WikiSimple: Automatic Simplification of Wikipedia Articles

Kristian Woodsend and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, UK
k.woodsend@ed.ac.uk, mlap@inf.ed.ac.uk

Abstract

Text simplification aims to rewrite text into simpler versions and thus make information accessible to a broader audience (e.g., non-native speakers, children, and individuals with language impairments). In this paper, we propose a model that simplifies documents automatically while selecting their most important content and rewriting them in a simpler style. We learn content selection rules from same-topic Wikipedia articles written in the main encyclopedia and its Simple English variant. We also use the revision histories of Simple Wikipedia articles to learn a quasi-synchronous grammar of simplification rewrite rules. Based on an integer linear programming formulation, we develop a joint model where preferences based on content and style are optimized simultaneously. Experiments on simplifying main Wikipedia articles show that our method significantly reduces the reading difficulty, while still capturing the important content.

MainEW: Baker was born in Scotland Road, Liverpool, the son of Mary Jane (nee Fleming), a cleaner, and John Stewart Baker, a sailor who was rarely at home.

SimpleEW: Baker was born in Liverpool. Baker's father was a sailor and was Jewish. Baker's mother was Roman Catholic.

MainEW: Owls are the order Strigiformes, comprising 200 bird of prey species. Owls hunt mostly small mammals, insects, and other birds though some species specialize in hunting fish.

SimpleEW: An owl is a bird. There are about 200 kinds of owls. Owls' prey may be birds, large insects (such as crickets), small reptiles (such as lizards) or small mammals (such as mice, rats, and rabbits).

Figure 1: Examples of MainEW and SimpleEW articles.

Introduction

The aim of text simplification is to make texts easier to comprehend for human readers or easier to process automatically. The ability to rewrite text into simpler versions would be of great benefit to people with language[1] impairments, who often encounter problems in understanding written text, to children, who typically lack the high level of linguistic skills required to read texts written for adults, to second language learners whose reading fluency is not as developed as in their first language, and to individuals that face other cognitive demands whilst reading (e.g., driving). Text simplification could be also used as a preprocessing step to improve the performance of other natural language processing tasks such as parsing, machine translation, summarization, and semantic role labeling (Chandrasekar, Doran, and Srinivas 1996; Beigman Klebanov, Knight, and Marcu 2004; Vickrey and Koller 2008).

A large repository of simplified text is the Simple English Wikipedia (henceforth SimpleEW), a project that started as a response to the needs of English learners and English teachers. SimpleEW uses a smaller vocabulary and simpler grammar than the main English Wikipedia (henceforth MainEW). It is aimed at non-native English speakers, children, people with learning disabilities or low reading proficiency. SimpleEW articles are mostly simplifications (both in terms of style and content) of MainEW articles, although main and simple articles can be also written independently of each other. Editors follow a series of guidelines in order to enforce simplicity of language. For instance, they adhere to a basic English vocabulary (i.e., try to use common words), prefer active over passive voice, simple sentence structures (e.g., subject-verb-object), and so on. Examples of MainEW and corresponding SimpleEW texts are shown in Figure 1.

The MainEW and SimpleEW projects both started in 2001. SimpleEW has since accumulated 67,239 articles, whereas MainEW currently counts 3.5M. The growth rate of SimpleEW has therefore been much slower, presumably due in part to the effort involved in rewriting the articles in simple English. Table 1 gives some overview statistics of MainEW and SimpleEW articles. Overall, we observe a high degree of compression at the document level. This is because many of the SimpleEW articles are just "stubs", comprising a single paragraph of just one or two sentences. We believe this is a sign that the SimpleEW articles are less mature, rather than a desired feature. There is also evidence

[1] An example is aphasia, a disorder that impairs the expression and understanding of language as well as reading and writing. It often results from damage (e.g., due to stroke or brain tumor) to portions of the brain responsible for language.

	MainEW	SimpleEW
Sections	10.3 ± 7.9	1.4 ± 2.4
Paragraphs	27.6 ± 24.7	3.3 ± 6.0
Sentences	108.4 ± 115.7	13.3 ± 27.6
Words	2064.9 ± 2295.6	185.7 ± 418.1

Table 1: Overview statistics on corpus of MainEW and SimpleEW articles (mean and standard deviation).

that SimpleEW articles are written in a simpler style, with an average of 13.9 words per sentence compared to 19.0 in the MainEW articles.

