Learning to Simplify Children Stories with Limited Data

Tu Thanh Vu[1], Giang Binh Tran[2], and Son Bao Pham[1]

[1]University of Engineering and Technology, Vietnam National University, Hanoi
{tuvt,sonpb}@vnu.edu.vn
[2]L3S Research Center, Germany
gtran@l3s.de

Abstract. In this paper, we examine children stories and propose a text simplification system to automatically generate simpler versions of the stories and, therefore, make them easier to understand for children, especially ones with difficulty in reading comprehension. Our system learns simplifications from limited data built from a small repository of short English stories for children and can perform important simplification operations, namely splitting, dropping, reordering, and substitution. Our experiment shows that our system outperforms other systems in a variety of automatic measures as well as human judgements with regard to simplicity, grammaticality, and semantic similarity.

Keywords: text simplification, readability, comparable corpora

1 Introduction

The Internet contains a tremendous amount of documents written in very different readability levels. That fact causes difficulties in understanding the content for readers. For example, children may understand simple fairy-tales or articles about nature and animals, but may not understand technical reports which are more intended for engineers or scientists. Unfortunately, traditional information retrieval systems have not paid much attention to the comprehension ability of users, and very likely deliver some texts that are beyond their readability levels.

A lot of research in psycholinguistics showed that there is a large number of people struggling to understand general text documents, such as children, hearing/deaf people, aphasic readers or second language learners (e.g., [12]). Hence, it is desirable to adapt and simplify written texts to make them more comprehensible to these individuals. The aim of our work is to provide appropriate reading materials for children, however, people with poor reading comprehension skills can also benefit from such simplified texts.

Lots of efforts put into understanding poor comprehenders found that they fail to master high-level cognitive text processing skills, particularly, (s1) coherent use of cohesive markers such as connectives ('because', 'before', 'after') that signal relations in text, (s2) inference-making from different or distant parts of a text, integrating them coherently (e.g., [8, 1]), (s3) detection of inconsistencies in texts ([4, 9]). Such reasoning skills are very likely to be causally implicated in the development of deep text comprehension - "integration and inference making are crucial for good text comprehension" ([2]). Hence, a good text simplifier should make text documents clear on these text processing skills.

One of the conspicuous challenges for text simplification is the lack of data available. Current trends are to leverage a large repository of simplified language called Simple English Wikipedia (SEW), a simpler version of Main English Wikipedia (MEW). We can generate a large parallel corpus by pairing sentences from MEW with corresponding sentences from SEW (e.g., [13]), and then explore data-driven methods to learn simplification models. However, there is no guarantee that the generated corpus will satisfy (s1), (s2), and (s3). It seems to be more suitable for sentence simplification purpose rather than document simplification purpose. Recently, there is such a corpus of high quality made available by TERENCE[1] consortium which contains a set of English stories[2], of which each is simplified (by experts) into several versions with different reading difficulty levels that reflects (s1), (s2) and (s3). The corpus appears to be a valuable resource for learning simplifications which can mimic simplifying procedures, mirroring high-level text processing skills. Nonetheless, its small size makes it challenging for previous methods in learning efficient simplification models. The work presented in this paper investigates the use of the afore-mentioned corpus to learn a Bayesian probability model for simplifying children documents (or stories).

Summary of contribution:

- A Bayesian-based framework learnt from limited data for simplifying children stories.
- A method that leverages shallow parsing rather than full parsing in defining the syntax of sentences to improve the performance of simplification model.
- Strategies for rule generalization and manual revision of simplification rules to maximize the benefits of data, especially for small-scale dataset.
- We do quantitative and qualitative evaluations of the proposed methods in comparison with the-state-of-the-art methods.

2 Related Work

Most earlier work in text simplification utilized hand-crafted rules to split long and complicated sentences into several simpler ones ([3]). Other work focuses on lexical simplifications and substitutes low-frequency words by more common synonyms derived from WordNet, or paraphrases found in a predefined vocabulary list ([5]) or their dictionary definitions ([6]). The task can be treated as a monolingual machine translation task with the complex sentence as the source and the simple one as the target and make use of large-scale parallel corpora of paired articles from SEW and MEW for training models ([16, 14]). Wikipedia revision history is also an useful resource for learning simplifications ([15]).

