

Text Simplification: A Survey

Lijun Feng

March 4, 2008

Contents

1	Introduction	2
2	Motivations and Needs for Text Simplification	3
3	Linguistic Issues posed by Text Simplification	5
4	Current State of the Art	7
4.1	Text simplification systems for general NLP applications	9
4.1.1	Chandrasekar et al.'s early work on syntactic simplification .	9
4.1.2	Siddharthan's continuation on syntactic simplification	11
4.1.3	Klebanov et al.'s Easy Access Sentences generation system for information-seeking applications	14
4.2	Text simplification systems for human readers	16
4.2.1	Systems for users with various language disabilities	16
4.2.2	System for people without disabilities who have low literacy	21
4.2.3	System for language teachers, children and adult second lan- guage learners	23
4.3	Summary of the systems	24
5	Conclusions	25
6	Future prospects	26

1 Introduction

Natural human languages often contain complex compound constructions. Long and complex sentences are not only difficult for many human readers to process, they prove to be a stumbling block for automatic systems which rely on natural language input (Chandrasekar et al., 1996). In order to ease the task of processing such sentences, it is desirable to simplify them grammatically and structurally into shorter and simpler sentences while preserving the meaning and information contained within them. Text simplification is an NLP task that tries to simplify natural language texts in such a fashion.

Text simplification can be very useful and important for many automatic NLP applications, such as parsing, machine translation, text summarization, and information retrieval. For human readers, the need for text simplification is varied depending on the target population: the range and level of desired simplification of natural language texts for people with disabilities can be quite different from these for automatic systems or people without disabilities. Despite text simplification being of practical use in many applications, it is not a well explored field and has not received much attention in NLP research until the last decade. In 1996, Chandrasekar et al. at University of Pennsylvania first provided a formalism of syntactic simplification motivated by the need to improve the performance of their full parser. They not only developed techniques to analyze the syntactic information of input text efficiently, but more importantly, they established a formalism of dependency-tree based syntactic transformation. They also raised many linguistic questions posed by the task of text simplification that were not answered in their work. Since then there has been increasing research interest in using NLP technologies for development of broad coverage text simplification systems, many have focused on assisting human readers, some of which serve as assistive technology for people with various language disabilities.

In the past decade, a number of user- and task-oriented text simplification systems have been developed worldwide, notably in United Kingdom, Japan, and United States. To better understand these systems and the research that has been done in this field, the following literature survey presents the current state of the art and reports what major linguistic issues are there to be addressed in designing such a system. This paper focusses on what has been achieved technically in the field, what technologies related to text simplification are made available today, and what still remains as open research topics for future work. The remainder of the paper is organized as follows. Section 2 discusses the motivations and needs for text simplification. Section 3 discusses linguistic issues posed by the task of

text simplification. Section 4 investigates the development of current state of the art. Section 5 provides a summary of technical development of text simplification. Section 6 describes future prospects of the field.

2 Motivations and Needs for Text Simplification

Many NLP systems relying on natural language input have difficulties in processing long sentences with complex syntactic structures. To ease the task, text simplification can be a useful and effective preprocessing tool by greatly simplifying the grammatical and syntactic structures of these sentences. This was exactly the initial motivation for Chandrasekar et al.'s (1996) early work on text simplification. The problem they faced were the challenges posed by long and complex sentences to their full parser. As the syntactic structures of a sentence become more complex, the number of possible parses increases, which inevitably leads to increased ambiguity and a greater likelihood of incorrect parsing (Chandrasekar et al., 1996). Intuitively, if these sentences are transformed into shorter and simpler ones first before they are fed to the full parser, the level of ambiguity will be reduced and the performance of the parser is expected to become more robust and less error-prone.

Besides parsing, automatic NLP applications such as machine translation, information retrieval, and text summarization can also benefit from text simplification. Machine translation, for example, faces similar problems to a full parser. The performance of the system deteriorates rapidly when input sentences become long and complex. More mistakes are likely to occur due to increased ambiguity contained in long sentences; shorter and simpler sentences would be easier and more likely to be translated correctly. Simplifying complex sentences into shorter ones would also reduce the amount of information contained per sentence. This can make the search for relevant factual information more robust in information retrieval. For the same reason, text summarization systems based on sentence extraction can also benefit from simplified texts.

Text simplification can also be very useful for human readers, especially for people with various language disabilities. Aphasia, for example, is a loss of ability to produce and/or comprehend language due to injury to brain areas that are specialized for certain language functions. There are a variety of language problems associated with aphasia depending on factors such as the extent and location of brain injury, aphasia type, and literacy level before aphasia. These problems can be of a lexical or syntactic nature. Providing lexical simplification and reducing linguistic complexity of written texts would make them more comprehensible to

these individuals.

Various hearing impairments also have a significant impact on people's ability to understand spoken and written language. The problem with language comprehension can be more severe when the impairment exists before the individual (not born into a signing family) has acquired speech and language. Due to a lack of exposure to accessible language input during the critical language-acquisition years of childhood, many of these individuals have low levels of literacy. Tests conducted by the Gallaudet Research Institute to assess the reading achievement of deaf and hard of hearing students show that the average reading comprehension of the 17-year-olds and the 18-year-olds deaf and hard of hearing students without mental retardation corresponds to about a fourth-grade level for hearing students (Holt et al., 1997). Because of their low levels of literacy, texts that are written for people at that age level without language disabilities can be challenging for them to understand. Simplifying written texts to lower reading levels would help them access information and materials that are useful and interesting to them.

People with intellectual disabilities also have problems in reading. A recent study conducted by Jones et al. (2006) assessing the reading comprehension of adults with mild mental retardation (IQ range 50-75 inclusive) reported that the average reading ages of the subjects are below average 7-year-old readers without disabilities. For these individuals, reading materials of interest at their corresponding low reading levels are hard to find. This is because reading materials at lower reading levels are typically written for children, and texts written for adults without disabilities often require high level of linguistic skills and sufficient real world knowledge that these individuals often lack. Text simplification could help these individuals feel more connected to the community by reading and understanding what is going on in the world surrounding them.

Simplifying written texts lexically and syntactically to lower reading levels would also make texts more readable for children, adult native speakers with low literacy, and adult second language (L2) learners at beginning and intermediate levels. Such a text simplification system can also save language teachers the time they may have spent adapting teaching materials to appropriate reading levels manually.

There has been some effort in making texts, especially on-line information more accessible to the user groups described above. Simple English Wikipedia, for example, aims to serve children, English language learners, people partially literate in English, and those with learning disabilities by providing shorter and simpler versions of articles carefully written in basic English. These handwritten articles correspond to articles that appear in the traditional English Wikipedia. Unfortu-

nately, there are only a limited number of them. If there is a text simplification system that adapts and simplifies the original articles automatically to their simpler versions, more information can be shared and accessed by this group of users.

3 Linguistic Issues posed by Text Simplification

Text simplification is a complicated natural language processing task, which has to deal with many different linguistic issues depending on whether the simplification task is at lexical, syntactic, or discourse level. This section discusses various linguistic issues arising at these three different levels in the development of text simplification systems. A brief overview of possible resources, tools, and techniques that can be used to solve these problems is also provided. Additional technical details will be discussed in section 4 when individual systems are introduced and analyzed in more detail.