In this paper we propose to simplify Wikipedia articles automatically, i.e., to select the most important content and rewrite it in a simpler style. A key insight in our approach is to utilize Wikipedia itself as a large-scale data source for training a simplification model. We learn which content to include in the simpler version from existing articles with versions in MainEW and SimpleEW. To learn simplification rewrites, we take advantage of Wikipedia's collaborative editing process and extract a quasi-synchronous grammar (QG, Smith and Eisner, 2006) from SimpleEW edit histories. QG does not assume a strictly synchronous structure over the source and target sentences. Instead, it identifies a "sloppy" alignment of parse trees (assuming that the target tree is in some way "inspired by" the source tree) and is well suited for modeling the noisy nature of Wikipedia revisions.

Rather than learning the rules of content selection and simplification independently we develop a *joint* model where the entire space of possible simpler versions is searched efficiently through the use of integer linear programming. Advantageously, the ILP framework allows us to capture, through the use of constraints, additional requirements such as overall document length. Aside from promising to drastically speed up the development of SimpleEW, our work is relevant to the general problem of document simplification. Indeed, there is nothing in our approach that limits us to just Wikipedia articles; newspaper text and classroom reading materials are other possible applications.

Related Work

Previous work on text simplification has focused primarily on the sentential- rather than document-level and has been mostly rule-based. Several approaches use hand-crafted syntactic simplification rules aimed at splitting long and complicated sentences into several simpler ones (Carroll et al. 1999; Chandrasekar, Doran, and Srinivas 1996; Vickrey and Koller 2008; Siddharthan 2004). Other work focuses on lexical simplifications and substitutes difficult words by more common WordNet synonyms or paraphrases found in a predefined dictionary (Devlin 1999; Inui et al. 2003; Kaji et al. 2002).

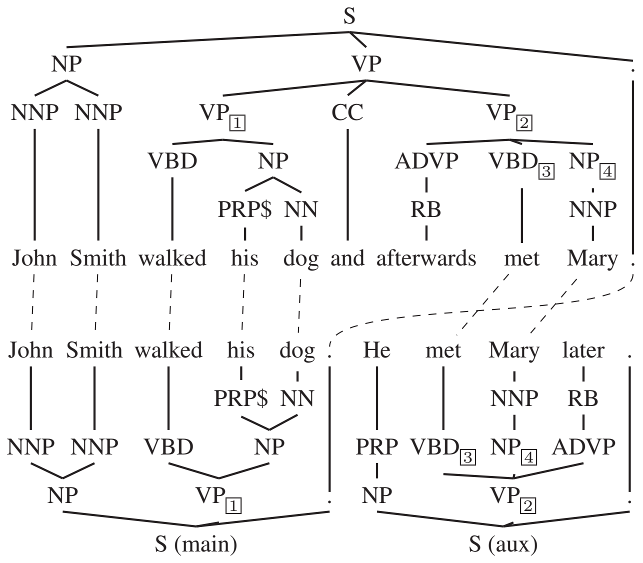
More recently, Yatskar et al. (2010) explore data-driven methods to learn lexical simplifications from Wikipedia revision histories. A key idea in their work is to utilize SimpleEW edits, while recognizing that edits serve other functions, such as vandalism removal or introducing new content. Other researchers treat main and simple Wikipedia articles as a training corpus for learning to automatically discriminate or translate between those two styles (Napoles and Dredze 2010; Zhu, Bernhard, and Gurevych 2010). Yamangil and Nelken (2008) use revision histories from the MainEW as training data for learning a sentence compression model. Sentence compression is related to but distinct from simplification. The latter aims to preserve the meaning of the sentence while rendering it simpler, whereas compression creates shorter sentences (and unavoidably leads to some information loss) without necessarily reducing complexity.

