

# Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules

**Advaith Siddharthan**

Computing Science  
University of Aberdeen

UK

advaith@abdn.ac.uk

**M. A. Angrosh**

Computing Science  
University of Aberdeen

UK

angroshmandya@abdn.ac.uk

## Abstract

We present an approach to text simplification based on synchronous dependency grammars. The higher level of abstraction afforded by dependency representations allows for a linguistically sound treatment of complex constructs requiring reordering and morphological change, such as conversion of passive voice to active. We present a synchronous grammar formalism in which it is easy to write rules by hand and also acquire them automatically from dependency parses of aligned English and Simple English sentences. The grammar formalism is optimised for monolingual translation in that it reuses ordering information from the source sentence where appropriate. We demonstrate the superiority of our approach over a leading contemporary system based on quasi-synchronous tree substitution grammars, both in terms of expressivity and performance.

## 1 Introduction

*Text simplification* is sometimes defined as the process of reducing the grammatical and lexical complexity of a text, while still retaining the original information content and meaning. The main goal of simplification is to make information more accessible to the large numbers of people with reduced literacy. The National Literacy Trust (<http://www.literacytrust.org.uk>) estimates that one in six adults in the UK have poor literacy skills. The situation is often worse in developing countries. Aluísio et al. (2008) report that 68% of Brazilians between 15 and 64 years who have studied up to 4 years only reach the rudimentary level of literacy, and even among those who have studied for 8 years, only a quarter can be considered fully literate. While there is a large

body of evidence that manual text simplification is an effective intervention (Anderson and Freebody, 1981; L’Allier, 1980; Beck et al., 1991; Anderson and Davison, 1988; Linderholm et al., 2000; Kamalski et al., 2008), there has till recently been little work on automatic simplification. The pace of research has picked up in recent years though, with many teams applying machine translation approaches to perform “monolingual translation” from English to simplified English. The goals of this paper are to (1) identify the limitations of recently published approaches to text simplification with regard to their coverage of linguistic constructs, (2) to describe an approach based on synchronous grammars operating on typed dependency representations that permits a more sophisticated handling of many linguistic constructs, and (3) to present a hybrid system that combines a small set of hand written grammar rules for purely syntactic constructs with a much larger set of automatically acquired rules for lexicalised constructs in one synchronous formalism.

We summarise work on text simplification in Section 2, before describing our method in Section 3 and presenting our results in Section 4.

## 2 Related work

There are two largely distinct bodies of work on automatic text simplification – those that use hand-crafted rules, and those that apply machine translation approaches.

### 2.1 Hand-crafted text simplification systems

The first body of work uses hand-crafted rules to perform syntactic simplification operations (e.g., splitting coordinated and subordinated clauses, and disembedding apposition and relative clauses). Some early systems (Chandrasekar et al., 1996; Siddharthan, 2002) used flat representations (chunked and part-of-speech tagged text). More commonly, text simplification systems use

hand crafted rules that apply to hierarchical representations, including constituency-based parses (Canning, 2002; Candido Jr et al., 2009; De Belder and Moens, 2010) and dependency parses (Bott et al., 2012; Siddharthan, 2010; Siddharthan, 2011). For languages without corpora of simplified texts, hand crafted systems are typically the only available alternative.

## 2.2 Text simplification as monolingual translation

Recent years have seen the increased application of machine translation approaches to text simplification, often referred to as “monolingual translation”, and driven by the new availability of corpora of simplified texts such as Simple English Wikipedia (SEW).

Wubben et al. (2012) and Coster and Kauchak (2011) apply Phrase Based Machine Translation (PBMT) to the task of text simplification. PBMT can only perform a small set of simplification operations, such as lexical substitution, deletion and simple paraphrase. They are not well suited for reordering or splitting operations. Specifically, the syntactic simplification operations that hand-crafted systems focus on are out of scope.

Zhu et al. (2010) in contrast present an approach based on syntax-based SMT (Yamada and Knight, 2001). Their translation model encodes probabilities for four specific rewrite operations on the parse trees of the input sentences: substitution, reordering, splitting, and deletion. Splitting is encoded as two probabilities: A segmentation table stores probabilities of sentence splitting at particular words (e.g., *which*). A completion table stores probabilities of the splitting word to be deleted from the translation, and for the governing phrase to be inserted to complete the sentence. This allows the translation model to handle constructs such as relative clauses and apposition.