[13] investigate the use of an aligned MEW-SEW corpus and SEW edit histories to learn a model based on Quasi-synchronous grammar then use an integer linear programming model to select the most appropriate simplification. Our work is similar to [13] in learning rewrite rules which involve sentence splitting, syntactic and lexical simplification. However, rather than exploiting Wikipedia, we examine children documents i.e., stories created and rewritten into different complexity levels by experts. For each type of rules, we suggest a general

[1] www.terenceproject.eu - An European project supporting poor comprehenders and their educators)

[2] http://www.terenceproject.eu/repository/booken/booken.html

form suitable for text analysis processes. More concretely, we use shallow parsing rather than full parsing in defining the syntax of sentences for less error-prone. Furthermore, we propose some strategies for rule generalization and manually revise all of rewrite rules to maximize the benefits of the small-scale dataset. Our approach also differs from [13] in the sense that instead of using integer linear programming, we rely on a Bayesian probability model to identify the most appropriate simplification from the space of possible ones.

3 Simplification Dataset

We construct a simplification dataset from two story levels: Level 1 and Level 2 published in TERENCE story repository. Stories at Level 2, which are inherited from all simplifications that reflects high-level text processing skills of human (coherence and cohesion levels), are then simplified on lexicon and grammar to create corresponding stories at Level 1. We therefore consider stories at Level 2 as the original texts, and corresponding stories at Level 1 as the simplified texts. To generate simplification dataset, we first pair each story at Level 2 to its simpler one and then represent them as list of continuous sentences, of which each is pre-processed by tokenization/POS-tagging with the GATE[3] toolkit and the Stanford Parser package ([7]). At the sentence level, we modify METEOR[4] to build up an automatic sentence alignment module based on exact, stem, synonym, and paraphrase matches between words and phrases in sentences. To achieve the best sentence alignment on our dataset, we manually revise all automatically generated alignments to recognize bad ones. Finally, we obtain a list of 1-1 pairs in which each sentence at Level 2 is aligned with only one sentence at Level 1, and 1-n pairs in which each sentence at Level 2 is aligned with several sentences at Level 1. Table 1 shows some statistics for our dataset.

Table 1. Statistics for the simplification dataset. #Sen.Pairs: the number of sentence pairs; #(1-n).Sen.Pairs: the number of 1 to n sentence alignments; Avg.Sen.Len, Avg.Token.Len: the average sentence length and average token length, respectively

#Sen.Pairs	#(1-n).Sen.Pairs	Avg.Sen.Len		Avg.Token.Len	
		Level 2	Level 1	Level 2	Level 1
1050	46	17.17	16.63	3.94	3.80

4 Sentence Simplification Model

Our model operates on individual sentences. The simplification process first considers whether or not to split a given original sentence into several shorter ones, then may drop or reorder its components, or substitute difficult words with their simpler synonyms or paraphrases. We rely on a Bayesian probability model to identify the most appropriate simplification for the input sentence. Our model architecture is illustrated in Figure 1.

[3] http://gate.ac.uk/

[4] An automatic system for machine translation evaluation (available at http://www.cs.cmu.edu/∼alavie/METEOR/)

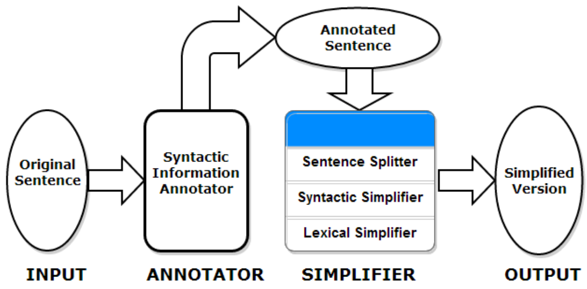

Fig. 1. Sentence simplification model architecture.

4.1 Rule Generation

We annotate sentences with syntactic information in the form of phrase structure trees. For each parse tree pairs $T1 \rightarrow T2$, corresponding to an aligned sentence pair $S1 \rightarrow S2$ from the training dataset, we further align nodes in $T1$ with corresponding nodes in $T2$ as follows: first, we construct a list of leaf node alignments from the results of METEOR toolkit as used in preparing simplification data. Next, we recursively align the parent of aligned nodes in $T1$ with corresponding ones in $T2$, as showed in Algorithm 1. After that, rewrite rules are

Algorithm 1: Parent node alignment.

```
ParentNode node1, node2;
for each (node1 in T1, node2 in T2) if (!isAligned(node1, node2))
   if (node1.numChildren() == 1 && node2.numChildren() == 1)
      if(numChildNodeAlignment(node1, node2) == 1)
         align(node1, node2);
   else
      if(numChildNodeAlignment(node1, node2) > 1)
         align(node1, node2);
```

generated from aligned nodes by considering tree-to-tree (lexical and structural) transformation patterns. In what follows we detail how we extract rewrite rules.

