Relative clause extraction for syntactic simplification

Iustin Dornescu, Richard Evans, Constantin Orăsan
Research Group in Computational Linguistics
University of Wolverhampton
United Kingdom
{i.dornescu2,r.j.evans,c.orasan}@wlv.ac.uk

Abstract

This paper investigates *non-destructive simplification*, a type of syntactic text simplification which focuses on extracting embedded clauses from structurally complex sentences and rephrasing them without affecting their original meaning. This process reduces the average sentence length and complexity to make text simpler. Although relevant for human readers with low reading skills or language disabilities, the process has direct applications in NLP. In this paper we analyse the extraction of relative clauses through a tagging approach. A dataset covering three genres was manually annotated and used to develop and compare several approaches for automatically detecting appositions and non-restrictive relative clauses. The best results are obtained by a ML model developed using crfsuite, followed by a rule based method.

1 Introduction

Text simplification (TS) is the process of reducing the complexity of a text while preserving its meaning (Chandrasekar et al., 1996; Siddharthan, 2002a; Siddharthan, 2006). There are two main types of simplification: syntactic and lexical. The focus of syntactic simplification is to take long and structurally complicated sentences and rewrite them as sequences of sentences which are shorter and structurally simpler. Lexical simplification focuses on replacing words which could make reading texts difficult with more common terms and expressions. The focus of this paper is on syntactic simplification and more specifically on how to identify noun post-modifying clauses from complex sentences.

The occurrence of embedded clauses due to subordination and coordination increases the structural complexity of sentences, especially in long sentences where such phenomena are more prevalent. Simple sentences are usually much easier to understand by humans and can be more reliably processed by Natural Language Processing (NLP) tools. Psycholinguistic and neurolinguistic imaging studies show that syntactically complex sentences require more effort to process than syntactically simple ones (Just et al., 1996; Levy et al., 2012). For this reason, complex sentences can cause problems to people with language disabilities. At the same time, previous work indicates that syntactic simplification can improve the reliability of NLP applications such as information extraction (Agarwal and Boggess, 1992; Rindflesch et al., 2000; Evans, 2011), and machine translation (Gerber and Hovy, 1998). In the field of syntactic parsing, studies show that parsing accuracy is lower for longer sentences (Tomita, 1985; McDonald and Nivre, 2011). Therefore, the impact of this paper can be two-fold: on the one hand, it can help increase the accuracy of automatic language processing, and on the other hand, it can be used to make text more accessible to people with reading difficulties.

The research presented in this paper was carried out in the context of FIRST[1], an EU funded project which develops tools to make texts more accessible to people with Autism Spectrum Disorders (ASD). In order to have a proper understanding of the obstacles which pose difficulties to people with ASD, a survey of the literature on reading comprehension and questionnaires completed by people with ASD were

[1] http://first-asd.eu

Proceedings of the Workshop on Automatic Text Simplification: Methods and Applications in the Multilingual Society, pages 1–10,
Dublin, Ireland, August 24th 2014.