Our work presents an end-to-end system that simplifies Wikipedia articles. We follow Yatskar et al. (2010) in our use of SimpleEW revision histories as training data. However, we do not only learn lexical simplifications. We use quasi-synchronous grammar to learn a wide range of rewrite operations pertaining to simplification. This leads to a more expressive model that captures both lexical and structural simplifications naturally without additional engineering. The QG formalism has been previously applied to parser adaptation and projection (Smith and Eisner 2009), paraphrase identification (Das and Smith 2009), question answering (Wang, Smith, and Mitamura 2007), and title generation (Woodsend, Feng, and Lapata 2010). We also depart from previous work in our formulation of the simplification problem as document- rather than sentence-specific. Our approach simplifies Wikipedia articles as a whole, both in terms of their content and sentential structure. Our model is cast as an integer linear program and optimizes content selection and simplification preferences jointly.

Finally, our work relates to a large body of recent literature on Wikipedia and its potential for a wide range of NLP tasks. Examples include semantic relatedness (Ponzetto and Strube 2007), information extraction (Wu and Weld 2010), ontology induction (Nastase and Strube 2008), and the automatic creation of overview articles (Sauper and Barzilay 2009).

The Simplification Model

Our model takes a MainEW article as input and creates a version that is simpler to read. Some of the desiderata for such a model are the ability to render syntactically complex structures simpler (e.g., through sentence splitting), to use more common words (e.g., that a second language learner may be familiar with), and to capture the gist of the article while reducing its length. In addition, the output must be grammatical and coherent. These constraints are *global* in their scope, and cannot be adequately satisfied by optimizing each one of them individually. Our approach therefore uses an ILP formulation which will provide a globally optimal solution, and which can be efficiently solved using standard optimization tools. Given a MainEW article, our model selects salient phrases (and their sentences) to form the simple article, each of which is simplified (lexically and structurally) through QG rewrite rules. In what follows we first detail how we extract QG rewrite rules as these form the backbone of our model and then formulate the ILP proper.



Rule for splitting into main constituent and auxiliary sentence:
$\langle$VP, VP, S$\rangle \rightarrow \langle$[VP$_{\boxed{1}}$ *and* VP$_{\boxed{2}}$],
[VP$_{\boxed{1}}$], [NP [PRP *He*] VP$_{\boxed{2}}$.]$\rangle$

Rule involving lexical substitution:
$\langle$VP, VP$\rangle \rightarrow \langle$[ADVP [RB *afterwards*] VBD$_{\boxed{3}}$ NP$_{\boxed{4}}$],
[VBD$_{\boxed{3}}$ NP$_{\boxed{4}}$ ADVP [RB *later*]]$\rangle$

Figure 2: A source sentence (upper tree) is revised into two shorter sentences. Dotted lines show word alignments, while boxed subscripts show aligned nodes used to form QG rules. Below, two QG rules learnt from this data.

Quasi-synchronous grammar Our model operates on documents annotated with syntactic information i.e., phrase structure trees. In our experiments, we obtain this information from the Stanford parser (Klein and Manning 2003) but any other broadly similar parser could be used instead.

Given an input sentence S1 or its parse tree T1, the QG constructs a monolingual grammar for parsing, or generating, possible translation trees T2. A grammar node in the target tree T2 is modeled on a subset of nodes in the source tree, with a rather loose alignment between the trees. We extract a QG automatically from a parallel corpus of simplification revisions, however, there is nothing inherent in our method that restricts us to this particular corpus. We take aligned sentence pairs represented as phrase structure trees and build up a list of leaf node alignments based on lexical identity. We align direct parent nodes where more than one child node aligns. A grammar rule is created if the all the nodes in the target tree can be explained using nodes from the source, with a small amount of substitution allowed provided the words are not proper nouns; this helps to improve the quality in what is inherently a noisy process. Finally, QG rules are created from aligned nodes above the leaf node level. An example of the alignment and rule extraction procedure is shown in Figure 2.

We also extract rules in cases where a source sentence is aligned with two or more of target sentences. Such alignments are typically due to sentence splitting, a syntactic transformation commonly used for simplifying long and complicated sentences. Rather than expecting a sentence to split into two at the top level of the parse tree, our intuition is that any node in the source parse tree can generate the second, *auxiliary* sentence, while also aligning with a (simpler) node in the main target sentence (see Figure 2).

Simplified text is created from source sentence parse trees by applying suitable rules recursively. Suitable rules have matching structure; they may also require lexical matching (shown in the example rules in Figure 2 using italics). Where more than one simplification is possible, the alternatives are incorporated into the target parse tree, and it is for the ILP model (described in the next section) to choose which one to use. Note that unlike previous QG approaches, we do not use the probability model proposed by Smith and Eisner (2006); instead the QG is used to represent rewrite operations, and we simply record a frequency count for how often each rule is encountered in the training data.