The task of **lexical** simplification often involves replacing difficult words with their simpler synonyms. Two questions to ask here are: how to identify difficult words? And how to replace them with simpler synonyms? The common approach to the first question is to use word frequency count. Word frequency here means the frequency of the word in typical English usage, not in some specific piece of text. A word with lower frequency indicates that the word is not commonly used, therefore it is likely a more difficult word. Another approach is to treat each content word as being equally difficult and let user decide which words need to be simplified (Devlin, 1999). To replace a low-frequency word with a more common synonym, WordNet is a widely used resource. WordNet is an electronic semantic lexicon for English language developed by Miller et al. (1993) at Princeton University. It groups English nouns, adjective, verbs and adverbs into sets of synonyms called *synsets*, each representing one sense or meaning of a word. PSET, HAPPI and ETS's ATA v.1.0, three text simplification systems to be discussed in section 4, all used WordNet for lexical simplification. Difficult words can also be simplified by paraphrasing using a pre-defined vocabulary list (Inui et al., 2003) or their dictionary definitions (Kaji et al., 2002).

Syntactic simplification is a much more complex task than lexical simplification, because input texts need to be linguistically analyzed to produce detailed tree-structure representations required by syntax transformation. Syntactic simplification typically involves two steps of work: first syntactic constructs that can be simplified need to be identified, and then a set of simplification rules, either hand-crafted or automatically generated, are applied to simplify them. There are

several factors to be considered when trying to identify simplifiable constructs. Past research on syntactic simplification has focused mainly on splitting conjoined (coordinated and subordinated) sentences, extracting relative clauses and embedded noun phrases, and converting passive voice to active voice. To identify these constructs, particular linguistic features need to be specified and marked. These features include coordinated and subordinated conjunctions, relative pronouns, clause or phrase boundaries, and noun phrases to which clauses are attached. More analysis is needed to provide required information when simplifying these identified constructs. For example, to simplify a sentence with an embedded relative clause, the noun phrase to which the relative clause is attached to needs to be annotated with number information such as singular or plural, the dependency relationship of word groups contained within the sentence needs to be analyzed in order to provide basic information such as subject, verb, and object to form a new simpler sentence. To convert a passive construct into an active form, the agent of the action needs to be identified, and verbs need to be annotated with grammatical information such as voice, tense and aspect.

Many NLP technologies are available to perform the syntactic analysis required for simplification. For example, part-of-speech (PoS) tagging, parsing, pattern matching, and punctuation marks are often used to identify simplifiable complex constructs. Parsing is used to analyze dependency relationship of word groups in sentences. Because a full parser is less robust and more error-prone when input sentences are long and complex, shallow parsing is often preferred. Shallow parsing, also called “chunking,” provides an analysis of a sentence at a coarse granularity. Word groups such as noun and verb phrases are chunked, but the internal structure of chunks is not analyzed. There are various shallow parsing techniques available. For example, in the text simplification systems to be discussed in section 4, Chandrasekar et al (1996, 1997) used Finite State Grammar (FSG), Lexicalized Tree Adjoining Grammar (LTAG) and Supertagging techniques to identify noun and verb chunks; Carroll et al. used a probabilistic LR parser (Briscoe and Carroll, 1995) to perform shallow analysis on the input texts. Chunking is very efficient in defining the syntax of a sentence because it provides an analysis with the necessary level of detail but does not waste time on analyzing internal structure of chunks. Moreover, chunking also helps structure simplification rules, especially in corpus-based analysis. When a sentence and its simplified version are both represented by dependency trees after chunking, simplification rules can be generated either automatically (Chandrasekar and Srinivas, 1997) or manually (Inui et al., 2003) by observing the tree-to-tree transformation patterns. For details of the

above mentioned techniques, see section 4 when individual systems that deployed such techniques are addressed.

At the **discourse** level, the major linguistic issue is how to maintain coherence and cohesion of simplified text. Syntactic simplification is often done one sentence at a time, and interactions across sentences are not considered. This inevitably leads to some lack of cohesion in the resulting text for the following reasons. For example, when passive voice is changed into active voice, the order in which the noun or pronoun is introduced into the discourse is changed. This might break the correct link of pronouns and its referring expressions across sentences. Similar problems can happen when a relative clause is extracted and the relative pronoun is replaced by a noun or pronoun to form a new sentence. Besides possible broken anaphoric links, it is difficult to decide the order of resulting simplified sentences so as to maintain the rhetorical relations contained in the original sentences. Siddharthan (2002) offered a solution to sentence ordering using a strategy based on entity salience ranking. To fix broken anaphoric links, both Siddharthan and Copestake (2002) and Canning (2002) provided algorithms to resolve pronouns.

It is worth pointing out that little study has been done on **semantic** simplification in this field. It is partly because for many systems designed for specific applications and/or particular target users, semantic simplification has not become an important factor for the researchers to consider. Moreover, it is technically quite challenging to perform the task of semantic simplification because semantic transformations and simplifications would generally require deep analysis of the input text and sophisticated methods for knowledge representation. Both of these can be computationally costly and in practice difficult to carry out.

4 Current State of the Art

Current text simplification systems fall in two categories: those aimed to be a useful tool to improve the performance of other automatic NLP applications (such as parsing, machine translation, text summarization, and information retrieval), and those intended to serve as reading assistance for human readers. Depending on the specific purposes of each system, the linguistic issues addressed are at diverse levels. For example, a full parser or a machine translation system often has difficulty with long and complex sentences because of increased syntactical ambiguity contained within them. For these type of applications, syntactic simplification is expected by intuition to have significant impact on the improvement of system's performance. In systems designed for human readers, linguistic issues at all levels

become important for text simplification – depending on whether the target users are children, adults with low literacy, L2 learners, or people with various language disabilities. Generally speaking, these issues can be categorized at three levels: lexical, syntactic and discourse level. To better understand the current state-of-the-art text simplification systems and appreciate the differences among them, all systems are first broadly categorized into one of the above mentioned two groups, and then three factors are used to organize and illustrate each of them in a subsection. These three factors are:

- purpose of the system,
- approaches to the linguistic issues that are key to the system,
- evaluation of the system's performance.

At the end of each subsection, a brief critique of that system is made.

To the author's knowledge, currently, there are eight notable text simplification systems developed in United Kingdom, Japan, and United States, among which some are still undergoing refinement and/or new development. The following three systems have been developed to improve automatic NLP applications (the first group):

- Chandrasekar et al.'s (1996) syntactic simplification system, developed at UPenn to improve the performance of their full parser,
- Siddharthan's (2003) text simplification system, developed at University of Cambridge, UK, intended for both general NLP applications and human readers,
- Klebanov's (2004) small Easy Access Sentence (EAS) transformation system, developed to improve the performance of information retrieval.

And the following five text simplification systems have been developed to provide human readers with reading assistance; they can be sub-categorized in three groups:

- systems developed for users with various language disabilities:
 - PSET (1999), developed at University of Sunderland, UK, for people with aphasia
 - HAPPI (2006), based on PSET and developed in UK for people with aphasia

- KURA (2003), developed in Japan for people who are deaf
- system developed for adult native speakers with low literacy and math skills:
 - SkillSum (2005), developed at UK
- system developed for language teachers, children and adults as second language learners
 - ATA v.1.0 (2007), developed by ETS for teachers and English language learners (ELLs)

4.1 Text simplification systems for general NLP applications

4.1.1 Chandrasekar et al.'s early work on syntactic simplification

Chandrasekar et al.'s research on text simplification was motivated by the need to improve the performance of their full parser. Long and complex sentences pose big problems for a full parser, because the parsing ambiguity increases with sentence's length and complexity. The performance of the full parser is expected to be improved greatly if input sentences are short and simple with less ambiguity.

