Cohn and M. Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34(1):637–674.

A. Copestake, D. Flickinger, R. Malouf, S. Riehemann, and I. Sag. 1995. Translation using minimal recursion semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 15–32.

A. Copestake, D. Flickinger, I. Sag, and C. Pollard. 2005. Minimal recursion semantics: An introduction. *Research in Language and Computation*, 3:281–332.

D. Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

M. Galley and K. McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *HLT-NAACL 2007: Main Proceedings*, pages 180–187, Rochester, New York, April. Association for Computational Linguistics.

A. Ibrahim, B. Katz, and J. Lin. 2003. Extracting paraphrases from aligned corpora. In *Proceedings of The Second International Workshop on Paraphrasing*.

N. Kaji, D. Kawahara, S. Kurohash, and S. Sato. 2002. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 215–222, Philadelphia, USA.

K. Knight and D. Marcu. 2000. Statistics-based summarization — step one: Sentence compression. In *Proceeding of The American Association for Artificial Intelligence Conference (AAAI-2000)*, pages 703–710.

J.J. L'Allier. 1980. *An evaluation study of a computer-based lesson that adjusts reading level by monitoring on task reader characteristics*. Ph.D. thesis, University of Minnesota, Minneapolis, MN.

E.T. Levy. 2003. The roots of coherence in discourse. *Human Development*, pages 169–88.

T. Linderholm, M.G. Everson, P. van den Broek, M. Mischinski, A. Crittenden, and J. Samuels. 2000. Effects of Causal Text Revisions on More-and Less-Skilled Readers' Comprehension of Easy and Difficult Texts. *Cognition and Instruction*, 18(4):525–556.

L. G. M. Noordman and W. Vonk. 1992. Reader's knowledge and the control of inferences in reading. *Language and Cognitive Processes*, 7:373–391.

R. Power, D. Scott, and N. Bouayad-Agha. 2003. Generating texts with style. *Proceedings of the 4 thInternational Conference on Intelligent Texts Processing and Computational Linguistics*.

S. Riezler, T.H. King, R. Crouch, and A. Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In *HLT-NAACL 2003: Main Proceedings*, Edmonton, Canada.

A. Siddharthan and N. Katsos. 2010. Reformulating discourse connectives for non-expert readers. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, Los Angeles, CA.

A. Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

S. Williams and E. Reiter. 2008. Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, 14(04):495–525.

S. Williams, E. Reiter, and L. Osman. 2003. Experiments with discourse-level choices and readability. In *Proceedings of the European Natural Language Generation Workshop (ENLG), EACL'03*, pages 127–134, Budapest, Hungary.