ILP Formulation The model operates over phrase structure trees, augmented with alternative simplifications. Each phrase in the MainEW document is given a *salience score* representing whether it should be included in the simple version or not. We obtain salience scores using support vector machines (SVMs) but any other standard machine-learning classification technique could be used instead.

Let $\mathcal{S}$ be the set of sentences in a document, $\mathcal{P}$ be the set of phrases, and $\mathcal{P}_s \subset \mathcal{P}$ be the set of phrases in each sentence $s \in \mathcal{S}$. Let the sets $\mathcal{D}_i \subset \mathcal{P}, \forall i \in \mathcal{P}$ capture the phrase dependency information for each phrase i, where each set $\mathcal{D}_i$ contains the phrases that depend on the presence of i. In a similar fashion, $\mathcal{C} \subset \mathcal{P}$ is the set of nodes involving a choice of alternative simplifications (nodes in the tree where more than one QG rewrite rule can be applied); $\mathcal{C}_i \subset \mathcal{P}, i \in \mathcal{C}$ are the sets of phrases that are direct children of each such node, in other words they are the individual simplifications. Let $l_i^{(w)}$ be the length of each phrase i in words, and $l_i^{(sy)}$ its length in syllables. The model is cast as a binary integer linear program. A vector of binary decision variables $x \in \{0,1\}^{|\mathcal{P}|}$ indicates if each phrase is to be part of the output. The vector of auxiliary binary variables $y \in \{0,1\}^{|\mathcal{S}|}$ indicates which sentences are used.

$$\max_x \quad \sum_{i\in\mathcal{P}} (f_i + g_i)x_i + h_w + h_{sy} \tag{1a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{P}} l_i^{(w)} x_i \leq L_{\max} \tag{1b}$$

$$x_j \rightarrow x_i \qquad \forall i \in \mathcal{P}, j \in \mathcal{D}_i \tag{1c}$$

$$\sum_{j\in\mathcal{C}_i} x_j = x_i \qquad \forall i \in \mathcal{C}, j \in \mathcal{C}_i \tag{1d}$$

$$x_i \rightarrow y_s \qquad \forall s \in \mathcal{S}, i \in \mathcal{P}_s \tag{1e}$$

$$x_i \in \{0,1\} \qquad \forall i \in \mathcal{P} \tag{1f}$$

$$y_s \in \{0,1\} \qquad \forall s \in \mathcal{S}. \tag{1g}$$

Our objective function, given in Equation (1a), is the summation of a number of components. f_i, the salience score for each phrase i, measures the importance of the contents of phrase i. Each phrase also has a *rewrite penalty* g_i, where common lexical substitutions, rewrites and simplifications are given a smaller penalty (as we trust them more), compared to rare QG rules. The penalty is a simple log-probability measure, $g_i = \log\left(\frac{n_r}{N_r}\right)$, where n_r is the number of times the QG rule r was seen in the training data, and N_r the number of times all suitable rules for this phrase node were seen. If no suitable rules exist, we set $g_i = 0$.

The final two components of the objective, h_w and h_{sy}, guide the ILP towards simpler language. They draw inspiration from existing measures of readability whose primary aim is to assess whether texts or books are suitable for students at particular grade levels or ages (see Mitchell 1985 for an overview). Intuitively, texts with low readability scores must be simpler to read and understand. The Flesch-Kincaid Grade Level (FKGL) index is a commonly used such measure. It estimates readability as a combination of the average number of syllables per word and the average number of words per sentence. Unfortunately, it is non-linear[2] and cannot be incorporated directly into the objective of the ILP. Instead, we propose a linear approximation. We provide the ILP with targets for the average number of words per sentence (wps), and syllables per word (spw). $h_w(x, y)$ then measures the number of words below this target level that the ILP has achieved:

$$h_w(x, y) = \text{wps} \times \sum_{i \in \mathcal{S}} y_i - \sum_{i \in \mathcal{P}} l_i^{(w)} x_i.$$