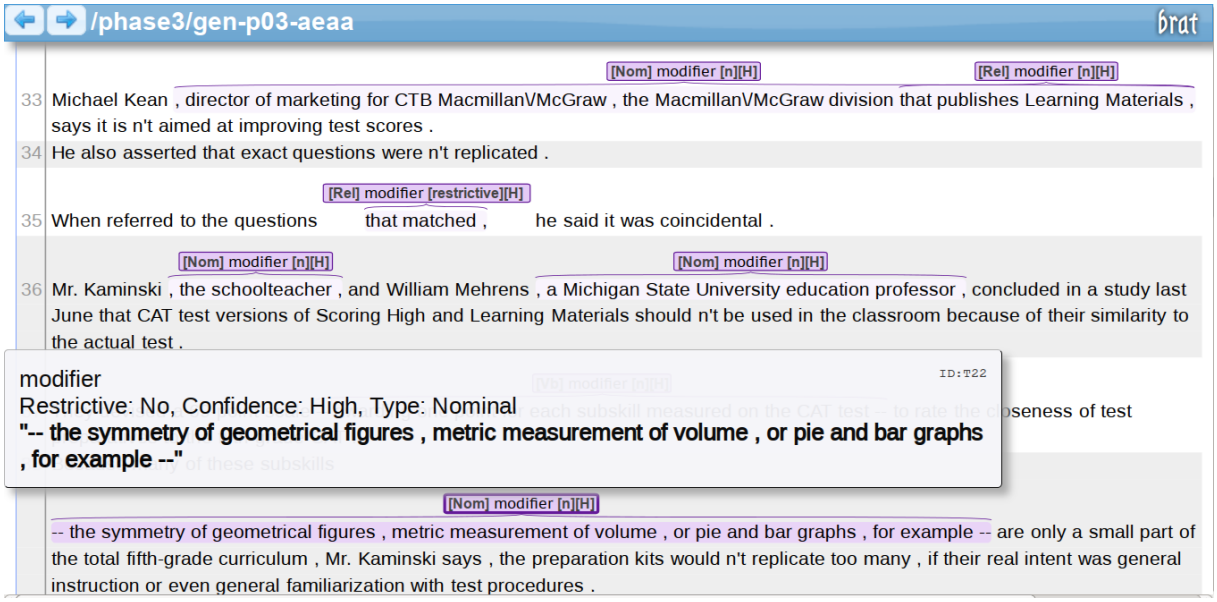

Figure 1: The online annotation using brat

conducted (Martos et al., 2013). The research confirmed that among other types of syntactic complexity, subordinated clauses should be processed by a syntactic simplifier to make the text easier to read.

In this paper, we tackle non-destructive simplification, a form of syntactic simplification in which a clause-based approach is employed to rephrase text in such a way that the meaning of the original text is preserved as much as possible. This is specifically linked to certain types of syntactic structures which can be extracted from the matrix clause without affecting meaning. subordinates, with These types include appositions and non-restrictive relative clauses (Siddharthan, 2002b). This paper presents a method specifically developed for identifying appositions and non-restrictive relative clauses which can be removed from a text without losing essential information.

This paper is structured as follows: Section 2 presents the dataset used to carry out the experiments presented in this paper, including the annotation guidelines and inter-annotator agreement. The machine learning method developed to detect relative clauses is presented in Section 3 and the evaluation results in Section 4. In Section 5, conclusions are drawn.

2 Dataset

To carry out the research presented in this paper, a corpus was annotated. This section presents the annotation guidelines used in the process and discusses issues encountered during the annotation. The annotation was performed using the BRAT[2] tool (Stenetorp et al., 2012). The guidelines were given to the annotators and were explained in a group discussion were several examples were also analysed. Subsequently annotators were given a small set of sentences to trial individually and their questions and feedback led to a revised set of guidelines. Once this training phase was complete, the actual annotation was carried out. The corpus was split randomly each part being annotated by at least two annotators.

2.1 Text genres

The corpus consists of sentences extracted from texts collected in the FIRST project and covers three genres: newswire, healthcare and literature, with some additional sentences from the Penn Treebank (Marcus et al., 1993). The corpus was developed to assist TS for people with ASD (Evans and Orăsan, 2013), following the notion that structurally complex constituents are explicitly indicated by *signs of complexity* such as conjunctions, complementisers and punctuation marks. Evans and Orăsan (2013) developed an annotation scheme and manually labeled these signs.

[2]brat rapid annotation tool `http://brat.nlplab.org/about.html`

Figure 1 shows the interface of the annotation tool. For each annotated span, annotators were asked to fill in three attributes: a) the *type* (relative, nominal, adjectival, verbal, prepositional), b) whether it is *restrictive* (no, yes, unknown) and c) the annotator's *confidence* (low, medium, high). The amount of data in the corpus is listed for each genre in Table 1, i.e. number of sentences and tokens. On average, around half of sentences contain an annotated span, but they occur more frequently in newswire and healthcare than in literature.

Table 1: Corpora used and the total number of annotated spans

Genre (Corpus)	Sentences	Tokens	Spans	Span tokens	Sent. len.	Span len.
healthcare	1214	27379	958	6094	22.55	6.36
news (METER1)	1038	28367	732	5592	27.33	7.64
news (METER2)	1377	37515	1165	9203	27.24	7.90
literature	1946	48620	431	3834	24.98	8.90
News (Penn T.B.)	1733	39740	625	5652	22.93	9.04
Overall	7308	181621	3911	30375	24.85	7.77

2.2 Annotation guidelines

The annotation task involved tagging contiguous sequences of words that comprise post-modifiers of nouns. These are syntactic constituents which follow the head noun in a complex noun phrase (NP), providing additional information about it. We are interested in those post-modifiers which provide additional information but are not part of the parent clause and can be extracted from the sentence without changing its core meaning. These constituents can be either phrases or clauses and are typically bounded by punctuation marks (such as commas, dashes, parentheses) or by functional words (prepositions, relative pronouns, etc.).