Dras (1999) was the first to apply synchronous grammars to monolingual tasks. His approach is to map between two TAG grammars using a Generalised Synchronous TAG formalism, and to use Integer Programming to generate a text that satisfies the externally imposed constraints (such as length or readability) using minimal paraphrasing. Woodsend and Lapata (2011) further develop this line of research. Their model is based on quasi-synchronous grammar (Smith and Eisner, 2006) and integer linear programming. Quasi-

synchronous grammars, like the Generalised Synchronous TAGs of Dras (1999), aims to relax the isomorphism constraints of synchronous grammars, in this case by generating a loose alignment between parse trees. The Woodsend and Lapata (2011) model is trained on two different datasets: one containing alignments between sentences in Wikipedia and English Simple Wikipedia, and one containing alignments between edits in the revision history of Simple Wikipedia. The latter performs best in their study, and also achieves better scores than the Zhu et al. (2010) system, both when evaluated using BLEU, and on human evaluations of simplicity, grammaticality and meaning preservation. We will directly compare our approach to Woodsend and Lapata (2011), as this is the best performing contemporary system that has the same linguistic scope as ours.

## 2.3 Formalisms and linguistic coverage

The systems summarised above differ primarily in the level of linguistic knowledge they encode. PBMT systems use the least knowledge, and as such are ill equipped to handle simplifications that require morphological changes, syntactic reordering or sentence splitting.

Syntax based approaches use syntactic knowledge. However, both Zhu et al. (2010) and Woodsend and Lapata (2011) use the Stanford Parser (Klein and Manning, 2003) for syntactic structure, and this representation lacks morphological information. This means that some simplification operations such as voice conversion are not handled well. For example, to simplify “trains are liked by John” to “John likes trains”, besides deleting auxiliaries and reordering the arguments of the verb “like”, the verb also needs to agree in number with the new subject (“John”), and take the tense of the auxiliary verb (“are”).

The grammar acquisition process leads to further problems. From an aligned pair “John, who was tired, went to sleep.” and “John was tired. He went to sleep.”, systems would learn a simplification rule that introduces the pronoun “He”. The governing syntax for this rule is the verb “went”; hence, “Susan, who was tired, went to sleep.” might later get simplified as “Susan was tired. He went to sleep.”.

Hand-crafted systems have an advantage here. Such systems would typically use rules that duplicate the noun phrase, generating “John was

tired. John went to sleep.” and “Susan was tired. Susan went to sleep.” Systems such as Sidharthan (2011) use transformation rules that encode morphological changes as well as deletions, re-orderings, substitutions and sentence splitting, and are well suited to handle the voice conversion example above. On the other hand, hand-crafted systems are limited in scope to syntactic simplification. While purely syntactic rules can be written manually, there are too many lexico-syntactic and lexical simplifications to enumerate by hand.

In this paper, we present a hybrid text simplification system that combines manually written synchronous grammars for common syntactic simplifications with a much larger automatically acquired synchronous grammar for lexicalised constructs. Our framework, using dependency representations, is better suited to text simplification. We demonstrate that the higher level of abstraction in dependency parses allows for linguistically correct rules for complex operations such as voice conversion, while also providing a better model of context for lexical simplification.

### 3 Method

We describe a text simplification system that uses a synchronous grammar defined over typed dependencies. We demonstrate that this has specific advantages over previous work on text simplification: (1) it allows for better linguistic modelling of simplification operations that require morphological changes, (2) the higher level of abstraction makes it easy to write and read grammar rules; thus common syntactic operations (such as conversion of passive to active voice) can be handled in this framework through accurate hand-written rules, and (3) It is easier and more elegant to automatically acquire a synchronous grammar from data, compared to synchronous grammars based on constituency-parses. In this section we describe our framework and text simplification system in more detail; then, in section 4, we report an evaluation that compares our system against a human simplification and the Woodsend and Lapata (2011) system.