When positive, this indicates that sentences are shorter than target, and contributes positively to the readability objective. Similarly, $h_{sy}(x, y)$ measures the number of syllables below that expected, from the target average and the number of words the ILP has chosen:

$$h_{sy}(x) = \text{spw} \times \sum_{i \in \mathcal{P}} l_i^{(w)} x_i - \sum_{i \in \mathcal{P}} l_i^{(sy)} x_i.$$

Constraint (1b) sets the maximum length of the output at $L_{\max}$ words, whereas constraint (1c) enforces grammatical correctness by ensuring that the phrase dependencies are respected and the resulting structure is a tree. Phrases that depend on phrase i are contained in the set $\mathcal{D}_i$. Variable x_i is true, and therefore phrase i will be included, if any of its dependents $x_j \in \mathcal{D}_i$ are true. Note that this constraint also links main phrases to auxiliary sentences, so that the latter can only be included in the output if the main phrase has also been chosen. Where the QG provides alternative simplifications, it makes sense of course to select only one. This is controlled by constraint (1d), and by placing all alternatives in the set $\mathcal{D}_i$ for the node i. Finally, constraint (1e) links phrases to sentences.

Experimental Setup

Corpus In order to train and evaluate the model presented in the previous sections, we downloaded snapshots of MainEW and SimpleEW[3] from which we extracted articles on Animals, Celebrities and Cities (1,654 in total). We selected these categories as a testbed for our approach since they are represented by a large number of articles in MainEW and would thus stand to benefit from an automatic simplification system. Each of these categories was then split into training/test sets (100/46 for Animals, 250/413 for Celebrities and 250/595 for Cities). The corpus was parsed using the Stanford parser (Klein and Manning 2003) in order to label the text with syntactic information.

$\langle\text{S, S, S}\rangle$	$\rightarrow$	$\langle[\text{S}_{\boxed{1}}\text{ , }\textit{and}\text{ S}_{\boxed{2}}], [\text{S}_{\boxed{1}}], [\text{S}_{\boxed{2}}]\rangle$
$\langle\text{S, S, S}\rangle$	$\rightarrow$	$\langle[\text{S}_{\boxed{1}}\ :\ \text{S}_{\boxed{2}}], [\text{S}_{\boxed{1}}], [\text{S}_{\boxed{2}}]\rangle$
$\langle\text{S, S, S}\rangle$	$\rightarrow$	$\langle[\text{S}_{\boxed{1}}\text{ , }\textit{but}\text{ S}_{\boxed{2}}], [\text{S}_{\boxed{1}}], [\text{S}_{\boxed{2}}]\rangle$
$\langle\text{S, S, S}\rangle$	$\rightarrow$	$\langle[\text{NP}_{\boxed{1}}\text{ VP}_{\boxed{2}}], [\text{S}_{\boxed{1}}], [\text{NP [PRP }\textit{It}\text{] VP}_{\boxed{2}}\text{ .}]\rangle$
$\langle\text{NP, S}\rangle$	$\rightarrow$	$\langle[\text{NP}_{\boxed{1}}\text{ , NP}_{\boxed{2}}], [\text{S (NP}_{\boxed{1}}\ \textit{is}\text{ NP}_{\boxed{2}}\text{ .}]\rangle$
$\langle\text{NP, S}\rangle$	$\rightarrow$	$\langle[\text{NP}_{\boxed{1}}, \bar{\text{S}}\ [\textit{which}\text{ VP}_{\boxed{2}}]], [\text{NP}_{\boxed{1}}\text{VP}_{\boxed{2}}\text{ .}]\rangle$
$\langle\text{S, S}\rangle$	$\rightarrow$	$\langle[\text{NP}_{\boxed{1}}\text{ VP}_{\boxed{2}}], [\textit{However}\text{ , NP}_{\boxed{1}}\text{ VP}_{\boxed{2}}\text{ .}]\rangle$

Table 2: QG rules involving syntactic simplification. Sentence-splitting shown as the tuple ⟨source, target, aux⟩ (upper box); others (lower box) as ⟨source, target⟩.