(a) Sentence splitting Sentence splitting rules are created from 1-n aligned parse tree pairs ($T \rightarrow T1.T2...Tn$), corresponding to 1-n aligned sentence pairs ($S \rightarrow S1.S2...Sn$) in which each sentence at Level 2 is aligned with two or more consecutive sentences at Level 1. These rules allow long and syntactically complicated sentences to be split into several shorter ones. In our work, we use shallow parsing also called "chunking" to describe the syntax in sentences. Concretely, constituents or word groups such as noun and verb phrases are identified, but their internal structures are ignored. This should help the performance of the system become more robust and less error-prone. To illustrate, we use a running example for a given sentence pair:

"Lewis installed big electric fans, while Charles installed big air conditioners." → "Lewis installed big electric fans. In the meanwhile, Charles installed big air conditioners."

After aligning corresponding nodes in the aligned parse tree pair and chunking, we obtain (subscripts show aligned nodes):

[NP Lewis]$_{[1]}$ [VP installed]$_{[2]}$ [NP big electric fans]$_{[3]}$, [SBAR while] [NP Charles]$_{[4]}$ [VP installed]$_{[5]}$ [NP big air conditioners]$_{[6]}$. → [NP Lewis]$_{[1]}$ [VP installed]$_{[2]}$ [NP big electric fans]$_{[3]}$. [PP In] [NP the meanwhile] , [NP Charles]$_{[4]}$ [VP installed]$_{[5]}$ [NP big air conditioners]$_{[6]}$.

A sentence splitting rule can be generated automatically as follows:

NP$_{[1]}$ VP$_{[2]}$ NP$_{[3]}$, while/SBAR NP$_{[4]}$ VP$_{[5]}$ NP$_{[6]}$.

→ NP$_{[1]}$ VP$_{[2]}$ NP$_{[3]}$. In the meanwhile, NP$_{[4]}$ VP$_{[5]}$ NP$_{[6]}$.

(b) Syntactic Simplification We can drop or reorder components in an original sentence to make it more concise or easier to comprehend than the original. To learn rules for these operations, we consider all aligned sub-tree pairs, where the syntactic structures of the original and simplified sub-trees do not match. We also use shallow parsing technique to render more general resulting rules. For example,

[VBD painted] [NP the car]$_{[1]}$ [JJ white]$_{[2]}$

→ [VBD made] [NP the car]$_{[1]}$ [IN with] [DT the] [JJ white]$_{[2]}$ [NN paint].

A syntactic rule is established automatically as follows:

painted/VBD NP$_{[1]}$ JJ$_{[2]}$ → made NP$_{[1]}$ with the JJ$_{[2]}$ paint.

(c) Lexical Simplification Lexical simplification rules are learned from leaf node alignments as described above, and also from aligned sub-tree pairs in which the original and simplified sub-trees have the same syntactic structure and only one lexical difference exists. We only extract the words and corresponding POS tags involved rather than take the syntactic context into consideration. For example, "trophy/NN → cup", "turquoise/NN → deep blue", "stunned/VBD → shocked".

Rule Manual Revision There are still a number of bad rules generated in the process of rules automatic generation. The underlying cause may be due to wrong alignments in the previous steps. In the task of text simplification, we must create simplifications that reduce the reading difficulty of the input text, while maintaining grammaticality and preserving its meaning. To satisfy these constraints, the rewrite rules need to be as good as possible and should be meticulously evaluated before application. We therefore revise all the automatically generated rules to recognize bad ones. Bad rules are then manually adjusted to acquire good versions or removed directly from the training data if necessary.

Rule Generalization We examine some strategies for generalizing rewrite rules to enrich the training data. Our strategies are mainly based on word form, word similarity, context similarity and grammatical roles. For examples, from a lexical rule "began → started" , we can further obtain other ones (based on word form), in particular, "begin → start", "begun → started", "begins → starts", "beginning → starting". Similarly, for a syntactic rule "shrunk PRP → made PRP smaller" extracted from an alignment that is "shrunk her → made her smaller" , we can place PRP (personal pronoun) with NP (noun phrase) as well as make the use of different word forms to produce a series of reliable rules as follows:

{"shrink PRP → make PRP smaller", "shrunk PRP → made PRP smaller", "shrinks PRP → makes PRP smaller", "shrinking PRP → making PRP smaller"},

{"shrink NP → make NP smaller", "shrank NP → made NP smaller", "shrunk NP → made NP smaller", "shrinks NP → makes NP smaller", "shrinking NP → making NP smaller"}.