Chandrasekar et al.'s research focuses on syntactic simplification. They viewed text simplification as a two stage process, analysis followed by transformation. The analysis stage provides a structural representation of the input text, the transformation stage uses this representation to identify simplifiable constructs and apply simplification rules to simplify them. In particular, they defined a set of general articulation points where sentences can be split for simplification. These points include punctuation marks, subordination and coordinating conjunctions, relative pronouns, and the beginnings and ends of clauses and phrases. Based on the articulation points, a set of transformation rules are defined which map from given sentence patterns to the simplified sentence patterns (Chandrasekar et al., 1996).

Chandrasekar et al. addressed the task of text simplification in two approaches. In their first approach, they provided fast and robust alternatives to analyzing input text. Intuitively, a full parser would be ideal to provide complete structural and dependency information for each input sentence. The drawback is that a full parser is not robust and is prone to failure, especially on complex sentences, which are central targets of their text simplification. To overcome the limitations of full parsers, Chandrasekar et al. (1996) used a Finite State Grammar and a dependency based model in the analysis stage. Both approaches differ from full parsers in that, instead of a complete structural representation, they provide syntactical information of a sentence at a coarser level. In the FSG approach, a sentence is considered

to be composed of a sequence of chunks. Each chunk identified by a FSG is a word group consisting of a noun or verb phrase with some attached modifiers (number information for a noun phrase, tense, voice and aspect information for a verb phrase). The output provided by the FSG approach is a sequence of noun and verb phrases without any hierarchical structure. A set of ordered simplification rules are then applied to simplify the chunked sentences. In the dependency based model, they used Lexicalized Tree Adjoining Grammar (LTAG) and the Supertagging techniques to provide richer syntactical information for text simplification. The primitive elements of LTAG are elementary trees, which localize dependencies by requiring that only the dependent elements be present within the same tree. As a result, a lexical item may be associated with more than one elementary tree. The Supertagging techniques are used to assign appropriate elementary trees to each word of input text. The disambiguated elementary trees are then combined by substitution and adjunction. The elementary trees associated with each word provides the constituent structure information. A Lightweight Dependency Analyzer is then used to establish the dependency links among the words of the sentence. To simplify input text, the constituent information contained in a supertag is used to determine whether a clausal constituent is present, and the dependency links among words are used to identify the span of the clause.

At the transformation stage, a variety of rules are needed to simplify input text. In their early approach, the simplification rules were manually made; however, hand-crafting rules is time-consuming and impractical. While a set of general simplification rules may be made applicable across domains, specific domains require specific rules. To overcome this problem, Chandrasekar and Srinivas (1997) developed an algorithm in their second approach to automatically induce simplification from training data. The training data consisted of a set of input sentences paired with their respective manually simplified versions. All the sentences were processed using Lightweight Dependency Analyzer. The resulting dependency representations of the paired sentences were chunked into phrasal structures with dependency information maintained. The chunked dependency tree of the complex sentence and that of the simplified ones were compared. Simplification rules were induced by computing tree-to-trees transformations that are required to convert the complex sentences to the simplified ones.

The overall performance of their text simplification system was not evaluated. Two alternative sub-components of their system for performing chunking (FSG and DSM) were evaluated on a small-scaled corpus of newswire data. The DSM model outperformed the FSG-based model by correctly recovering 25 out of 28

relative clauses and 14 of 14 appositives, while the FSG model only recovered 17 relative clauses and 3 appositives on the same data. Unfortunately, the quality of their transformation component was not measured.

Chandrasekar et al.'s early work on syntactical simplification and automatic induction of simplification rules made an important advance in text simplification. However, they also raised more questions than they answered. Their system only processed one sentence at a time; sentential interactions were not considered. This can lead to some lack of coherence in the resulting simplified text. To maintain the coherence of simplified text, more linguistic issues still need to be addressed, such as deciding the relative order of the simplified sentences, choosing appropriate referring or gap-filling expressions, and selecting correct tense while forming a new sentence.

4.1.2 Siddharthan's continuation on syntactic simplification

Chandrasekar et al. raised many discourse level questions on text simplification but didn't address any of them in their work. Siddharthan's research followed up on these questions while aiming to provide a more complete formalism on syntactic simplification.

Siddharthan's PhD thesis (Siddharthan, 2003) focuses on syntactic simplification with an emphasis on retaining discourse cohesion of the rewritten text. The architecture of his system consists of three stages. In addition to Chandrasekar and Srinivas' (1996) two-staged architecture – analysis and transformation – a third stage named as *regeneration* stage (Siddharthan, 2002) was added to the system to deal with discourse level issues raised by syntactic simplification.

Siddharthan's approach to syntactic simplification was similar to Chandrasekar et al.'s earlier work. Since the performance of a full parser is less robust and computationally more expensive when the input sentences are long and complex, shallow analysis of input text is preferred. While Chandrasekar et al. used a partial parser for this, Siddharthan chose LT Text Tokenization Toolkit (LT TTT) for text segmentation, part-of-speech tagging, and noun chunking. The LT TTT provides a set of tools that can be used to tokenize text by introducing XML mark-up. Text can be processed at either the character level or at the level of XML elements. The toolkit comes with built-in rule sets to mark-up words, sentences, and paragraphs; it can also perform basic chunking into noun phrases and verb groups (Grover et al., 2000). Pattern matching techniques were then used on the preprocessed input text to identify syntactic structures that can be simplified. These simplifiable complex constructs include relative clauses, coordinated and subordinated clauses,

appositive phrases, participial phrases, and passive voices. In his thesis, the focus was on the treatment of restrictive and non-restrictive relative clauses, conjoined clauses, and appositive phrases. The analysis stage outputs specifications of structural representation of a sentence required by next two stages. These specifications include POS tagged words, marked relative clause and appositive boundaries, marked noun phrases and phrasal attachments, and marked and annotated elementary noun groups with grammatical information. A set of hand-crafted rules were applied in the transformation stage to simplify the marked constructs.

Because the process of syntactic simplification proceeded one sentence at a time during the transformation stage, various problems related to the lack of discourse cohesion (such as duplicated noun phrases, broken pronominal and anaphoric links, etc.) can arise to make the resulting text hard to read. To avoid duplicated noun phrases and fix broken links of anaphora resulted from syntactic simplification, referring expressions were generated for the second time. According to Rhetorical Structure Theory (RST) (Mann and Thompson, 1987), a coherent text should not have gaps in it. Every text span has a purpose and is related to the rest of the text by means of some relation (such as concession, condition, motivation, elaboration, etc.). For example, non-restrictive relative clauses and appositive phrases have an elaboration relationship with the noun phrases they are attached to. When simplifying these constructs, the elaboration relationship contained in the original sentences should be preserved. To solve these kind of problems, the author developed a sentence ordering algorithm to preserve the rhetorical relation contained in the original sentence. To express rhetorical relations more clearly, cue words indicating these relations were restricted to simple and common ones. For example, *so* and *but* are used instead of *hence* and *however*.

Siddharthan's research provided a theory of syntactic simplification that formalizes the interactions taking place between syntax and discourse during the simplification process. The highlight of the thesis is that it addressed various discourse level problems resulting from the syntactic simplification process which have not been dealt with much before. The proposed techniques to these problems can be applicable to other NLP generation applications, such as text summarization, because similar discourse level problems occur during text transformation/generation there as well. The use of automatic syntactic simplification was also explored later to improve content selection in multi-document summarization (Siddharthan et al., 2004). It was shown that simplifying parentheticals by removing relative clauses and appositives resulted in improved sentence clustering.