The noun post-modifiers of interest are typically clauses or phrases rather than individual words, so not every noun modifier should be marked. Typically they follow the noun phrase whose head they are providing details about and cover several type of subordinated structures (appositions, relative clauses, etc.) In the annotation, the most important aspect is to detect correctly the extent of the annotation (e.g. include surrounding commas). The type is marked as an attribute and evaluated separately. Another attribute indicates whether or not the modifier is a restrictive relative clause.

2.2.1 Restrictive and non-restrictive relative clauses

Restrictive modifiers serve to restrict or limit the set of possible referents of a phrase. In (1a), the subject is restricted to one particular set of chips in a discourse model that may contain many different sets of chips. In (1b), no such restriction is in effect. In this discourse model, all of the chips are made of gallium arsenide and are fragile. Sentences containing restrictive noun post-modifiers require a different method for conversion into a more accessible form than sentences containing non-restrictive noun post-modifiers.

(1) a. *The chips made of gallium arsenide are very fragile* (restrictive)

b. *The chips, which are made of gallium arsenide, are very fragile* (non-restrictive)

Deletion of the noun post-modifier from (1a) produces a sentence that is inconsistent in meaning with the original. All chips, not just the set made of gallium arsenide, are then described as very fragile. When converting sentences with restrictive post-modifiers, it is necessary to generate two sentences: one to put the set of chips made of gallium arsenide into focus and to distinguish this set from the other sets that exist in the discourse model and the other to assert the fact that this set of chips is very fragile. By contrast, deletion of the post-modifier in (1b) produces a sentence that is still consistent in meaning with the original.

In a particular context, it can be quite clear to understand if a specific entity is meant or whether a restricted category of entities is referred to. Normally the sentence is read with two different intonations

to indicate the two different meanings (which is why commas usually mark non-restrictive clauses). The presence, or absence of commas should be used to differentiate ambiguous cases in which not enough context is available to decide which is the intended meaning.

(2) a. *They visited two companies today: one in Manchester and one in Liverpool. The company [which is located] in Manchester was remarkable.* (restrictive)

b. *They visited a company and a school. The company, [which is located] in Manchester, was remarkable.* (non-restrictive)

Restrictive relative clauses are also called *integrated*, *defining* or *identifying* relative clauses. Similarly, non-restrictive relative clauses are called *supplementary*, *appositive*, *non-defining* or *non-identifying* relative clauses.

2.2.2 Types of noun post-modifier

Depending on their syntactic function, there are five types of noun post-modifier:

1. **Relative clauses** (usually marked by relative pronouns *who(m)*, *which*, *that*). These finite clauses are constituents of subclausal elements (noun phrases) within a superordinate clause. They differ from other types of clause such as adverbial clauses, because they are not direct elements of the superordinate clause.[3] They have only an indirect link to the main clause.

(3) *A Bristol hospital that retained the hearts of 300 children who died in complex operations behaved in a "cavalier" fashion towards the parents, an inquiry was told yesterday.*

(4) *The florist [who was] sent the flowers was pleased.*

2. **Nominal-appositives** (which are themselves NPs). Apposition is a relation holding between two NPs (the appositives) in which one serves to define the other. The second NP commonly has a defining role with regard to the first.

(5) *Catherine Hawkins, regional general manager for the National Health Service in the South-west until 1992, appeared before the inquiry yesterday without a solicitor - one should have been provided by the department.*

(6) *My wife, a nurse by training, has helped the accident victim.*

(7) *Goldwater, the junior senator from Arizona, received the Republican nomination in 1964.*

3. **Non-finite clauses** (VP, typically start with an -ing participle or -ed participle verb). These clauses have a non-finite verb and are non-restrictive.[4] These post-modifiers can be regarded as examples of post-modification by a reduced relative clause. To illustrate, in example (8), the non-finite clause is a reduction of the relative clause *who was sitting across from the defendant*.