4.2 Rule Identification and Application

Where there is more than one possible simplification, we identify the best sequence of simplifications. In our work, rewrite rules are all written in a general form, LHS → RHS. In each rule, the LHS describes syntactic information of a matching object while the RHS presents its simplified version. Therefore, in a space of all possible rules, we appreciate rules in which the LHS describes more details about syntactic information of objects than in other possible rules. We further construct a function to assess the level of detail of each rule. Besides, we also apply a Bayesian probability model to identify the most likely rules among different possibilities. Specifically, the probability of a rule LHS → RHS from the training data is estimated as the product

$$\mathcal{P}(\mathbf{LHS}).\mathcal{P}(\mathbf{LHS} \rightarrow \mathbf{RHS}|\mathbf{LHS})$$

Note that $\mathcal{P}$(LHS)= $\frac{t_{LHS}}{n}$, where n is the total number of rules and t_{LHS} is the number of times LHS appears in the training data. The conditional probability $\mathcal{P}(LHS \rightarrow RHS|LHS)$ is the probability of some event LHS → RHS, given the occurrence of some other event LHS.

5 Experiments

5.1 Experiment Setup

32 English stories from the golden data were randomly partitioned into a training set (25 stories) and a test set (7 stories). We extracted 24 splitting rules, 298 syntactic rules and 464 lexical rules from the training set. The test set contains 230/238 original/simplified sentences excluded from training.

The performance of our system was compared to one of the state-of-the-art systems, namely [13]'s system[5] (**Woodsend**) which learns simplications from Wikipedia (revision histories and MEW-SEW aligned corpus), and a simple baseline (**SpencerK**) that merely relies on lexical substitutions provided by the SEW editor "SpencerK" (Spencer Kelly)[6]. We also included simplifications created by experts as a gold standard (**Experts**).

5.2 Evaluation

We evaluated the system's output in two ways, using automatic evaluation measures and human judgements as follows:

Automatic Evaluation Measures In line with previous work on text simplification (e.g., [16] [13]), we also used a variety of measures ranging from basic statistics such as the average length of tokens, the average number of tokens in one sentence, the total number of simplified sentences, to more complex measures such as OOV%, FKGL, BLEU, TERp.

[5] http://homepages.inf.ed.ac.uk/kwoodsen/demos/simplify.html

[6] We are grateful to Spencer Kelly for providing us with his list of simple words and simplications

Human Judgements We compared sentences generated by experts (the gold standard) and the three automated systems on our test set of 230 sentences. In total, our material consists of 920 (230 × 4) sentences. We conducted two experiments. First, we randomly selected 80 (20 × 4) original-simplified sentences, corresponding to 20 original sentences from our materials, to manually analyse their quality. Second, we recruited 15 participants to judge the quality of sentences from our material, without awareness of which method was used to generate them. Each participant was asked to rate (in five-point scale) whether each simplified sentence is simpler than the original, is good at grammar, and preserves the main content of the original, respectively. All of participants are college students who are proficient in English and committed to bringing us objective and serious assessments.

5.3 Results and Discussion

Basic statistics The first four columns in Table 2 show the results of basic statistics. All simplified sentences get better scores than the input sentences in terms of the average length of tokens (*tokLen*) which may roughly reflect the lexical difficulty. This indicates that at the lexical level, long and difficult words were partly substituted by simpler and shorter ones. The results of this score also show that our system performed lexical simplification better than Woodsend and SpencerK, however it was still not as good as Experts.

Table 2. System performance using basic statistics and automatic evaluation measures

	tokLen	senLen	senLen#S	#sen	OOV%	FKGL	FKGL#S	BLEU	TERp
input	4.01	16.47	13.82	230	59.95	6.78	5.82	100.00	0.000
Experts	**3.86**	15.98	13.92	238	59.57	6.11	**5.33**	**71.86**	**0.207**
OurSystem	3.91	14.92	**13.80**	260	**58.78**	5.87	5.39	74.31	0.090
Woodsend	3.94	**11.74**	13.82	384	62.20	**5.20**	5.81	86.64	0.075
SpencerK	4.00	16.49	13.84	230	59.53	6.77	5.79	96.20	0.015

#sen gives the total number of simplified sentences. We performed the splitting operation a little more than Experts (22 sentences, nearly 10% of the number of the original sentences). Woodsend is much different from the others with 96 simplified sentences. This indicates that our system generates closer output to that of Experts than that of Woodsend. In fact, Experts only performed the splitting operation on 8 sentences in all of the input ones. This small number indicates that on the story dataset we examine, sentence splitting does not really play a more important role than the other operations.