#### 3.1 Synchronous dependency insertion grammars

Ding and Palmer (2005) introduce the notion of a Synchronous Dependency Insertion Grammar (SDIG) as a tree substitution grammar defined on

dependency trees. They define elementary trees (ETs) to be sub-sentential dependency structures containing one or more lexical items. The SDIG formalism assumes that the isomorphism of the two syntactic structures is at the ET level, thus allowing for non-isomorphic tree to tree mapping at the sentence level. We base our approach to text simplification on SDIGs, but the formalism is adapted for the monolingual task, and the rules are written in a formalism that is suited to writing rules by hand as well as automatically acquiring rules from aligned sentences. Our system follows the architecture proposed in Ding and Palmer (2005), reproduced in Fig. 1. In this paper, we will present the ET Transfer component as a set of transformation rules. The rest of Section 3 will focus on the linguistic knowledge we need to encode in these rules, the method for automatic acquisition of rules from a corpus of aligned sentences, and the generation process.

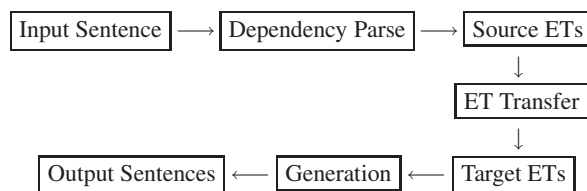


Figure 1: System Architecture

#### 3.2 Extracting synchronous grammars from aligned sentences

To acquire a synchronous grammar from dependency parses of aligned English and simple English sentences, we just need to identify the differences. For example, consider two aligned sentences from the aligned corpus described in Woodsend and Lapata (2011):

1. (a) Also, lichen fungi can reproduce sexually, producing spores.
- (b) Also, lichen fungi can reproduce sexually by producing spores.

An automatic comparison of the dependency parses for the two sentences (using the Stanford Parser, and ignoring punctuation for ease of presentation) reveals that there are two typed dependencies that occur only in the parse of the first sentence, and two that occur only in the parse of the second sentence (in italics):

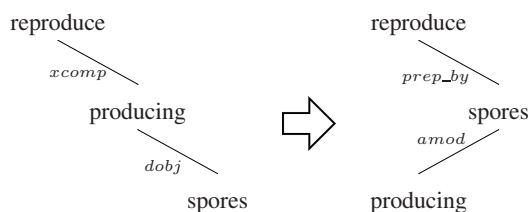


Figure 2: Transduction of Elementary Trees (ETs)

- |   |  |
|---|--|
| <p><b>1. (a)</b><br/> advmod(reproduce, Also)<br/> nn(fungi, lichen)<br/> nsubj(reproduce, fungi)<br/> aux(reproduce, can)<br/> advmod(reproduce, sexually)<br/> xcomp(reproduce, producing)<br/> dobj(producing, spores)</p> | <p><b>1. (b)</b><br/> advmod(reproduce, Also)<br/> nn(fungi, lichen)<br/> nsubj(reproduce, fungi)<br/> aux(reproduce, can)<br/> advmod(reproduce, sexually)<br/> amod(spores, producing)<br/> prep_by(reproduce, spores)</p> |
|---|--|

Thus, to convert the first sentence into the second, we need to delete two dependencies and introduce two others. The rule contains variables (?Xn), which can be forced to match certain words in square brackets:

RULE: PRODUCING2BY\_PRODUCING

1. DELETE
  - (a) xcomp(?X0[reproduce], ?X1[producing])
  - (b) dobj(?X1[producing], ?X2[spores])
2. INSERT
  - (a) amod(?X2, ?X1)
  - (b) prep\_by(?X0, ?X2)

By collecting such rules, we can produce a meta-grammar that can translate dependency parses in one language (English) into the other (simplified English). The rule above will translate “reproduce, producing spores” to “reproduce by producing spores”. This rule is alternatively shown as a transduction of elementary trees in Fig. 2. Such deletion and insertion operations are central to text simplification, but a few other operations are also needed to avoid broken dependency links in the Target ETs (cf. Fig. 1).