VP [created $\text{PP}_{\boxed{1}}$]	→ VP [made $\text{PP}_{\boxed{1}}$]
VP [located $\text{PP}_{\boxed{1}}$ $\text{PP}_{\boxed{2}}$]	→ VP [found $\text{PP}_{\boxed{1}}$ $\text{PP}_{\boxed{2}}$]
VP [received $\text{PP}_{\boxed{1}}$]	→ VP [got $\text{PP}_{\boxed{1}}$]
VP [announced $\bar{\text{S}}_{\boxed{1}}$]	→ VP [said $\bar{\text{S}}_{\boxed{1}}$]
NP [several $\text{NNS}_{\boxed{1}}$]	→ NP [many $\text{NNS}_{\boxed{1}}$]
NP [$\text{DT}_{\boxed{1}}$ largest $\text{NN}_{\boxed{2}}$]	→ NP [$\text{DT}_{\boxed{1}}$ biggest $\text{NN}_{\boxed{2}}$]
S [$\text{NP}_{\boxed{1}}$ can refer to $\text{NP}_{\boxed{2}}$]	→ S [$\text{NP}_{\boxed{1}}$ could mean $\text{NP}_{\boxed{2}}$]

Table 3: QG rules involving lexical substitution.

Salience Scores To learn salience scores, MainEW document phrases were labeled (as positive or negative) automatically, dependent on if there was a unigram overlap (excluding stop words) between the MainEW phrase and the SimpleEW article. Our feature set comprised a total of 78 surface features, such as section name, sentence and paragraph position information, POS tags, and whether high-scoring tf.idf words were present in the phrase. We learned the feature weights with a linear SVM using the software SVM-OOPS (Woodsend and Gondzio 2009). The hyper-parameters chosen were the ones that gave the best F-scores, using 10-fold validation. The raw prediction values from the SVM were used as salience scores.

QG Rule Extraction QG rules were learned from the revision histories of SimpleEW articles (of all categories, not just those specific categories used in training). We identified revisions where the author had mentioned simplification in the revision comments, and compared each revision to the previous version. Modified sections were identified using the Unix `diff` program, resulting in 14,831 paired

[2] $\text{FKGL} = 0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 1.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$

[3] Dated on 2010-09-16 (MainEW) and 2010-09-13 (SimpleEW) and available from `http://download.wikimedia.org/`.

System	Token count	FKGL Index
MainEW	4726 ± 3066	10.48 ± 2.08
SimpleEW	196 ± 111	8.81 ± 2.65
Preamble	203 ± 149	11.23 ± 2.76
SpencerK	238 ± 52	9.79 ± 2.13
QG-ILP	165 ± 53	7.34 ± 1.79

Table 4: Output length of each system (as number of tokens), and Flesch-Kincaid reading grade levels (lower is simpler). Results shown as mean and standard deviation.

> Baker was born in Scotland Road, Liverpool, the son of Mary Jane, and John Stewart Baker. John Stewart Baker was a sailor. He was rarely at home.

> Owls are the order Strigiformes, making up 200 bird of prey species. Owls hunt mostly small mammals, insects, and other birds.

Figure 3: Examples of text generated by our model, sections corresponding to those in Figure 1.

sentences. QG rules were created by aligning nodes in these sentences as described earlier.

We obtained 269 syntactic simplification rules and 5,779 QG rules involving lexical substitutions. Table 2 illustrates rules involving sentence splitting (see the first four rules). In the 4^{th} example, an NP constituent needs to be transformed into a sentence. The lower box of Table 2 gives examples of QG rules for rewriting a sub-tree into a sentence. Here, the 5^{th} and 6^{th} rules could be used to perform the NP-to-S rewrite needed in the 4^{th} rule. Common lexical substitution rules are shown in Table 3.

SimpleEW Article generation We generated simplified articles for the test documents in our corpus from parsed representations of the corresponding MainEW articles. For each document, we created and solved an ILP (see Equation (1)) parametrized as follows: the maximum token length $L_{\max} = 250$, the target words per sentence (wps) 8, and syllables per word (spw) 1.5. These parameters were empirically tuned on the training set. To solve the ILP model we used the ZIB Optimization Suite software (Achterberg 2007; Koch 2004). The solution was converted into an article by removing nodes not chosen from the tree representation, then concatenating the remaining leaf nodes in order.

Results

We compared the output of our model to the gold standard SimpleEW article, and two baselines. The first baseline is simply the “‘preamble” (the introductory sentences before any section heading) of the MainEW article. The second baseline is a summary based on sentence extraction (the highest scoring sentences according to our SVM), with lexical simplifications[4] provided by the SimpleEW editor “SpencerK” (Spencer Kelly).