The three-staged system is pipelined, each module was evaluated indepen-

dently, and a holistic evaluation of the system was provided in the end. The system was evaluated in the following three aspects: correctness, readability, and the level of simplification achieved. Three human subjects were asked to evaluate the correctness of the regenerated sentences of 95 examples from the Guardian news corpus in terms of grammaticality and preservation of meaning and text cohesion. They were asked to answer *yes* or *no* to the grammaticality questions, and use 0 (meaning altering) or non-0 (1 or 2 or 3 to indicate different level of meaning preserving) to rate meaning preservation, and use 0, 1, 2, and 3 to rate coherence (with 0 or 1 indicating major coherence disruptions, 2 indicating a minor reduction of coherence, and 3 meaning no loss of coherence). The author reported 80% of unanimous vote and 94.7% of majority vote on grammaticality, and 85.3% of unanimous vote and 94.7% of majority vote on meaning preservation. On coherence, 41% of examples were rated 3, and 75% of them were rated above 2 on average.

The author used the Flesch Reading Ease formula to evaluate readability and the level of simplification achieved of text generated by the system. The Flesch Reading Ease formula is a readability test that uses word length and sentence length as core measures to indicate comprehension difficulty of a written text. Higher scores indicate material that is easier to read; lower numbers mark harder-to-read passages. The author reported that the Flesch reading ease score for the original corpus of 15 Guardian news reports is 42.0 (suitable for a reading age of 19.7). And after syntactic simplification, the score increased to 50.1 (suitable for reading age of 17.8).

Since the author's primary goal focused on syntactic simplification and did not provide a treatment for lexical simplification, this may explain why the reading difficulty of the simplified text was not reduced significantly after syntactic simplification. The reason for this result may also have something to do with the evaluation metric he chose. Although the Flesch Formula has gained wide acceptance and it is easy to calculate, it relies on shallow features of text, such as sentence length (measured in words) and word length (measured in syllables) (Flesch, 1979). The drawbacks of this formula, as recent researchers (Schwarm and Ostendorf, 2005) have pointed out, are that sentence length is not an accurate measure of syntactic complexity and that the syllable count does not necessarily indicate the difficulty of a word.

The insignificant improvement of the readability produced by Siddharthan's text simplification system could also lie in the fact that the author had overly broad goals when designing the system. Or more precisely, his system was designed without any specific applications or target user groups in mind. The system was

motivated by the author's belief that syntactic simplification could be useful in various applications and may help make written texts easier for humans to read. It did not take specific needs and/or characteristics of particular application and target users into consideration.

4.1.3 Klebanov et al.'s Easy Access Sentences generation system for information-seeking applications

Text simplification can be a useful preprocessing tool for many NLP applications, such as machine translation, text summarization, and information retrieval. The particular aspects of sentence complexity that need to be addressed during the preprocessing may be different depending on each system's goals. To preprocess input text for information-seeking applications, such as summarization, information retrieval and extraction, the question to ask is what makes finding information in a text easy for a computer. Klebanov et al. (2004) claimed that easy access sentences (EASes) are easy for a computer to find information. A sentence is an easy access sentence (EAS) if it satisfies the following requirements:

- **Sentence:** it is a grammatical sentence;
- **Single Verb:** it has one finite verb;
- **Information Maintenance:** it does not make any claims that were not present, explicitly or implicitly, in input text;
- **Named Entities:** if a sentence satisfies all the previous three requirements, then the more named entities it contains, the better EAS it is.

Here the named entities refers to full names of entities, for example, *Bill Clinton* is preferred to pronoun *he*, or partial or indirect reference to the entity such as *Mr Clinton* or *the former president of the United States*.

To construct EASes from input text satisfying the above requirements, a list of linguistic issues need to be addressed, including: resolution of pronouns and anaphoric references, assigning correct tense to the verbs that depend on the governing verbs or other elements, deciding the implicit subject of the verb in relative clauses, etc. To solve these problems, they developed an algorithm to construct EASes automatically from input text. First, person names are identified using BBN's Identifinder (software that identifies significant documents and/or locates the most important information within them) (Bikel et al., 1999), and dependency structures of the input text are derived using MINIPAR. MINIPAR is an

English parser which represents the grammar as a network. The nodes represent grammatical categories and the links represent types of syntactic (dependency) relationships (Lin, 1998b). Sentences that do not contain factual information are called *semantically problematic environments*, they are excluded from the extraction of EASes. These semantically problematic environments can be detected by conditional markers, modal verbs and a pre-constructed list of other verbs such as *attempt, desire, request*, etc.

The algorithm proceeds verb-wise, trying to construct an EAS with each verb as its single finite verb. For every verb V ,

1. if an EAS is in a semantically problematic environment, skip V and go to next verb.
2. If V is not finite, go up the dependency structure and assign tense of the closest tensed governor verb.
3. Collect V 's dependents. If the deep subject of the verb is an empty string, follow the antecedence links provided by dependency structures to get the subject. If this does not help, the subject of the clause to which the current clause is attached is assigned as the default subject to the verb.
4. Try to increase the number of Named Entities among V 's dependents. Salience based anaphora resolution is implemented to resolve just pronouns *his, him, he, she, her*.
5. Output an EAS with V and its dependents.

The authors used a test set of 123 sentences from 10 newswire articles to evaluate the precision of the EAS construction algorithm. Out of 123 sentences, 68 (55%) were reported to have passed the EAS requirements. To evaluate the recall of the system, five people were asked to generate single verb sentence from an extract (7-sentence text) from Bertrand Russel's biography. Two other humans were then asked to evaluate the single verb sentences produced by these three people. Thirty-three sentences marked correct by both human judges were selected to form the gold standard set. The sentences generated by the system from the same extract were compared to the gold standard set. They reported approximately 30% recall. Their error analysis showed that many mistakes were mainly due to the dependency parser (MINIPAR) they used: the parser produced much incorrect dependency information.

EASes bring related dispersed information closer in single-verb sentences with Named Entities, which can be effectively used by information-seeking applications. Stylistically, EASes constructed from input text may appear somewhat tedious for most human readers to read, but they could be useful for people with specific kinds of memory or cognitive problems. For example, PSET, a text simplification system designed for people with aphasia, which will be discussed soon, substituted pronouns with their referents to 'jog' aphasic readers' memory.

4.2 Text simplification systems for human readers

4.2.1 Systems for users with various language disabilities

PSET

PSET (Practical Simplification of English Text) was a collaborative project between the University of Sunderland and Sussex between 1996 and 2000 (Carroll et al., 1998, 1999; Devlin, 1999; Canning, 2002). The goal of the project was to develop a automatic text simplification system to help aphasic readers understand English newspaper text.

Aphasia is a loss of ability to produce and/or comprehend language due to injury to brain areas that are specialized for these language functions. There are various language problems associated with aphasia depending on factors such as the extent and location of brain injury, aphasia type, and literacy level before aphasia. Aphasic people may encounter various problems in reading. These problems can be of a lexical or syntactic nature. Generally speaking, they often have difficulties with infrequent words, long sentences with complex structures, syntactic constructs that deviate from canonical subject-verb-object order, for example, the use of passive voice (Caplan and Hilderbrandt, 1988). Typical newspaper text often has a compact summary-like first paragraph, long and complex sentences, and frequent use of passive voice (Carroll et al., 1999). Because of these characteristics, it can be quite challenging for people with aphasia to understand these type of text.