(8) *Assistant Chief Constable Robin Searle, sitting across from the defendant, said that the police had suspected his involvement since 1997.*

(9) *Lord Melchett led a dawn raid on a farm in Norfolk, causing 17,400 of damage to a genetically modified crop and disrupting a research programme, a court was told yesterday.*

4. **Prepositional phrase post-modification** (PP, typically starting with a preposition). Similar to non-finite clauses, post-modification by PPs, can be regarded as post-modification by 'reduced' relative clauses. For example, the PP in example (10) can be considered a reduction of the relative clause *who is of Chelmsford, Essex*.

[3] In syntax, *elements* are the fundamental units of a clause: subject, verb, object, complement, or adverbial.

[4] They occur in a different tone unit, typically bounded by commas, from the noun head that they modify. They do not restrict or limit the set of possible referents of the complex noun phrase that they modify.

(10) *Boe, of Chelmsford, Essex, admitted six fraud charges and asked for 35 similar offences to be taken into consideration.*

5. **Adjectival post-modification** (AP, including attributes such as height or age). Similar to non-finite clauses, post-modification by adjectival phrases, can be regarded as post-modification by 'reduced' relative clauses. For example, the adjectival phrase in example (11) below can be considered a reduction of the relative clause *who is 58 [years old]*.

(11) *Stanley Cameron, 58 [years old], was convicted in August of 16 counts including vessel homicide and driving under the influence in the November 1997 crash in Fort Lauderdale, Florida.*

(12) *Student Richard, 5ft 1Oins tall, has now left home.*

2.3 Annotation insights

The corpus was split into chunks of roughly 100-150 sentences and each was annotated by 2 to 5 annotators. Sentences were randomly selected from the corpus based on length and the presence of signs of complexity (conjunctions, commas, parentheses). Where annotated spans did not match, a reviewer made a final adjudication.

The agreement on detecting the span of post-modifiers was relatively low, on average, the pairwise F1 score was 54.90%. This is mainly because of the way annotators interpreted the instructions. For example, some annotators systematically marked all parenthetical expressions whereas others never did this. The most frequent error was omission of relevant post-modifiers; annotators typically reached higher precision than recall. This suggests a systematic disagreement which affects files annotated by few annotators. A way to address this problem is by aggregating all annotated spans for each document and asking annotators to confirm which of them are indeed post-modifiers. This is being carried out in a separate study, where a voting scheme is used to mitigate the recall problem.

Looking only at the cases where two annotators marked the same span as a post-modifier, we can investigate the level of agreement reached on the individual attributes: *type* and *restrictiveness*. Annotators reached high pairwise agreement (kappa=0.78) when marking the type of the post-modifier. This suggests that the beginning of the post-modifiers has reliable markers which could be used to automatically predict type. The pairwise agreement for restrictiveness is lower (kappa=0.51), but still good, considering that the two values are not equally distributed (70% of post-modifiers are non restrictive). Possible causes of this are: lack of context (sentences were extracted from their source documents), lack of domain knowledge (where the post-modifers are not about entities, but specialised terminology, such as symptoms, procedures, strategies).

Although agreement on the two attributes can be improved, the biggest challenge is to ensure that all post-modifiers are annotated, i.e. to address situations when only one annotator marks a span. One common cause of disagreement concerns noun modifiers within the same NP, such as prepositional phrases. For example:

(13) *The $2.5 billion Byron 1 plant near Rockford was completed in 1985.*

While this span modifies a noun, it is part of the NP itself, and it is arguably too short to be relevant for rephrasing the sentence in a TS system; it is more likely a candidate for deletion as is the case with sentence compression systems.

Another frequent issue concerns nested modifiers, where annotators usually marked only one of the possible constituents. A related issue is how to deal with nested and overlapping spans, not only from the point of view of the guidelines, but also in the way the annotation is used in practice.