Simplicity					
System	1^{st}	2^{nd}	3^{rd}	4^{th}	Rating
SimpleEW	**0.60**	0.20	0.05	0.15	**2.70**
Preamble	0.05	0.15	**0.40**	**0.40**	1.54
SpencerK	0.15	0.20	0.40	0.25	1.87
QG-ILP	0.20	**0.45**	0.15	0.20	2.20

Informativeness					
System	1^{st}	2^{nd}	3^{rd}	4^{th}	Rating
SimpleEW	0.20	0.05	**0.30**	0.45	1.66
Preamble	0.25	0.05	0.15	**0.55**	1.66
SpencerK	0.15	**0.55**	0.3	0	2.37
QG-ILP	**0.40**	0.35	0.25	0	**2.63**

Table 5: Rankings (shown as proportions) and mean ratings given to systems by human subjects.

Although there are no automatic measures of simplicity we are aware of, a plethora of measures have been developed over the years for assessing the readability of written documents. Since readability and simplicity are related (e.g., texts written for individuals with low reading proficiency must be simple), we used the FKGL index[5] to evaluate our model on the test set (1,054 articles). In addition, we evaluated the generated articles by eliciting human judgments. Participants were presented with the simple article generated by our system and the three comparison versions. They were also given access to the full article in MainEW. Participants were asked to rank the four versions in order of simplicity (is the text simple or complicated?) and informativeness (does the article capture the most important information in the original MainEW article?). We obtained rankings for nine documents in total (three randomly selected from each category). The study was completed by 15 volunteers, all non-native English speakers with IELTS scores at least 6.5 (“competent user”). Our experimental study deliberately addressed non-native speakers as these are more likely to consult the SimpleEW. As the evaluators were non-native speakers, we did not ask them explicitly to rank the documents in terms of fluency; instead, we instructed our participants to consider a text complicated to read in cases where the sentences were ungrammatical and the text was difficult to comprehend.

Table 4 shows the results of the FKGL index on the output of our system (QG-ILP), the two baselines (Preamble, SpencerK) and the SimpleEW upper bound. We also measured the FKGL index of the MainEW. As can be seen, the MainEW text is considerably harder to read than that of SimpleEW, and the introductory text in the preamble is more complex still. The output of our model, QG-ILP, is simpler than both baselines, and also simpler than the SimpleEW gold standard. The lengths of articles generated by each system are comparable, and length does not appear to in-

[4] http://www.spencerwaterbed.com/soft/simple/about.html

[5] The FKGL readability score corresponds to a grade level; e.g., a score of 8.2 would indicate that the text is understandable by an average student in 8th grade (usually around ages 13–14 in the US). Other readability measures such as the Flesch Reading Ease obtained similar results which we omit for lack of space.

fluence simplicity. Examples of the output of our system are given in Figure 3. All FKGL means in Table 4 are significantly different with 99% confidence using a paired samples t-test, with the exception of the pair MainEW–SpencerK: it appears that the simplifications introduced by lexical substitutions alone make little impact on the FKGL index.

The results of our human evaluation study are shown in Table 5. Specifically, we show, proportionally, how often our participants ranked each system 1^{st}, 2^{nd} and so on. SimpleEW is considered simplest (and ranked 1^{st} 60% of the time). QG-ILP is ranked second best 45% of time followed by the Preamble which is ranked either 3^{rd} or 4^{th}. With regard to informativeness, participants prefer QG-ILP and SpencerK over the SimpleEW and the Preamble. Both QG-ILP and SpencerK base their output on sentences highlighted by the SVM, and thus potentially capture more of the article's content than SimpleEW (where the editor may decide to mention only a few topics) and the Preamble (which by being introductory contains only partial information).

We further converted the ranks to ratings on a scale of 1 to 4 (assigning ratings 4…1 to rank placements 1…4). This allowed us to perform Analysis of Variance (ANOVA) which revealed a reliable effect of system type. We used post-hoc Tukey tests to examine whether the mean ratings for each system (shown in Table 5) differed significantly. They showed that all systems were significantly worse ($p < 0.01$) than SimpleEW in terms of simplicity but QG-ILP was significantly better than SpencerK and Preamble ($p < 0.01$). In terms of informativeness, QG-ILP was significantly better than all other systems, and SpencerK better than Preamble and SimpleEW ($p < 0.01$).