The PSET project developed various text simplifications options to help aphasic readers understand newspaper text. The early stage of PSET was Devlin's work on lexical simplification of newspaper text. The idea was to replace content words with simpler synonyms using WordNet (Miller et al., 1993) as a lexicon resource. For each content word of the newspaper text, a set of synonyms was retrieved from WordNet. Then the original word together with a percentage of the synonyms were looked up in the Oxford Psycholinguistic Database (Quinlan, 1992) for the corresponding Kucera-Francis frequencies. The Kucera-Francis frequency is the word frequency count based on statistical analysis on what is known today as the

Brown Corpus. The corpus was compiled by Henry Kucera and W. Nelson Francis and consists of 500 samples, distributed across 15 genres, three of which are 'Press' (Kucera and Francis, 1967). The one with the highest frequency was selected to replace the original word.

Although Devlin's research covered both lexical and syntactic simplification, in the practical developments only lexical simplification was covered. PSET was developed directly from Devlin's work, the research concentrated mainly on syntactic simplification. In particular, their syntactic simplification focused on splitting long and conjoined sentences into shorter ones and changing passive voice into active voice. PSET's approach to syntactic simplification was similar to that of Chandrasekar et al.'s. The architecture consisted of a sequence of analyzers followed by a syntactic and a lexical simplifier. In the analysis stage, each word of input text was PoS tagged, and the morphological information of the word was analyzed. The tagged input was then shallow-parsed by the parser using a featured-based unification grammar of PoS and punctuation tags coupled with probabilistic LR disambiguation (Carroll et al., 1999). Then a set of ordered simplification rules were iteratively and repeatedly applied to the phrase marker trees produced by the analyzer until further simplification was impossible.

To maintain coherence of the simplified text, anaphoras contained in the sentences that had been split were resolved and replaced by their referents. Their anaphora resolution was based on CogNIAC (Baldwin, 1997).

The performance of PSET's syntactic simplification was evaluated on 100 news articles (Canning, 2002), out of which 75 coordinated sentences and 33 agentive passive constructions were identified. Canning (2002) reported an accuracy of 75% for simplifying subordinated sentences and an accuracy of only 55% for converting passive constructs into grammatically correct active voice. To evaluate the effect of pronoun replacement on reading comprehension, an experiment was designed to determine whether replacing cross-sentence anaphoric pronouns with their antecedents would reduce reading time for aphasic readers and/or help them to correctly identify the referent for the pronoun. Six texts, each consists of two sentences, were selected from the *Sunderland Echo* newspaper to include three types of anaphoric pronouns in the second sentence: singular, plural and possessive. Based on these six texts, a resolved version of each was produced with the pronouns replaced by their referents. Following each text was a question designed to determine the reader's understanding of the referent for the pronoun. In total twelve texts (six original unresolved and six resolved) were presented to sixteen anaphoric participants. The reading times measured on each subject starts from reading the texts

and questions and ends until all questions were answered. The authors reported that anaphoric pronoun replacement did have some impact on reading comprehension: the statistical analysis showed that on average reading times were shorter (about 20%) for resolved versions of text; and on average the score on question answering tests for resolved versions of text improved by 7%.

Based on Canning's (2002) evaluation results, PSET's major contribution to aphasic readers was its anaphora replacement, which resolved referents that many aphasic people might have difficulties in inferring. The syntactic simplification does not seem to have had any significant impact on improving reading comprehension due to the fact that only a very small set of sentences (one per article) in the corpus were simplified. This limited application of their syntactic transformation rules does not come as a surprise, because the types of specific syntactic constructs the PSET project aimed to simplify are very limited: it only tried to split conjoined sentences and convert passive voice into active voice. And the mixed success of the latter can be explained and justified by the fact that many passive constructs are often either agentless, or deeply embedded in sentences, which makes it difficult to recover the agent. Sentence complexity can have a wide range of dimensions. To find out which aspects of sentence complexity challenges the target user's reading comprehension most in the target text domain, empirical experiments in addition to theoretical research should be useful to identify more practical directions.

HAPPI

The HAPPI (Helping Aphasic People Process Online Information) project is a web based text simplification system that followed directly from Devlin's earlier work on lexical simplification of newspaper text (Devlin and Unthank, 2006). With more technologies made available in the intervening years, Devlin revisited the topic seven years later to add more alternatives and refinements to the system. The goal of HAPPI is to provide aphasic readers alternative means to access and comprehend online information by simplifying the language that they find most difficult. In particular, it aims to make it possible for end users to select the difficult words that they wish to be simplified. WordNet is still used in HAPPI as a lexicon databank for synonyms. What is new is that the underlying tool is replaced by the MySQL port of WordNet (Princeton University) developed by Android Technologies. In addition, online speech tools and large scale online image databanks are employed to refine the system. By clicking on difficult words, images representing the concept of the words can be shown to the end user, the user can also hear the words read aloud. The authors argue that these three alternative approaches

to the word level simplification, according to the studies in psycholinguistics and aphasiology, help 'jog the memory' of aphasia readers and to increase their comprehension (Devlin and Unthank, 2006).

The HAPPI project took advantage of new NLP technologies and added more options of lexical simplification for the end users. However, replacing difficult words with their simpler synonyms is useful in lexical support, it does not necessarily reduce the reading difficulty of an input text, which is often posed by complex syntactic and grammatical structures. Although the authors mentioned that HAPPI was built on resources and tools that had been used in PSET, they were not clear about whether the syntactic simplification tool of PSET was used in HAPPI in addition to its lexical simplification. Considering that PSET's syntactic simplification was too narrow in scope and not successful in practice because of the limited syntactic constructs (conjoined sentences and passive voice) it attempted to simplify, one can imagine that it wouldn't have a significant impact on HAPPI even if it was used there.

The other potential problem with Devlin's approach to lexical simplification is that substituting infrequent words with common words can result in increased polysemous words. The semantic meaning of the original word may not be preserved after substitution, and this might add a new level of comprehension complexity to the text for readers with disabilities. There is an opportunity for research on word sense disambiguation to provide a solution for this potential problem.

KURA

Deaf people who for various reasons may have difficulty reading written language text may also benefit from a text simplification system similar to what has been developed for aphasic people. In addition, these users may have unique literacy impairments that can be addressed by NLP technology. KURA is such a system developed in Japan for adult deaf users.

In 2003, Japanese researchers (Inui et al., 2003) reported on a text simplification project designed for deaf users. The goal of the project was to develop a text simplification system as reading assistance for deaf people, in particular, for deaf students at (junior-) high schools with language disabilities. The system can process an input text one sentence at a time in order to lexically and structurally paraphrase the given text to make it simpler and more comprehensible for language-impaired deaf students.

Their approaches to building such a text simplification system are empirical and corpus-based. The development of technologies required by the system were

driven and guided by four issues, which are readability assessment, paraphrase acquisition, paraphrase representation, and post-transfer error detection.

KURA is a text simplification system that generates multiple paraphrases for a given input sentence. If there is a statistically reliable ranking model that can assess the readability of each paraphrase automatically, then this ranking model can guide the system to choose the most readable paraphrase and output it to the end user. To build such a statistically reliable readability ranking model, the researchers conducted a large-scale questionnaire survey targeting teachers with expert knowledge at schools for the deaf. The authors selected 50 morpho-syntactic features to list in the survey, among which many are considered influential in sentence readability for deaf people. For each feature, several simple example sentences were collected from various sources. Each set of survey questions consists of a source sentence and a few paraphrases with the specific morpho-syntactic feature removed. 770 set of questions in total were prepared, out of which a random set of 240 were selected for each questionnaire survey. The questionnaires were sent to teachers at schools for deaf for readability ranking. Based on the collected data, a support vector machine classification technique was used to build the readability ranking model. The model achieved promising results with 95% precision and 89% recall.