Consider lexical simplification; for example, where the word “extensive” is replaced by “big”, resulting in one *amod* relation being deleted and a new one inserted. Now, a third list is automatically created when a variable (?X1) is present in the DELETE list but not the INSERT list. This is a command to move any other relations (edges) involving the node ?X1 to the newly created node ?X2, and ensures correct rule application in new

contexts where there might be additional relations involving the deleted word.

RULE: EXTENSIVE2BIG

1. DELETE
  - (a) amod(?X0[network], ?X1[extensive])
2. INSERT
  - (a) amod(?X0, ?X2[big])
3. NODE OPERATION
  - (a) MOVE: ?X1 → ?X2

We also apply a process of generalisation, so that a single rule can be created from multiple instances in the training data. For example, if the modifier “extensive” has been simplified to “big” in the context of a variety of words in the ?X0 position, this can be represented succinctly as “?X0[networks, avalanches, blizzard, controversy]”. Note that this list provides valid lexical contexts for application of the rule. If the word is seen in sufficient contexts, we make it universal by removing the list. An example of a generalised rule follows:

RULE: \*2BIG

1. DELETE
  - (a) amod(?X0, ?X1[extensive, large, massive, sizable, major, powerful, unprecedented, developed, giant])
2. INSERT
  - (a) amod(?X0, ?X2[big])
3. NODE OPERATION
  - (a) MOVE: ?X1 → ?X2

This rule states that any of the words in “[extensive, large, massive, sizable, major, powerful, unprecedented, developed, giant]” can be replaced by “big” in any lexical context ?X0; i.e., these words are not ambiguous. We acquire rules such as the above automatically, filtering out rules that involve syntactic constructs that we require manually-written rules for (relative clauses, apposition, coordination and subordination). We have extracted 3180 rules from SEW revision histories and aligned SEW-EW sentence pairs. From the same data, Woodsend and Lapata (2011) extract 1431 rules, but these include rules for deletion, as well as for purely syntactic sentence splitting. The 3180 rules we derive are only lexical simplifications or simple paraphrases. We do not perform deletion operations, and use manually written rules for sentence splitting rules

Our approach allows for the encoding of local lexico-syntactic context for lexical simplification. Only if a simplification is seen in many contexts do we generalise the rule by relaxing the lexical context. We consider this a better solution to that implemented in Woodsend and Lapata (2011), who have to discard lexical rules that are only seen once, because they do not model lexical context.

### 3.3 Manual grammars for common syntactic cases

In addition to the automatically acquired grammar as described above, our system uses a small hand-crafted grammar for common syntactic simplifications. As discussed earlier, these rules are difficult to learn from corpora, as difficult morphology and tense manipulations would have to be learnt from specific instances seen in a corpus. In practice, it is easy enough to code these rules correctly. We have 26 hand-crafted rules for apposition, relative clauses, and combinations of the two. A further 85 rules handle subordination and coordination. These are greater in number because they are lexicalised on the conjunction. 11 further rules cover voice conversion from passive to active. Finally, we include 14 rules to standardise quotations; i.e., reduce various constructs for attribution to the form “X said: Y.” Performing this step allows us to simplify constructs embedded within quotations - another case that is not handled well by existing systems. One of the rules for converting passive to active voice is shown below:

RULE: PASSIVE2ACTIVE

#### 1. DELETE

- (a) nsubjpass(?X0, ?X1)
- (b) auxpass(?X0, ?X2)
- (c) agent(?X0, ?X3)

#### 2. INSERT

- (a) nsubj(?X0, ?X3)
- (b) dobj(?X0, ?X1)

#### 3. NODE OPERATIONS

- (a) AGR-TENSE: ?X0 ← ?X2
- (b) AGR-NUMBER: ?X0 ← ?X3

The rule specifies that the node ?X0 should inherit the tense of ?X2 and agree in number with ?X3. This rule correctly captures the morphological changes required for the verb, something not achieved by the other systems discussed in Section 2. The dependency representation makes such

linguistic constraints easy to write by hand. However, we are not yet in a position to learn such constraints automatically. Our argument is that a small number of grammar rules need to be coded carefully by hand to allow us to express the difficult syntactic constructions, while we can harvest large grammars for local paraphrase operations including lexical substitution.