(14) *The new plant, located in Chinchon**, about 60 miles from Seoul,** will help meet increased demand.*

An interesting debate concerns ambiguous constituents which can have several interpretations. In the previous sentence, the second constituent *about 60 miles from Seoul* can be considered an apposition modifying the proper noun *Chinchon*, or a prepositional phrase modifying the verb *located*; both entail a

similar meaning to a human reader. This example illustrates a situation which frequently occurs in natural language text: for stylistic or editorial reasons writers omit words which are implied by the context. The effect is that the syntactic structure becomes ambiguous, but the information communicated to the reader is nevertheless unaffected. This issue also suggests that distinguishing the type of a post-modifier (i.e. relative, nominal, adjectival, verbal, prepositional) only reflects its form and less so its role. For example, it is easy to rephrase most post-modifiers as relative clauses, e.g.:

(15) a. Nominal-appositives: *My wife, **who is** a nurse by training, has helped the accident victim.*

b. Verbal-appositives: *Lord Melchett led a dawn raid on a farm in Norfolk, **which caused** ~~causing~~ 17,400 of damage... , a court was told yesterday.*

c. Prepositional-appositives: *Boe, **who lives in** ~~of~~ Chelmsford, Essex, admitted six fraud charges and asked for 35 similar offences to be taken into consideration.*

d. Adjectival-appositives: *Student Richard, **who is** 5ft 1Oins tall, has now left home.*

During the annotation process there were several issues raised by the annotators, both seeking clarifications of the guidelines as well as identifying new situations in the corpus. A conclusion of the feedback we gathered is that the most important decision is whether a post-modifier is restrictive or not, as this will lead to different strategies for rephrasing the content in order to preserve the meaning as much as possible. The type can be deduced based on the first tokens in the post-modifier.

3 Detection of relative clauses

In this paper we follow the sign complexity scheme introduced by Evans and Orăsan (2013), where punctuation marks and functional words are considered explicit markers of coordinated and subordinated constituents, the two syntactic mechanisms leading to structurally complex sentences.

The signs of syntactic complexity comprise conjunctions (*[*and*]*, *[*but*]*, *[*or*]*), a complementiser (*[*that*]*), wh-words (*[*what*]*, *[*when*]*, *[*where*]*, *[*which*]*, *[*while*]*, *[*who*])*, *punctuation marks ([,], [;], [:]*), and 30 compound signs consisting of one of these lexical items immediately preceded by a punctuation mark (e.g. *[*, and*]*). These signs are automatically tagged with a label indicating type of constituent they delimit, such as finite clauses (EV) or strict appositives (MN), and the position of the sign, such as start/left boundary (SS*) or end/right boundary (ES*). For example, the label ESMA indicates end of an adjectival phrase. An automatic tagger for signs of syntactic complexity was developed using a sequence tagging approach (Dornescu et al., 2013) and is used in this work to select complex sentences from the corpus and to provide linguistic information to the proposed approach.

The two baselines used are rule based systems for detecting post modifiers. System RC1 uses a set of rules to detect appositives which are delimited by punctuation marks and do not contain any verbs. Such expressions are typically nominal appositives or parenthetical expressions e.g.

(16) a. The chief financial officer, Gregory Barnum, announced the merger in an interview.

b. Oxygen can be given with a face mask or through little tubes (nasal cannulae or 'nasal specs') that sit just under your nostrils.

c. The business depends heavily on the creativity of its chief designer, Seymour Cray.

3.1 Rule-based approach

The second baseline used as a reference, DAPR (Detection of Adnominal Post-modifiers by Rules), is a component of a text simplification system for people with autistic spectrum disorders (Evans et al., 2014). Although the system can also rephrase complex sentences, in this paper we only used the appositive constituents detected in a sentence by DAPR.

It employs several hand-crafted linguistic rules which detect the extent of appositions based on the presence of signs of syntactic complexity, in this case punctuation marks, relative pronouns, etc. DAPR

exploits rules and patterns to convert sentences containing noun post-modifiers such as finite clauses (EV), strict appositives (MN), adjective phrases (MA), prepositional phrases (MP), and non-restrictive non-finite clauses (MV) into a more accessible form.

The conversion procedure is implemented as an iterative process. When a pattern matches the input sentence, the detected post-modifier is deleted and the resulting sentence is then processed. The priority of each pattern determines the order in which they are matched when processing sentences which contain multiple left boundaries of relevant constituents (i.e. signs of complexity tagged with certain labels). The patterns are implemented to match the first (leftmost) sign of syntactic complexity in the sentence.