The system's formalism for representing paraphrases was based on tree-to-tree transfer patterns. The input texts were internally represented as dependency trees. A broad range of transfer rules were handcrafted by observing lexical and structural transformation patterns between the paraphrase pair's dependency trees. In addition to the formalism of the tree-to-tree pattern representation, a text editor was added on top of that so a human rule writer can edit rules and specify transformation patterns more easily and efficiently using extended natural language.

To find out what kind of errors tend to occur in lexical and structural paraphrasing, a set of 1220 randomly selected sentences from newspaper articles were fed to the paraphrasing engine KURA. Typical errors observed in the output sentences included inappropriate verb conjugation, problems with verb valences, and errors related to the difference of meaning between substituted synonyms.

Many observed post-transfer errors have not been effectively resolved by the time when the paper was published. For this reason, the authors explained that an evaluation was not conducted. Without user- and task-oriented evaluations, it's unknown how effective the system may be at improving the comprehension of an input text by deaf people. Nevertheless, the Japanese researchers did explore some new approaches in text simplification by considering readability rank-

ing, paraphrase representation, and post-error detection, which have not been explored much before in the field. It's especially notable that they successfully built a statistically reliable readability ranking model based on a large-scale questionnaire survey. As Williams et al. (2003) pointed out, it is very important to base the development of text simplification systems for human readers on solid empirical evidence rather than on our own intuitions. The data collected from teachers who have expert knowledge with the target deaf users certainly contributed to the success of building such a model.

4.2.2 System for people without disabilities who have low literacy

SkillSum

SkillSum is a web-based system developed to generate readable texts for readers with low literacy skills. It's an on-going project between Cambridge Training and Development Ltd. and NLG researchers at Aberdeen University (Williams and Reiter, 2005). The primary goal of the system is to assess target users' literacy or numeracy skills and generate easy-to-read feedback that summarizes their performance. The system also makes suggestions to encourage those who are concerned about their skills to take steps to improve them. The way in which text simplification relates to this overall problem is that the authors want the "feedback" that is given to users to be presented in a simple, easy-to-read manner.

The linguistic issues the researchers currently focus on SkillSum are the choices related to the expression of discourse structure. The central question is how to order and express the phrases that are related by a discourse relation. Discourse relations in this research are based on (Mann and Thompson, 1987) rhetorical structure theory (RST). There are many rhetorical relations according to RST, including: motivation, antithesis, background, evidence, concession, condition, contrast, etc. Cue phrases, such as *if*, *then*, *so*, *however*, etc. are often used to signal these relations. Core messages of a document that are related by discourse relations are represented by RST trees. To express the phrases that are related by discourse relations in an easy-to-understand manner for poor readers, a list of discourse related choices need to be made, including:

- how many and which cue phrases to use?
- how to order the constituents that are related by a discourse relation?
- and how to punctuate the constituents?

A microplanner is created to make such choices based on hard constraints and optimization rules. Hard-constraint rules are used to forbid combining phrases ungrammatically. These rules are learned by analyzing the RST Discourse Treebank Corpus (RST-DTC) (Carlson et al., 2002). For each type of discourse relation, a number of instances of the same type are extracted from the RST-DTC and analyzed. Hard-constraint rules are formed by forbidding any combinations that are not present at any of the instances analyzed. The optimization rules make appropriate expression choices among legal combinations of phrases. The authors created two sets of optimization rules: control and enhanced-readability (ER). The ER model which the authors believed would outperform the control model in terms of readability has the following preferences:

- each discourse relation should be expressed by a single cue phrase;
- lexically common cue phrases are preferred;
- a cue phrase should be placed between constituents if possible, and the nucleus (core) should come first if possible;
- constituents should preferably be in separate sentences, or separated by a comma if they are in the same sentence.

To test the readability of the texts generated by the SkillSum system, the authors conducted several pilot studies with target users and adjusted the evaluation metrics accordingly based on their findings in each study. They initially tried to measure readability by asking comprehension questions and measuring reading rate by self-timed silent reading. These approaches turned out to be problematic. The comprehension questions asked the subjects how they performed on each task. The subjects responded these questions based on their beliefs about their literacy and math skills, not on the content of the text report generated by the system that summarizes their skills. Self-timed silent reading did not work for poor readers either because they tended to skim-read or even press the button without reading the text. So in the end they decided to ask subjects to read the generated texts aloud, and time the subjects' reading rate and count the number of reading errors as evaluation measures for the readability of the generated text. The system was evaluated on 60 subjects with moderately poor skills. The authors reported that on average the text generated using ER model was read 16 words per minute faster than that using control model, and the subjects spent on average an extra 714ms making errors on the control text, which is 82% more time than they did on ER text.

It is notable that their evaluation was user-centered. To best evaluate the performance of the system, they studied the characteristics of target users carefully during pilot studies and adjusted their final evaluation metric accordingly based on the findings mentioned above. An important lesson one can learn from their approach is that, when evaluating the system, users are also an important part of the process. To best evaluate the system, particular user characteristics and preferences need to be taken into account. For example, reading aloud as an evaluation metric apparently worked for their experimental setting, it may not be the best for systems designed for people with language or cognitive disabilities, who have difficulty in reading aloud. Also, it might be misleading to take the message from their evaluation approach that comprehension questions are not good for measuring readability. On the contrary, comprehension questions are one of the most commonly used evaluation measures to test readability. The reason they abandoned comprehension questions is that the text was about the user and that answering these questions involved user's self evaluation, which is a subjective variable that can not be controlled easily.

Williams and Reiter's research pointed out a new direction in the field of text simplification, especially when users with low reading skills are taken into consideration. The results of their experiment show how the constituents of a discourse relation are ordered and expressed can have big impact on reader's comprehension. However, the authors only considered a limited number of linguistic choices at discourse level (rearrange the order and express the constituents governed by a discourse relation), and did not provide any lexical or syntactic simplification. The overall readability of the generated text was not improved significantly.

4.2.3 System for language teachers, children and adult second language learners

ATA V.1.0 is an automated text adaptation tool developed by ETS for English language teachers to adapt written texts on topics of interest to appropriate reading levels. Such a tool could potentially save English teachers a lot of time in preparing classroom reading materials at an appropriate level of difficulty. English language learners such as children and adult L2 learners can also make use of it for independent study.

Reading-level appropriate texts are often hard to find, teachers often have to do some adaptations manually to make texts suitable for a certain reading-level. But manual text adaptation is time consuming. This motivates the creation of ETS's automated text adaptation tool (ATA v.1.0) (Burststein et al., 2007). The ATA v.1.0 tool

is a recent NLP application developed to help English language learners (ELLs) with their reading comprehension and English language skills development. It learns from teacher-based text adaptation methods, which includes text summarization, vocabulary support, and translation. The ATA v.1.0 uses the Rhex automatic summarization tool (Marcu, 2000) to generate marginal notes in English. The amount of marginal notes can be decreased or increased based on student's needs. The English marginal notes can be translated using Language Weaver's English-to-Spanish machine translation system. At the lexical level, synonyms for difficult words with lower frequency are generated using a statistically-generated word similarity matrix (Lin, 1998a), antonyms are generated using WordNet, and cognates (words with same spelling and meaning in two languages) are generated using ETS English/Spanish cognate lexicon. The ATA v.1.0 tool also provides English and Spanish text-to-speech for pronunciation support that students can optionally enable when reading.

The effectiveness of the tool has not been evaluated yet. But a small scale pilot survey by 12 teachers who were allowed to interact with the system showed that the tool can serve either as a potentially effective support for teachers with their lesson plans or as a student tool for independent work. The teachers commented that they liked the features of English marginal notes and vocabulary support, and the text-to-speech support was their least liked feature.