### 3.4 Elementary tree transfer

In this work we apply the simplification rules exhaustively to the dependency parse; i.e., every rule for which the DELETE list is matched is applied iteratively. As an illustration, consider:

The cat was chased by a dog that was barking.

```
det(cat-2, The-1)
nsubjpass(chased-4, cat-2)
auxpass(chased-4, was-3)
det(dog-7, a-6)
agent(chased-4, dog-7)
nsubj(barking-10, dog-7)
aux(barking-10, was-9)
rmod(dog-7, barking-10)
```

Two rules match; the first simplifies relative clauses:

RULE: RELATIVECLAUSE

#### 1. DELETE

- (a) rmod(??X0, ??X1)
- (b) nsubj(??X1, ??X0)

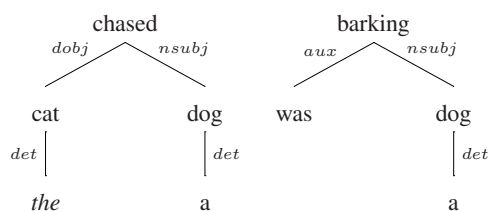
#### 2. INSERT

- (a) nsubj(??X1, ??X0)

This rule removes the embedding “rmod” relation, when there is a subject available for the verb in the relative clause. Then we apply the rule to convert passive to active voice, as described in Section 3.3. Following these two rule applications, we are left with the following list of dependencies:

```
det(cat-2, The-1)
dobj(chased-4, cat-2)
det(dog-7, a-6)
nsubj(chased-4, dog-7)
aux(barking-10, was-9)
nsubj(barking-10, dog-7)
```

This list now represents two trees with *chased* and *barking* as root nodes:



### 3.5 Generating from typed dependency representations

Generating from constituency-based parse trees is trivial, in that leaf nodes need to be output in the order processed by a depth first LR search. The higher level of abstraction of dependency representations makes generation more complicated, as the dependencies abstract away from constituent ordering and word morphology. One option is to use an off the shelf generator; however, this does not work well in practice; e.g., Siddharthan (2011) found that misanalyses by the parser can result in unacceptable word and constituent orders in the generated texts. In the system described here, we follow the generation-light approach adopted by Siddharthan (2011). We reuse the word order from the input sentence as a default, and the synchronous grammar encodes any changes in ordering. For example, in Rule PASSIVE2ACTIVE above, we include a further specification:

#### 4 Traversal Order Specifications

- (a) Node ?X0: [?X3, ?X0, ?X1]

This states that for node ?X0, the traversal order should be subtree ?X3 followed by current node ?X0 followed by subtree ?X1. 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 exists. In the above instance, this would lead us to simplify “*The cat is chased by the dogs*” to “*the dogs chase the cat*”. Details of the generation process can be found elsewhere (Siddharthan, 2011, for example), but to summarise, the gen-light approach implemented here uses four lists:

1. DELETE: List of relations to delete.
2. INSERT: List of relations to insert.
3. ORDERING: List of nodes with subtree order specified
4. NODE-OPERATIONS: List of morphological changes and deletion operations on nodes.

At present the automatically harvested rules do not encode morphological changes. They do however encode reordering information, which is automatically detected from the relative word positions in the original and simplified training sentences.

## 4 Evaluation

We performed a manual evaluation of how fluent and simple the text produced by our simplification system is, and the extent to which it preserves meaning. We use the evaluation set previously used by Woodsend and Lapata (2011), Zhu et al. (2010) and Wubben et al. (2012). This consists of 100 sentences from English Wikipedia, aligned with Simple English Wikipedia (SEW) sentences. Previous work report various automatic measures, including BLEU and readability metrics such as the Flesch-Kincaid Grade Level Index (FKGL). None of these have been validated for the automatic text simplification task, however, and we prefer to conduct an evaluation with human raters.