The rules used to convert sentences containing noun post-modifiers exploit patterns to identify both the noun post-modifier and the preceding part of the matrix NP, which can be used to re-phrase the post-modifier as a coherent, stand-alone sentence. Table 3 provides examples of patterns and the strings that they match for each class of signs serving as the left boundary of a noun post-modifier. The patterns are expressed using terms described in Table 2.

Table 2: Terms used in the patterns

Element	Description
w_v	Verbal words, including –ed verbs tagged as adjectives
w_n	Nominal words
w_a	Adjectival words with POS tags JJ, JJS, and VBN
w_{nmod}	Nominal modifiers: adjectives, nouns
w_{nspec}	Nomimal specifiers: determiners, numbers, possessive pronouns
w_{POS}	Word with part-of-speech tag POS (from the Penn Treebank tagset (Santorini, 1990) utilised by the part of speech tagger distributed with the LT TTT2 package)
CLASS	Sign of syntactic complexity of functional class CLASS (Evans and Orăsan, 2013).
”	Quotation marks
B-F	Sequences of zero or more characters

Table 3: Rules used to used to detect noun post-modifiers

Type	Rule	Trigger pattern & Example
SSEV	61	w_{IN} w_{DT}* w_n {wn|of}* SSEV w_{VBD} C sb ”*
		But he was chased for a mile-and-a-half by a passer-by who gave police a description of the Citroen driver.
	7	$w_{\{n\|DT\}}$* w_n SSEV B ESCCV
		Some staff at the factory, which employed 800 people, said they noticed cuts on his fingers.
SSMA	81	w_{NNP}* w_{NNP} SSMA $w_{\{RB\|CD\}}$* w_{CD} ESMA w_{VBD}
		Matthew's pregnant mum Collette Jackson, 24, collapsed sobbing after the pair were sentenced.
	83	w_{NNP}* w_{NNP} SSMA w_{CD} ESMA
		The court heard that Khattab, 25, a trainee pharmacist, confused double strength chloroform water with concentrated chloroform.
SSMN	6	$w_{\{NNP\|NNPS\}}$* $w_{\{n\|a\}}$* w_n SSMN B ESMN
		Mr Justice Forbes told the pharmacists that both Mr Young and his girlfriend, Collette Jackson, 24, of Runcorn, Cheshire, had been devastated by the premature loss of their son.
	3	$w_{\{DT\|PRP\$\}}$ {$w_{\{n\|a\}}$|of}* w_n SSMN B ESMN
		Police became aware that a car, a VW Golf, was arriving in Nottingham from London.
SSMP	4	$w_{\{NNP\|NNPS\}}$* $w_{\{NNP\|NNPS\}}$ SSMP ”* w_{IN} B ESMP
		Justin Rushbrooke, for the Times, said: ”We say libel it is, but it's a very, very long way from being a grave libel.
	1	$w_{\{NNP\|NNPS\}}$* $w_{\{NNP\|NNPS\}}$ {is|are|was|were} w_{CD} SSMP w_{IN} B ESMA
		In the same case Stephen Warner, 33, of Nottingham, was jailed for five years for possession of heroin with intent to supply.
SSMV	12	w_{PRP} w_{RB}* w_{VBD} B SSMV w_{RB}* w_{VBG} C {sb|}
		He attended anti-drugs meetings with Nottinghamshire police, sitting across from Assistant Chief Constable Robin Searle.
	2	$w_{\{NNP\|NNPS\}}$* $w_{\{NNP\|NNPS\}}$ SSMV $w_{\{VBG\|VBN\}}$ B ESMV
		Andrew Easteal, prosecuting, said police had suspected Francis might be involved in drugs and had begun to investigate him early last year.

The underlined examples in Table 3 mark only the noun post-modifier. The patterns also identify the

preceding part of the matrix NP (in square brackets in the example below). The rules include substitutions of indefinite articles by demonstratives or definite articles. Following the method applied to sentences containing noun post-modifiers, rule SSEV-63 would convert:

(17) *But he was chased for a mile-and-a-half by a passer-by who gave police a description of the driver.*

Into the more accessible sequence of sentences:

(18) a. *But he was chased for a mile-and-a-half by [a passer-by].*

b. *[That passer-by] gave police a description of the driver.*

3.2 ML-based approach

As many types of appositive modifiers are simple in structure, we also follow a tagging approach for the task of detecting noun post-modifiers. We employ the common IOB2 format where the beginning of each noun post-modifier is tagged as `B-PM` and tokens inside it are tagged as `I-PM`. All other tokens are tagged as other: `O`. This is similar to a named entity recognition or to a chunking task where only one type of entity/chunk is detected. For comparison we compare the performance of the approach with a rule based method for detecting appositive post-modifiers.