ETS's ATA tool is the first system to consider text summarization as part of text simplification process. While many other text simplification systems implemented syntactic simplification tool, or addressed syntactic simplification as an important factor in text simplification process, ETS did not attempt to syntactically simplify the original text.

Research on a system similar to ATA is currently also being carried out at University of Washington. Schwarm and Ostendorf have proposed to build a similar text simplification system for language learners. This system would utilize text summarization techniques together with paraphrasing and other simplification techniques to adapt texts into appropriate reading levels. At the moment, their work still focuses on readability assessment (Schwarm and Ostendorf, 2005; Petersen and Ostendorf, 2006, 2007).

4.3 Summary of the systems

Section 4 investigated the development of current text simplification systems. Depending on the purpose of the system and the target users it intended to serve, researchers in the field developed and applied various simplification techniques to

build each particular system. These simplification techniques can be categorized at three linguistic levels: lexical, syntactic, and discourse level. Critiques were also made in the end after each system was introduced and analyzed.

To summarize the analysis and critique of the systems discussed above, the Appendix includes a set of tables that provide a quick comparison of the technical advancements made in each system. Table 1 and 2 compare the technical achievements of each system made at lexical, syntactic and discourse level. Table 3 compares the strength and weakness of each system in terms of target users and the evaluation of system's performance. All three tables are listed in the Appendix at the end of this survey.

5 Conclusions

Looking back at the brief development history of text simplification systems in the past decade, simplification techniques went through three major stages. The field began with syntactic simplification, and important advances made at this stage are the following. First various chunking techniques that provide robust shallow syntactic analysis were developed and utilized to both identify and simplify complex sentence constructions. Second a formalism has been established for rule-based sentence transformation. Typical complex syntactic structures tackled include coordinated and subordinated sentences, relative clauses, appositives and passive voice.

In the second stage, lexical simplification techniques were developed to augment syntactic simplification for systems developed for human readers. Various simplification options were developed for lexical support. The main idea was to replace difficult words with their simpler synonyms. A common approach was to use WordNet as resources for finding synonyms. Dictionary explanations and pre-defined vocabulary list could also be used for paraphrasing. Later on, images that represent the concepts of the content words were also utilized for lexical support. Text-to-speech techniques were used as optional pronunciation support.

In the third stage, the focus shifted to discourse related problems. To preserve text coherence and cohesion after syntactic simplification, techniques for problems such as anaphora resolution and replacement, ordering of simplified sentences, choices of cue phrases, and ordering and expressing constituents governed by a particular discourse relation, were developed and employed in various user-oriented text simplification systems.

These three stages were not necessarily developed in an entirely sequential way.

Some of lexical simplifications and discourse related problems were addressed in parallel with syntactic simplifications.

6 Future prospects

After almost a decade of development, text simplification techniques have started to mature at the lexical and syntactic levels. Simplification at the discourse level has also drawn research attention in recent years; various experiments on discourse simplification (Carroll et al., 1999; Canning, 2002; Siddharthan, 2003; Williams and Reiter, 2005) have been conducted to make natural language texts more comprehensible for human readers, especially for people with language disabilities or people with low literacy skills. Still, there are many research topics related to text simplification that remain open for future work.

A common approach to lexical simplification is to replace difficult words with simpler synonyms. The problem with this approach is that the semantic meaning of the original word may be altered after simplification. Many words are polysemous – they have more than one meaning depending upon context in which they are used. None of the current approaches to lexical simplification has taken the meaning of a word in a particular context into consideration when replacing it with a simpler synonym. In future work, more research on word sense disambiguation should be done to refine lexical simplification.

Current approaches to syntactic simplification also bring some problems with them. Many systems simplified input text one sentence at a time. One problem with this approach is that it could lead to the lack of coherence and cohesion of resulting simplified text, because interactions across sentences were not taken into account. These discourse related issues have drawn research attention in the field (Carroll et al., 1999; Canning, 2002; Williams and Reiter, 2005). Because long and complex sentences are split into shorter ones, another direct consequence of this approach is the increased length of the overall document after simplification. Considering the computing capacity of today's machines, the impact of the increased length can have on automatic NLP systems is insignificant. But for human readers, especially those with language disabilities, the increased length of the document could be challenging for their reading comprehension, because longer text requires more attention and/or a longer memory span.

None of the current text simplification systems developed for people with or without language disabilities have yet offered a treatment for increased length of simplified text. Although ETS's text adaptation tool did use a text summarization

technique to make the texts shorter, it did not offer syntactic simplification.

A possible solution for this problem is to combine text simplification and text summarization techniques together to reduce the length of simplified text while preserving the meaning and important factual information contained within the original text. There are many technical and linguistic questions to consider when combining technologies from these two fields of NLP. For example, technically, how can we combine these technologies to produce the best result? How do we deal with new discourse level problems that result from combining text simplification and text summarization techniques together? Further research on technical and linguistic issues like these should have the potential to advance the state of the art of discourse-level text simplification/generation.

The performance of many developed text simplification systems, such as Chandrasekar et al.'s system, KURA, HAPPI, and ETS's ATA v.1.0, has not been evaluated. So another aspect that deserves future research attention in the field is how to design evaluation metrics to best evaluate them. In order to design feasible and effective evaluation metrics that can best evaluate the performance of a system, one has to take both the particular characteristics of the application and the target users into consideration. The performance of a text simplification system can and should be evaluated differently depending on the purpose of the system and its target users. For example, if the system is designed as a preprocessing tool for automatic NLP applications such as a parser, the performance of simplification would be better measured by its impact on the performance of the parser: whether it reduces parsing ambiguity, whether it improves parsing throughput, etc. All this can be done automatically. If the system is designed to assist humans with reading comprehension, then the system's performance is mainly reflected by the improved readability the simplified text achieved with the target users. Many evaluation examples discussed in section 4 (Siddharthan, 2003; Williams and Reiter, 2005) show that readability assessment is a tough problem. Many factors need to be considered while designing good and effective evaluation metrics to measure readability, for example, whether the target users have good literacy or whether they have language disabilities. And if the target users have language disabilities, it's important to do some research to find out what impact a particular language disability may have on these individuals affected, and how they may interact with the system differently from other users. Readability is one of many important aspects in evaluating the performance of a text simplification system; more future research still needs to be done on readability assessment.

Last but not least, there is also much to be done in future research to improve

human computer interaction of the state-of-the-art text simplification systems. Past research has paid some attention to this. For example, HAPPI project proposed to facilitate more autonomous interactions between users and the system by allowing the user to select the word they wish to simplify. ETS's researchers are implementing an editing capability to their text adaptation tool to allow teachers to make changes to generated outputs. More could be done for systems designed for people with language disabilities to make human computer interaction easier.