Our system (henceforth, HYBRID) is compared to QTSG (the system by Woodsend and Lapata (2011) that learns a quasi-synchronous grammar from the same data as the automated component of HYBRID), and the manual gold standard SEW. We selected the first 25 sentences from the evaluation set for which both QTSG and HYBRID had performed at least one simplification<sup>1</sup>. Five human raters<sup>2</sup> were shown sets containing the original Wikipedia sentence, followed by QTSG, HYBRID and SEW in a randomised order. For each such set, they were asked to rate each simplified version for fluency, simplicity and the extent to which it preserved the meaning of the original, using a Likert scale of 1–5, where 1 is totally unusable output, and 5 is output that is perfectly usable. The results are shown in Table 1. Our HYBRID system outperforms QTSG on all three metrics, and is comparable to the SEW version. Raters R1–3 provide very similar ratings, while R4–5 demonstrate a greater preference for the HYBRID system relative to the SEW. The HYBRID system performs best on meaning preservation (in

<sup>1</sup>36 sentences were considered and 11 sentences were excluded in this process. QTSG did not simplify 3 sentences and HYBRID as many as 9, as it does not perform compression operations. One sentence was left unchanged by both systems.

<sup>2</sup>R1–R4 are Computational Linguists, while R5 is a doctoral student in Public Health Communication. None of them are connected with this research, and none of them have previously seen the output of text simplification systems.

Rater	FLUENCY			SIMPLICITY			MEANING PRESERVATION		
	QTSG	HYBRID	SEW	QTSG	HYBRID	SEW	QTSG	HYBRID	SEW
R1	2.60	4.44	4.60	3.04	3.88	4.36	3.16	4.68	4.24
R2	3.08	4.24	4.52	3.20	4.08	4.48	3.28	4.76	4.36
R3	2.40	4.20	4.68	3.12	3.80	4.44	2.96	4.52	3.80
R4	2.32	3.88	3.48	2.92	3.44	3.44	2.72	4.52	3.56
R5	2.00	3.44	3.48	2.00	3.52	3.56	2.48	4.52	3.84
Mean	2.48	4.04	4.15	2.85	3.74	4.05	2.92	4.60	3.96
Median	2	4	4	3	4	4	3	5	4

Table 1: Results of human evaluation with five raters R1–R5. QTSG is the system by Woodsend and Lapata (2011). HYBRID is the system described in this paper, with manual and automatically acquired rules. SEW is the human generated simplification from Simple English Wikipedia. All differences in means for Simplicity and Meaning Preservation are significant ( $p < 0.001$ ; t-test). For Fluency, HYBRID and SEW are significantly better than QTSG ( $p < 0.001$ ; t-test).

large part because it is the only version that does not delete information through sentence compression).

Table 2 shows some examples of simplifications from the evaluation dataset, along with their average scores for fluency, simplicity and meaning preservation. These examples have been selected to help interpret the results in Table 1. QTSG frequently generates fragments (“Komiya is a.”, etc.), likely through incorrect splitting rules in the grammar; this is penalised heavily by the raters. The HYBRID system uses manually written rules for sentence splitting and is more robust in this regard. This is confirmed by looking at standard deviations of ratings. For fluency, QTSG has  $sd = 1.41$ , almost twice that of HYBRID ( $sd = .76$ ). A similar trend is observed for meaning preservation, where QTSG has  $sd = 1.29$ , compared to  $sd = .68$  for HYBRID.

QTSG does perform very elegant compressions in some cases; this is a strength of that system. Our system aims to preserve meaning, which it does rather well. However, this is not necessarily a valid objective. Perhaps future evaluations should distinguish between modifying information in misleading ways (undesirable) and removing peripheral information (desirable). It is clear that the latter, done well, is useful and will be addressed in future work.

An error analysis shows that the main cause of errorful output for our system is parser errors, particularly mistakes in relative clause attachment and clause boundary identification. Methods such as those in Siddharthan (2003b) can be used to improve parser performance on these tasks.

Finally, this work and the cited related work only investigate sentence-level text simplification. There are various discourse level effects that also need to be considered when simplifying larger texts, including sentence ordering (Barzilay et al., 2002; Siddharthan, 2003a; Barzilay and Lapata, 2008), discourse connectives (Siddharthan and Katsos, 2010) and anaphora choice (Nenkova et al., 2005; Siddharthan et al., 2011).