We now turn to the results of the average number of tokens in one sentence (*senLen*) which roughly reflects the syntactic complexity. All systems except SpencerK usually rendered shorter sentences corresponding to the original ones. This suggests that complex sentences were partly simplified into ones with simpler syntactic structures. SpencerK has the highest *senLen*, showing it performs mostly lexical substitutions. The splitting operation has a significant impact on the results of *senLen* above as this operation reduces the average length, and therefore reduces the syntactic complexity of sentences. To better understand this impact, we measured the average length of simplified sentences that are not generated through the splitting operation (*senLen#S*). The result indicates that our model produced shorter sentences than the others. However, there is no significant difference between all systems in terms of *senLen#S*. Basically, in the absence of the splitting operation, the number of tokens in one sentence is nearly identical for the output of the systems.

As Experts did not produce shorter sentences than the other systems, we learn out that in sentence simplification problem, simplified sentences that are easier to comprehend are not necessarily shorter than the original ones. This notion is also perceived in some previous work (e.g., [13]) or in the SW guidelines[7]

Readability and Translation Assessment The results of the automatic measures are also displayed in Table 2. OOV is the percentage of words that are not mentioned in the Basic English 850 Words list[8], where lower scores mark sentences that are simpler to read. The results of OOV% show that our system used the most basic English words as it scores best, with 58.58% even lower than Experts. SpencerK also has a good score by dint of lexical substitutions. However, this do not assert that Experts is not good because the core elements here are whether basic words used are reasonable or not and whether their nuanced meanings are preserved. In some cases, lexical substitutions make distortions on the original meaning of sentences.

In terms of the Flesch-Kincaid grade level score[9] (FKGL) which estimates readability, higher scores indicate more complex sentences. Generally, this score is under the major impact of the splitting operation since the fact that it significantly reduces when the total number of sentences increases. However, with regard to the FKGL score for simplified sentences that are not generated through the splitting operation (FKGL#S), Experts scores best with 5.33 (lowest), closely followed by our system (5.39). Again, our system is the closest to Experts and altogether outperforms Woodsend (5.81) and SpencerK (5.79). In the absence of splitting operation, Woodsend has the highest FKGL score compared to the other systems. This roughly reflects that this system uses a considerable number of splitting operations. Besides, the original sentences have the highest reading level.

Both BLEU ([10]) and TERp ([11]) are commonly used for automatic machine translation evaluation. In our experiments, BLEU scores the simplified sentences by counting n-gram matches with the original and thus lower BLEU scores are better. TERp measures the number of edit operations (insertion, deletion, substitution, and shift) needed to transform simplified sentences into the original and thus higher TERp scores are better. In both these scores, Experts is the best, followed by our system, Woodsend, and SpencerK. Experts and our system are significantly different from the others. These results also show how close to the gold standard our system is, and once again confirm that our system is more flexible than Woodsend, and SpencerK in using words.

Quality Analysis The results of this experiment indicate that our system is the closest to the gold standard. Woodsend performed many splitting operations, however most of them made the content of the original sentences become more

---

[7] http://simple.wikipedia.org/wiki/Main_Page/Introduction

[8] http://simple.wikipedia.org/wiki/Wikipedia:Basic_English_alphabetical_wordlist

[9] $FKGL = 0.39(\frac{total\ words}{total\ sentences}) + 11.8(\frac{total\ syllables}{total\ words}) - 15.59$

disjointed and less coherent in fact. Table 3 shows examples of simplifications produced by the systems (we ignore the output of SpencerK that is generally similar to the original sentence). The results also verify that our system can create simplifications that significantly reduce the reading difficulty of the input text while maintaining grammaticality and preserving its meaning.

Table 3. Examples of simplifications produced by the systems

input #1	Effy took the microphone and began to sing.
Experts	Effy took the microphone and began to sing.
OurSystem	Effy took the microphone and started to sing.
Woodsend	Effy took the microphone. It started to sing.
input #2	So she had spent her whole camping holiday practising her singing.
Experts	So she had spent her whole camping holiday practising her singing.
OurSystem	So she had spent her holiday practising her singing.
Woodsend	So she had spent her whole camping holiday practising her singing.
input #3	She smiled because she was surprised at the question, and said it was the Mayor's decision.
Experts	She smiled because she was surprised at the question , and said it was because of the Mayor's decision.
OurSystem	She smiled . In this way, she was surprised at the question, and said it was the Mayor's decision.
Woodsend	She smiled because she was surprised at the question. He said it. It was the Mayor 's decision.