Table 1: A Comparison Table of Linguistic Issues Addressed by Each System

Systems	Lexical-level	Syntactic-level	Discourse-level
Chandrasekar et al. (1996, 1997)	not addressed	define articulation points where sentences can be logically split, adapt Supertagging techniques in a Dependency-based model to identify the clausal constituent and its span, then simplify using a set of ordered simplification rules.	not addressed
Siddharthan (2002)	not addressed	use pattern matching techniques on POS tagged text with noun and verb phrases chunked to identify simplifiable syntactic structures and phrasal boundaries, then apply hand-crafted rules to transform complex sentences into simpler ones.	generate referring expression, select determiners, decide the order of simplified sentences, preserve rhetorical relations and relative salience of entities, fix broken anaphoric links.
Klebanov et al. (2004)	not addressed	transform input text sentence-wise into Easy Access Sentences (EASes) containing a single verb and as many full Named Entities as possible.	not addressed
PSET (1996-2000)	replace words with common words using WordNet	apply a set of ordered simplification rules to change passive voice to active voice, split long conjoined sentences into shorter ones.	resolve and replace pronouns with their corresponding referents

Table 2: *Comparison Table Continued*

Systems	Lexical-level	Syntactic-level	Discourse-level
HAPPI (2006)	use WordNet which has been ported into MySQL database format, apply online speech tools to read a selected word aloud, and/or show images of the word using large scale online image databanks.	built on PSET, no new development	built on PSET, no new development
KURA (2003)	paraphrase and restrict the vocabulary to a top-2000 basic word set	handcraft tree-to-tree transformation rules for a broad range of lexical and structural transformation patterns and use these rules to generate various paraphrases for an input sentence.	not addressed
SkillSum (2005)	not addressed	not addressed	use hard constraints and optimization rules to order the constituents that are related by a discourse relation and express the relation in a more comprehensible way for readers with low skills.
ATA v.1.0 (2007)	generate synonyms for lower frequency words using a word similarity matrix, antonyms using WordNet and cognates using an ETS English/Spanish cognate lexicon.	not addressed	not addressed

Table 3: A Comparison Table of Systems Performance

System	Target Users	Performance Evaluation	Strength	Weakness
Chandrasekar	Parser	Not evaluated	Established the formalism for syntactic transformation, which influenced other researchers later in the field.	Raised many linguistic issues that could result from their approach of syntactic simplification, but did not answer any of them.
Siddharthan	Not specified	Evaluated on correctness (in term of grammaticality, meaning preservation and cohesion), readability and simplification level achieved.	Provided a more complete formalism for syntactic simplification, addressed many discourse issues related to maintaining text coherence	The system was developed without concentrating on specific application and/or target users, the goals of the system are too broad.
Klebanov et al.	Information-seeking applications	Evaluated on 123 sentences and achieved 50% precision and 30% recall.	Designed an algorithm to construct EASes (Easy Access Sentences) to make the search for relevant information easier and more effective.	The parser (MINIPAR) they used provided much incorrect parsing information.
PSET	People with aphasia	Only syntactic simplification was evaluated, impact was insignificant.	Combined lexical and syntactic simplification together.	The syntactic simplification was too narrow in scope.

Table 4: *Performance Comparison Table Continued*

HAPPI	People with aphasia	Not evaluated	Provided more options for lexical simplification.	Did not provide syntactic simplification measures.
KURA	Deaf people	Not evaluated	Explored readability ranking, paraphrase representation and post-error detection.	The readability ranking model was built in isolation, it was not clear how the model will be used.
SkillSum	People with low literacy (without disabilities)	Evaluated reading aloud and counting the number of reading errors	Made new approach in discourse level simplification by experimenting on ordering and expressing constituents of a discourse relation.	Choices of discourse simplifications are very limited, did not provide lexical or syntactic simplification.
ATA	English teachers, children and English language learners	Not evaluated	The first to considered text summarization as part of simplification process.	Did not provide syntactic simplification.

References

- Breck Baldwin. 1997. Cogniac: high precision coreference with limited knowledge and linguistic resources. In *Proceedings of a Workshop sponsored by the ACL (Operational factors in practical, robust anaphora resolution for unrestricted texts)*.
- Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34:211–231.
- Edward Briscoe and John Carroll. 1995. Generalized probabilistic lr parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19:25–60.
- *Jill Burstein, Jane Shore, John Sabatini, Yong-Won Lee, and Matthew Ventura. 2007. The automated text adaptation tool. In *NAACL-HLT '07*.
- Yvonne Canning. 2002. *Syntactic simplification of Text*. Ph.D. thesis, University of Sunderland, UK.
- David Caplan and Nancy Hilderbrandt. 1988. *Disorders of Syntactic Comprehension*. MIT Press.
- Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2002. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current Directions in Discourse and Dialogue*. Kluwer Academic Publisher.
- *John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of AAAI98 Workshop on Intergrating Artificial Intelligence and Assistive Technology*.
- *John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL '99)*.
- *Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, pages 1041–1044.
- *Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge Based Systems*, 10:183–190.
- Siobhan Devlin. 1999. *Simplifying natural language for aphasic readers*. Ph.D. thesis, University of Sunderland, UK.
- *Siobhan Devlin and Gary Unthank. 2006. Posters and demos: Helping aphasic people process online information. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility Assets '06*.

- Rudolf Flesch. 1979. *How to write plain Englis*. Harper and Brothers, New York.
- Claire Grover, Colin Matheson, Andrei Mikheev, and Marc Moens. 2000. Lt ttt - a flexible tokenisation tool. In *Proceedings of Second International Conference on Language Resources and Evaluation*.
- Judith A. Holt, Carol B., Traxler, and Thomas E. Allen. 1997. Interpreting the scores: A user's guide to the 9th edition stanford achievement test for educators of deaf and hard-of-hearing students. Technical report, Gallaudet Research Institute Technical Report 97-1, Washington, DC.
- *Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, and Ryu Iida. 2003. Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 9–16.
- Fergal.W. Jones, K. Long, and Michael L. Finlay. 2006. Assessing the reading comprehension of adults with learning disabilities. *Intellectual Disability Research*, 50:410–418.
- Nobuhiro Kaji, Daisuke Kawahara, Sadao Kurohash, and Satoshi Sato. 2002. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.
- *Beata Beigman Klebanov, Kevin Knight, and Daniel Marcu. 2004. Text simplification for information-seeking applications. In *On the Move to Meaningful Internet Systems*, volume 3290, pages 735–747. Springer-Verlag.
- Henry Kucera and W. Nelson Francis. 1967. *Computational analysis of present-day American English*. Brown University Press.
- Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *35th Annual Meetings of the Association for Computational Linguistics*.
- Dekang Lin. 1998b. Dependency-based evaluation of minipar. In *Workshop on the evaluation of parsing systems*.
- William C. Mann and Sandra A. Thompson. 1987. *Rhetorical Structure Theory: A Theory of Text Organization*. Information Sciences Institute.
- Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, Cambridge, Massachusetts.
- George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1993. *Five Papers on WordNet*. Technical report. Princeton University, Princeton, N.J.
- Sarah E. Petersen and Mari Ostendorf. 2006. A machine learning approach to reading level assessment. Technical report, University of Washington CSE Technical Report.

- Sarah E. Petersen and Mari Ostendorf. 2007. Text simplification for language learners: A corpus analysis. In *SLaTE Workshop on Speech and Language Technology in Education*.
- P. Quinlan. 1992. *The Oxford Psycholinguistic Database*. Oxford University Press.
- Sarah E. Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the Association for Computational Linguistics*.
- *Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Proceedings of the Language Engineering Conference*.
- Advaith Siddharthan. 2003. *Syntactic Simplification and Text Cohesion*. Ph.D. thesis, University of Cambridge.
- Advaith Siddharthan and Ann Copestake. 2002. Generating anaphora for simplifying text. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2002)*.
- Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of 20th International Conference on Computational Linguistics*.
- *Sandra Williams and Ehud Reiter. 2005. Generating readable texts for readers with low basic skills. In *Proceeding of the 10th European Workshop on Natural Language Generation*. Aberdeen.
- Sandra Williams, Ehud Reiter, and Liesl Osman. 2003. Experiments with discourse-level choices and readability. In *Proceedings of the 9th European Workshop on Natural Language Generation*, pages 127–134.