## 5 Conclusions

We have presented a framework for text simplification based on synchronous grammars over typed dependency representations. Our HYBRID system, that uses hand-written rules for common syntactic simplifications, and automatically harvested rules for a much larger set of lexicalised simplifications is more robust than a similar system based on quasi-synchronous tree substitution grammars, outperforming it in terms of fluency, simplicity and meaning preservation. By abstracting away from constituent ordering and morphological variations, our approach allows for linguistically sound rules to be written for complex lexico-syntactic transformations, including passive to active voice. In the version of the system described and evaluated here, changes to morphology and constituent ordering are specified within the rules. Alternately, an off the shelf surface realiser could be used to generate from the dependency representation.

## Acknowledgements

This research is supported by an award made by the EPSRC; award reference: EP/J018805/1.

ORIGINAL	QTSG	HYBRID	SEW
Takanobu Komiyama (born October 3, 1984 in Chiba, Japan) is a Japanese football player who currently plays for the J-league team Kawasaki Frontale.	His father. Komiyama is a.  F=1, S=1.4, M=1	Takanobu Komiyama (born October 3, 1984 in Chiba, Japan) is a Japanese football player. Takanobu Komiyama at present plays for the J-league team Kawasaki Frontale.  F=4, S=3.8, M=4.8	Takanobu Komiyama (born 3 October 1984) is a Japanese football player. He plays for Kawasaki Frontale.  F=4.6, S=4.4, M=4.2
The occupants of Swadlincote often shorten its name to simply 'Swad'.	Swadlincote watch. The occupants often shorten its name to simply 'Swad'. F=2.6, S=2.6, M=3.2	The occupants of Swadlincote often shorten its name to just 'Swad'. F=4.4, S=4, M=5	People from Swadlincote often shorten its name to simply Swad.  F=4.6, S=4.6, M=4.8
Today the nearest railway station is Burton upon Trent, about five miles away, as the railway between Burton and Leicester lost its passenger service under the Beeching Plan in the 1960s and now carries only freight.	Today the nearest railway station is Burton upon Trent, about five miles away, as the railway between Burton and Leicester lost service under the Beeching Plan in the 1960s. It now carries freight.  F=4.2, S=3.6, M=4.6	The closest railway station is now Burton upon Trent, about five miles away. This is because the railway between Burton and Leicester now carries only freight. The railway lost its passenger service under the Beeching Plan in the 1960s.  F=4.4, S=3.8, M=5	The nearest railway station is Burton upon Trent which is five miles away.  F=4, S=4, M=2.2
Since December 2005 it has also been a candidate for joining the European Union and has applied for NATO membership.	Since December 2005, it is a candidate for joining the European Union.  F=4, S=4.2, M=3.6	Since December 2005 it has also been a candidate for joining the European Union. And it has applied for NATO membership. F=4.2, S=4, M=4.8	Since December 2005 it has also been a candidate for joining the European Union. It has applied for NATO membership. F=4.2, S=4, M=4.8
Although most Irish political parties recognize his contribution to the foundation of the modern Irish state, supporters of Fine Gael hold his memory in particular esteem, regarding him as their movement's founding father, through his link to their precursor Cumann na nGaedhael.	The modern Irish state watch. Most Irish political parties recognize his contribution to foundation. Supporters of Gael hold his memory in particular esteem, regarding him as their movement's founding father, through his link to their precursor Cumann na nGaedhael. F=2.6, S=3.2, M=3.8	Supporters of Fine Gael hold his memory in very esteem, regarding him as their movement's founding father, through his link to their precursor Cumann na nGaedhael. But, all Irish political parties recognize his contribution to the foundation of the modern Irish state.  F=3.4, S=3.6, M=4.2	Most Irish political parties think his contributions were important to make the modern Irish state. Members and supporters of Fine Gael remember him in particular as one of the founders of their movement, or its predecessor Cumann na nGaedhael.  F=3.6, S=3.4, M=4.6

Table 2: Examples of simplifications from the test set, along with average scores for (F)luency, (S)implicity and (M)eaning Preservation. 729