Table 4. Average human ratings for the output of the systems

Systems	Simplicity	Grammaticality	Semantic Similarity
Experts	3.78	4.91	4.88
OurSystem	2.99	4.37	4.02
Woodsend	2.74	4.05	3.83
SpencerK	1.52	4.85	4.82

Average Human Ratings Table 4 presents the average human ratings for the output of the four systems. With regard to simplicity score, Experts scores highest with 3.78, followed by our system (2.99) and Woodsend (2.74). SpencerK scores lowest with 1.52 as this baseline solely focuses on lexical substitutions. With regard to grammaticality and semantic similarity, again Experts is rated highest. SpencerK is also rated highly since the fact that it does not change the syntactic structure as well as the main content of the original sentences. Our system continues to score higher than Woodsend.

6 Conclusions

This paper investigates English document simplification task on the domain of children stories. Differently to previous approaches that rely on large-scale parallel corpora derived from MEW and SEW articles, we presented a model which extracts and generalizes simplification rules from a small corpus of 32 stories written by experts and then applies a Bayesian model to select the optimal set of rules to make a text simpler. The corpus is small but reflects different reading comprehension processing skills, making our model promising in generating texts that are suitable for children. Our model is able to perform important simplification operations, namely splitting, dropping, reordering, and substitution. The evaluation shows that our model is close to the gold standard created by experts, and can achieve better results than baseline and state-of-the-art systems in a variety of automatic measures as well as human judgements.

Acknowledgments

We would like to thank The Vietnam National Foundation for Science and Technoloogy Development (NAFOSTED) for financial support.

References

1. Cain, K., Oakhill, J.V., Elbro, C.: The ability to learn new word meanings from context by school-age children with and without language comprehension difficulties. Journal of Child Language **30** (2003) 681–694
2. Cain, K.: Making sense of text : skills that support text comprehension and its development. Perspectives on Language and Literacy **35** (2009) 11–14
3. Carroll, J., Minnen, G., Pearce, D., Canning, Y., Devlin, S., Tait, J.: Simplifying text for language-impaired readers. In: In Proceedings of EACL (1999) 269–270
4. Ehrlich, M.f., Remond, M., Tardieu, H.: Processing of anaphoric devices in young skilled and less skilled comprehenders : Differences in metacognitive monitoring. Reading **11** (1999) 29–63
5. Inui, K., Fujita, A., Takahashi, T., Iida, R., Iwakura, T.: Text simplification for reading assistance: A project note. In: In Proceedings of IWP (2003) 9–16
6. Kaji, N., Kawahara, D., Kurohash, S., Sato, S.: Verb paraphrase based on case frame alignment. In: Proceedings of ACL (2002) 215–222
7. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of ACL (2003) 423–430
8. Oakhill, J., Cain, K.: Inference making and its relation to comprehension failure. Reading and Writing An Interdisciplinary Journal **11** (1999) 489–503
9. Oakhill, J., Hartt, J., Samols, D.: Levels of comprehension monitoring and working memory in good and poor comprehenders. Reading and Writing **18** (2005) 657–686
10. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of ACL (2002) 311–318
11. Snover, M.G., Madnani, N., Dorr, B., Schwartz, R.: Ter-plus: Paraphrase, semantic, and alignment enhancements to translation edit rate (2010)
12. Wauters, L.N., Bon, W.H.J., Tellings, A.E.J.M.: Reading comprehension of dutch deaf children. Reading and Writing **19** (2006) 49–76
13. Woodsend, K., Lapata, M.: Learning to simplify sentences with quasi-synchronous grammar and integer programming. In: Proceedings of EMNLP (2011) 409–420
14. Wubben, S., van den Bosch, A., Krahmer, E.: Sentence simplification by monolingual machine translation. In: Proceedings of ACL (2012) 1015–1024
15. Yatskar, M., Pang, B., Danescu-Niculescu-Mizil, C., Lee, L.: For the sake of simplicity: unsupervised extraction of lexical simplifications from wikipedia. In: Proceedings of NAACL-HLT (2010) 365–368
16. Zhu, Z., Bernhard, D., Gurevych, I.: A monolingual tree-based translation model for sentence simplification. In: Proceedings of COLING (2010) 1353–1361