Conclusions

In this paper we have presented an end-to-end system that simplifies Wikipedia articles. Central to our approach is the formulation of the document simplification task as a joint content selection and text rewriting problem. Our model learns appropriate content from observations of same-topic articles in the main Wikipedia and its simpler variant. Articles are rewritten in simpler language using a quasi-synchronous grammar that captures a wide range of lexical and structural simplifications, including sentence splitting. Importantly, these rules are learned from the revision history of Simple Wikipedia itself without any manual intervention. An integer linear program optimizes the output for both informativeness and simplicity. Experimental results show that our model creates informative articles that are simpler to read than competitive baselines. We argue that our approach is computationally efficient, portable, and viable in practical applications. In the future, we intend to enrich our model with some notion of discourse-level document structure which we also plan to learn from Wikipedia.

References

Achterberg, T. 2007. *Constraint Integer Programming*. Ph.D. Dissertation, Technische Universität Berlin.

Beigman Klebanov, B.; Knight, K.; and Marcu, D. 2004. Text simplification for information-seeking applications. In *Proceedings of ODBASE*, volume 3290 of *Lecture Notes in Computer Science*, 735–747. Agia Napa, Cyprus: Springer.

Carroll, J.; Minnen, G.; Pearce, D.; Canning, Y.; Devlin, S.; and Tait, J. 1999. Simplifying text for language-impaired readers. In *Proceedings of the 9th EACL*, 269–270.

Chandrasekar, R.; Doran, C.; and Srinivas, B. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th COLING*, 1041–1044.

Das, D., and Smith, N. A. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the ACL-IJCNLP*, 468–476.

Devlin, S. 1999. *Simplifying Natural Language for Aphasic Readers*. Ph.D. Dissertation, University of Sunderland.

Inui, K.; Fujita, A.; Takahashi, T.; Iida, R.; and Iwakura, T. 2003. Text simplification for reading assistance: A project note. In *Proceedings of Workshop on Paraphrasing*, 9–16.

Kaji, N.; Kawahara, D.; Kurohashi, S.; and Sato, S. 2002. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th ACL*, 215–222.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st ACL*, 423–430.

Koch, T. 2004. *Rapid Mathematical Prototyping*. Ph.D. Dissertation, Technische Universität Berlin.

Mitchell, J. V. 1985. *The Ninth Mental Measurements Year-book*. Lincoln, Nebraska: University of Nebraska Press.

Napoles, C., and Dredze, M. 2010. Learning simple Wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, 42–50.

Nastase, V., and Strube, M. 2008. Decoding Wikipedia categories for knowledge acquisition. In *Proceedings of the 23rd AAAI*, 1219–1224.

Ponzetto, S. P., and Strube, M. 2007. Knowledge derived from Wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research* 30:181–212.

Sauper, C., and Barzilay, R. 2009. Automatically generating Wikipedia articles: A structure-aware approach. In *Proceedings of ACL-IJCNLP*, 208–216.

Siddharthan, A. 2004. Syntactic simplification and text cohesion. *Research on Language and Computation* 4(1):77–109.

Smith, D., and Eisner, J. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of Workshop on Statistical Machine Translation*, 23–30.

Smith, D. A., and Eisner, J. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*, 822–831.

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

Wang, M.; Smith, N. A.; and Mitamura, T. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of EMNLP-CoNLL*, 22–32.

Woodsend, K., and Gondzio, J. 2009. Exploiting separability in large-scale linear support vector machine training. *Computational Optimization and Applications*. published online.

Woodsend, K.; Feng, Y.; and Lapata, M. 2010. Title generation with quasi-synchronous grammar. In *Proceedings of EMNLP*, 513–523.

Wu, F., and Weld, D. S. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th ACL*, 118–127.

Yamangil, E., and Nelken, R. 2008. Mining Wikipedia revision histories for improving sentence compression. In *Proceedings of ACL-08: HLT, Short Papers*, 137–140.

Yatskar, M.; Pang, B.; Danescu-Niculescu-Mizil, C.; and Lee, L. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Proceedings of NAACL*, 365–368.

Zhu, Z.; Bernhard, D.; and Gurevych, I. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 1353–1361.