## References

- Sandra M Aluísio, Lucia Specia, Thiago AS Pardo, Erick G Maziero, and Renata PM Fortes. 2008. Towards brazilian portuguese automatic text simplification systems. In *Proceedings of the eighth ACM symposium on Document engineering*, pages 240–248. ACM.
- Richard C. Anderson and Alice Davison. 1988. Conceptual and empirical bases of readability formulas. In Alice Davison and G. M. Green, editors, *Linguistic Complexity and Text Comprehension: Readability Issues Reconsidered*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Richard Anderson and Peter Freebody. 1981. Vocabulary knowledge. In John Guthrie, editor, *Comprehension and Teaching: Research Reviews*, pages 77–117. International Reading Association, Newark, DE.
- R. Barzilay and M. Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34(1):1–34.
- R. Barzilay, N. Elhadad, and K. McKeown. 2002. Inferring Strategies for Sentence Ordering in Multi-document News Summarization. *Journal of Artificial Intelligence Research*, 17(3):35–55.
- Isabel L. Beck, Margaret G. McKeown, Gale M. Sinatra, and Jane A. Loxterman. 1991. Revising social studies text from a text-processing perspective: Evidence of improved comprehensibility. *Reading Research Quarterly*, pages 251–276.
- Stefan Bott, Horacio Saggion, and Simon Mille. 2012. Text simplification tools for spanish. In *LREC*, pages 1665–1671.
- Arnaldo Candido Jr, Erick Maziero, Caroline Gasperin, Thiago AS Pardo, Lucia Specia, and Sandra M Aluísio. 2009. Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42. Association for Computational Linguistics.
- Yvonne Canning. 2002. *Syntactic simplification of Text*. Ph.D. thesis, University of Sunderland, UK.
- Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, pages 1041–1044, Copenhagen, Denmark.
- William Coster and David Kauchak. 2011. Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9. Association for Computational Linguistics.
- Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*, pages 19–26.
- Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 541–548. Association for Computational Linguistics.
- Mark Dras. 1999. *Tree adjoining grammar and the reluctant paraphrasing of text*. Ph.D. thesis, Macquarie University NSW 2109 Australia.
- J. Kamalski, T. Sanders, and L. Lentz. 2008. Coherence marking, prior knowledge, and comprehension of informative and persuasive texts: Sorting things out. *Discourse Processes*, 45(4):323–345.
- D. Klein and C.D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.
- 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.
- 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.
- Ani Nenkova, Advait Siddharthan, and Kathleen McKeown. 2005. Automatically learning cognitive status for multi-document summarization of newswire. In *Proceedings of HLT/EMNLP 2005*, pages 241–248, Vancouver, Canada.
- Advait Siddharthan and Napoleon 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.
- Advait Siddharthan, Ani Nenkova, and Kathleen McKeown. 2011. Information status distinctions and referring expressions: An empirical study of references to people in news summaries. *Computational Linguistics*, 37(4):811–842.
- Advait Siddharthan. 2002. An architecture for a text simplification system. In *Proceedings of the Language Engineering Conference (LEC’02)*, pages 64–71, Hyderabad, India.
- Advait Siddharthan. 2003a. Preserving discourse structure when simplifying text. In *Proceedings of*

*the European Natural Language Generation Workshop (ENLG), 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 103–110, Budapest, Hungary.

Advait Siddharthan. 2003b. Resolving pronouns robustly: Plumbing the depths of shallowness. In *Proceedings of the Workshop on Computational Treatments of Anaphora, 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 7–14, Budapest, Hungary.

Advait Siddharthan. 2010. Complex lexico-syntactic reformulation of sentences using typed dependency representations. In *Proceedings of the 6th International Natural Language Generation Conference (INLG 2010)*, Dublin Ireland.

Advait Siddharthan. 2011. Text simplification using typed dependencies: a comparison of the robustness of different generation strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 2–11. Association for Computational Linguistics.

David A Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 23–30. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 409–420. Association for Computational Linguistics.

Sander Wubben, Antal van den Bosch, and Emiel Kraahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530. Association for Computational Linguistics.

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.