The corpus was used to build two supervised tagging models based on Conditional Random Fields (Lafferty et al., 2001): CRF++[5] and crfsuite[6]. Four feature sets were used. Model A contains standard features used in chunking, such as word form, lemma and part of speech (POS) tag. Model B adds the predictions of baseline system RC1 as an additional feature: using the IOB2 models, tokens have one of three values: B-RC1, I-RC1 or O-RC1. Similarly, model C adds the predictions of baseline system DAPR also using the IOB2 approach. This allows us to test whether the baseline systems are robust enough to be employed as input to the sequence tagging models. Model D adds information about the tokens of the sentence which are signs of syntactic complexity. These are produced automatically.

4 Results and analysis

Results reported by `conlleval`[7] , the standard tool for evaluating tagging, are presented in Table 4. Although the two baselines, RC1 and DAPR, out-perform the CRF++ models, the best overall performance is achieved by the crfsuite models.

The rules employed by the RC1 baseline can be misled by sentences containing enumerations, numerical expressions and direct speech due to false positive matches. Although few and addressing the simplest post-modifiers, the rules perform well.

The more complex baseline, DAPR, appears to be more conservative (it makes the fewest predictions overall), which suggests it covers fewer types of appositions than covered by our dataset. Compared to the previous baseline, DAPR detects more complex appositions and relative clauses with better precision, but with reduced recall.

Although the CRF models also use as features the predictions made by the two baseline models, due to the level of noise, the improvement is small, between 1 and 2 points. Adding information about the tagged signs of syntactic complexity actually has a negative impact on both models, suggesting that the signs are less relevant for this type of syntactic constituent. A large difference in performance is noted between the two CRF tools: whereas CRF++ is outperformed by both baselines, crfsuite achieves much better performance despite using the same input features.

To gain better insights into the performance of the best model, Table 5 presents label-wise results. Given that the average length of a post-modifier is 7, inside tokens (I-PM) are 7 times more prevalent than beginning tokens (B-PM). Despite this, the model achieves similar performance for both (F1 score just below 0.60). The two tables also bring evidence suggesting that detecting the end token of a post-modifier is challenging: although the start is correctly detected for 48.89% of appositives, only 39.94%

[5] `http://crfpp.googlecode.com/svn/trunk/doc/index.html`
[6] `http://www.chokkan.org/software/crfsuite/`
[7] `http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt`

Table 4: Results reported by `conlleval` on the test set (90076 tokens, 2098 annotated post-modifiers)

		#predicted phrases	#correct phrases	accuracy	precision	recall	F1
RC1	baseline	1287	371	81.01	28.83	17.68	21.92
DAPR	baseline	535	163	81.25	30.47	7.77	12.38
CRF++	A:word & POS	3372	289	85.48	8.57	13.78	10.57
	+B:RC1 predictions	3381	315	85.66	9.32	15.01	11.50
	+C:DAPR predictions	3586	319	85.63	8.90	15.20	11.22
	+D:tagged signs	3680	319	85.60	8.67	15.20	11.04
crfsuite	A:word & POS	1391	790	87.54	56.79	37.65	45.29
	+B:RC1 predictions	1437	825	87.55	**57.41**	39.32	46.68
	+C:DAPR predictions	1470	**838**	87.56	57.01	**39.94**	**46.97**
	+D:tagged signs	1481	**838**	87.56	56.58	**39.94**	46.83

are a perfect match. This suggests that more work is necessary to improve the ability to detect post-modifiers but also to better determine their correct extent. The second part is critical to the perceived performance of the TS system, as incorrect detection usually leads to incorrect text being generated for users, whereas a loss in recall may be transparent.

Table 5: Label-wise performance for the best model (crfsuite C)

label	#match	#model	#ref	precision	recall	F1
O	70452	77884	73955	90.46	95.26	92.80
B-PM	1014	1469	2074	69.03	48.89	57.24
I-PM	7406	10723	14047	69.07	52.72	59.80
		Macro-average		76.18	65.63	69.95

5 Conclusions

The paper presents a new resource for syntactic text simplification, a corpus annotated with relative clauses and appositions which can be used to develop and evaluate non-destructive simplification systems. These systems extract certain types of syntactic constituents and embedded clauses and rephrase them as stand-alone sentences to generate less structurally complex text while preserving the meaning intact. A supervised tagging model for automatic detection of appositions was built using the corpus and will be included in a text simplification system.

Acknowledgements

The research described in this paper was partially funded by the European Commission under the Seventh (FP7-2007-2013) Framework Programme for Research and Technological Development (FP7-ICT-2011.5.5 FIRST 287607). We gratefully acknowledge the contributions of all the members of the FIRST consortium for their feedback and comments, and to the annotators for their useful insights.

References

Rajeev Agarwal and Lois Boggess. 1992. A simple but useful approach to conjunct identification. In *Proceedings of the 30th Annual Meeting on Association for Computational Linguistics*, ACL '92, pages 15–21, Stroudsburg, PA, USA. Association for Computational Linguistics.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivation and Methods for Text Simplification. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, pages 1041–1044.

Iustin Dornescu, Richard Evans, and Constantin Orăsan. 2013. A Tagging Approach to Identify Complex Constituents for Text Simplification. In Galia Angelova, Kalina Bontcheva, and Ruslan Mitkov, editors, *Proceedings of Recent Advances in Natural Language Processing, RANLP'13*, pages 221 – 229, Hissar, Bulgaria. RANLP 2011 Organising Committee / ACL.

Richard Evans and Constantin Orăsan. 2013. Annotating signs of syntactic complexity to support sentence simplification. In Ivan Habernal and Vclav Matoušek, editors, *Text, Speech, and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 92–104. Springer Berlin Heidelberg.

Richard Evans, Constantin Orăsan, and Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 131–140, Gothenburg, Sweden, April. Association for Computational Linguistics.

Richard J. Evans. 2011. Comparing methods for the syntactic simplification of sentences in information extraction. *LLC*, 26(4):371–388.

Laurie Gerber and Eduard H. Hovy. 1998. Improving translation quality by manipulating sentence length. In David Farwell, Laurie Gerber, and Eduard H. Hovy, editors, *AMTA*, volume 1529 of *Lecture Notes in Computer Science*, pages 448–460. Springer.

M. A. Just, P. A. Carpenter, and K. R. Thulborn. 1996. Brain activation modulated by sentence comprehension. *Science*, 274:114–116.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289.

J. Levy, E. Hoover, G. Waters, S. Kiran, D. Caplan, A. Berardino, and C. Sandberg. 2012. Effects of syntactic complexity, semantic reversibility, and explicitness on discourse comprehension in persons with aphasia and in healthy controls. *American Journal of Speech–Language Pathology*, 21(2):154 – 165.

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

Juan Martos, Sandra Freire, Ana Gonzlez, David Gil, Richard Evans, Vesna Jordanova, Arlinda Cerga, Antoneta Shishkova, and Constantin Orasan. 2013. User preferences: Updated report. Technical report, The FIRST Consortium.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Comput. Linguist.*, 37(1):197–230, March.

Thomas C. Rindflesch, Jayant V. Rajan, and Lawrence Hunter. 2000. Extracting molecular binding relationships from biomedical text. In *ANLP*, pages 188–195.

Beatrice Santorini. 1990. Part-Of-Speech tagging guidelines for the Penn Treebank project (3rd revision, 2nd printing). Technical report, Department of Linguistics, University of Pennsylvania, Philadelphia, PA, USA.

A. Siddharthan. 2002a. An architecture for a text simplification system. *Language Engineering Conference, 2002. Proceedings*.

Advaith Siddharthan. 2002b. Resolving Attachment and Clause Boundary Ambiguities for Simplifying Relative Clause Constructs. *Association for Computational Linguistics Student Research Workshop*, pages 60–65.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4:77–109.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: A web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 102–107, Stroudsburg, PA, USA. Association for Computational Linguistics.

Masaru Tomita. 1985. *Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems*. Kluwer Academic Publishers, Norwell, MA, USA.